

Lineer Cebir — MIT 18.06 (Strang)'tan

ML Builder için Türkçe Notlar

Phase 1

2026-05-29

İçindekiler

1	Önsöz	1
2	Bu kitap nedir?	3
3	Nasıl Okumalı	5
4	35 Ders	7
5	Notasyon	9
6	Türkçe Terminoloji	11
7	Lineer Denklemlerin Geometrisi	13
7.1	Bu Derste Ne Var?	13
7.2	İki Denklem, İki Bilinmeyen	14
7.3	Birinci Bakış: Satır Görünümü	14
7.4	İkinci Bakış: Kolon Görünümü	15
7.5	Tüm Kombinasyonlar Neyi Doldurur?	18
7.6	Üç Boyuta Çıkış: 3×3 Sistem	18
7.7	Büyük Soru: Her \mathbf{b} İçin Çözüm Var mı?	20
7.8	Yüksek Boyutlarda Düşünmek	21
7.9	Matris \times Vektör: İki Hesaplama Yolu	22
7.10	Bu Dersin Özeti	22
7.11	Kontrol Soruları	23
7.12	Egzersizler	24
7.13	Sonraki Ders İçin Hazırlık	24
7.14	Anahtar Kavramlar (Cheat Sheet)	25
7.15	ML Bağlantıları Özeti	26
8	Matrislerle Eliminasyon	27
8.1	Bu Derste Ne Var?	27
8.2	Problem — Strang'ın 3×3 Sistemi	28
8.3	İleri Eliminasyon — Adım Adım	28
8.4	Başarısızlık Modları — Pivot Sıfır	29
8.5	Geri Yerine Koyma	30
8.6	Matris Dili — E_{ij} ile Yazmak	31
8.7	Eliminasyon Matrisleri	31
8.8	Birleşme Yasası — Parantezleri Kaydır	32
8.9	Permütasyon Matrisleri	33
8.10	Ters Matrise İlk Bakış	33
8.11	Bu Dersin Özeti	34

8.12	Kontrol Soruları	34
8.13	Egzersizler	35
8.14	Sonraki Ders İçin Hazırlık	36
8.15	Anahtar Kavramlar (Cheat Sheet)	37
8.16	ML Bağlantıları Özeti	37
9	Matris Çarpımı ve Ters Matrisler	39
9.1	Bu Derste Ne Var?	39
9.2	Matris Çarpımı — Beş Yol	40
9.2.1	Yol 1: Standart (sıra × kolon)	40
9.2.2	Yol 2: Kolon Yolu	40
9.2.3	Yol 3: Satır Yolu	40
9.2.4	Yol 4: Kolon × Satır Toplamı (rank-1 expansion)	41
9.2.5	Yol 5: Blok Çarpımı	42
9.3	Ters Matris — Tanım ve Var Olma Koşulu	42
9.4	Tersi Olmayan Matrisler — Üç Bakış	42
9.5	Gauss–Jordan: A^{-1} 'i Bulma Algoritması	43
9.6	Gauss–Jordan Neden Çalışıyor?	44
9.7	Bu Dersin Özeti	44
9.8	Kontrol Soruları	45
9.9	Egzersizler	46
9.10	Sonraki Ders İçin Hazırlık	46
9.11	Anahtar Kavramlar (Cheat Sheet)	47
9.12	ML Bağlantıları Özeti	47
10	A = LU Faktörizasyonu	49
10.1	Bu Derste Ne Var?	49
10.2	$(AB)^{-1} = B^{-1}A^{-1}$ — Çarpımın Tersisi	49
10.3	$(A^T)^{-1} = (A^{-1})^T$ — Transpoz ve Ters Yer Değiştirir	50
10.4	A = LU — Basit 2×2 Örneği	50
10.5	A = LDU — Pivotları Köşegene Çekme	51
10.6	3×3 — Çarpanlar L'de Doğrudan Oturuyor	52
10.7	Eliminasyon Maliyeti — $n^3/3$	52
10.8	Permütasyon Matrisleri ve PA = LU	53
10.9	$P^{-1} = P^T$ — Permütasyonların Sihirli Özelliği	53
10.10	Bu Dersin Özeti	54
10.11	Kontrol Soruları	54
10.12	Egzersizler	55
10.13	Sonraki Ders İçin Hazırlık	56
10.14	Anahtar Kavramlar (Cheat Sheet)	56
10.15	ML Bağlantıları Özeti	57
11	Transpoz, Permütasyon, Vektör Uzayları \mathbb{R}^n	59
11.1	Bu Derste Ne Var?	59
11.2	Genel PA = LU	59
11.3	Permütasyon Matrislerinin Özellikleri	60
11.4	Transpoz	61
11.5	Simetrik Matrisler	61

11.6	$R^T R$ Her Zaman Simetrik	61
11.7	Vektör Uzayları — R^2, R^3, R^n	62
11.8	Sıfır Vektör Zorunlu	63
11.9	Alt-Uzaylar (Subspace)	63
11.10	R^2 'nin ve R^3 'ün Tüm Alt-Uzayları	63
11.11	Kolon Uzayı $C(A)$ — Matristen Doğan Alt-Uzay	64
11.12	Bu Dersin Özeti	64
11.13	Kontrol Soruları	65
11.14	Egzersizler	65
11.15	Sonraki Ders İçin Hazırlık	66
11.16	Anahtar Kavramlar (Cheat Sheet)	66
11.17	ML Bağlantıları Özeti	67
12	Kolon Uzayı ve Sıfır Uzayı	69
12.1	Bu Derste Ne Var?	69
12.2	Alt-Uzaylar Üzerinde İşlemler	70
12.3	Kolon Uzayı $C(A)$ — Tanım	70
12.4	$Ax = b$ Ne Zaman Çözülebilir?	70
12.5	Pivot Kolonlar — Bağımlı Kolonu At	71
12.6	Sıfır Uzayı $N(A)$ — Tanım	71
12.7	Null Uzayını Bulmak	71
12.8	$N(A)$ Gerçekten Alt-Uzay mı? — İki Satırlık İspat	72
12.9	$Ax = b$ 'nin Çözümleri Niye Alt-Uzay Değil?	72
12.10	Bir Alt-Uzayı Tanımlamanın İki Yolu	73
12.11	Bu Dersin Özeti	73
12.12	Kontrol Soruları	73
12.13	Egzersizler	74
12.14	Sonraki Ders İçin Hazırlık	74
12.15	Anahtar Kavramlar (Cheat Sheet)	75
12.16	ML Bağlantıları Özeti	75
13	$Ax = 0$ Çözme — Pivot Değişkenler ve Özel Çözümler	77
13.1	Bu Derste Ne Var?	77
13.2	Eliminasyon Null Uzayını Korur	78
13.3	Dikdörtgen Eliminasyon \rightarrow Echelon Form U	78
13.4	Rank, Pivot Kolonlar, Serbest Kolonlar	78
13.5	Özel Çözümler — Serbest Değişkene 1 Ver	79
13.6	$n - r$ Formülü	79
13.7	Reduced Row Echelon Form (rref) R	80
13.8	Null Uzayı Matrisi — $N = \begin{pmatrix} -F \\ I \end{pmatrix}$ Kalıbı	80
13.9	$\text{rank}(A) = \text{rank}(A^T)$	80
13.10	Bu Dersin Özeti	81
13.11	Kontrol Soruları	81
13.12	Egzersizler	82
13.13	Sonraki Ders İçin Hazırlık	82
13.14	Anahtar Kavramlar (Cheat Sheet)	83
13.15	ML Bağlantıları Özeti	83

14	Ax = b Çözme — Tam Çözüm ve Rank	85
14.1	Bu Derste Ne Var?	85
14.2	Augmented Matris [A b]	85
14.3	Çözülebilirlik	87
14.4	Particular Çözüm — Serbest Değişkenleri Sıfırla	87
14.5	Tam Çözüm = $\mathbf{x}_p + N(A)$	87
14.6	Rank ve Boyut İlişkileri	88
14.7	Full Column Rank ($r = n$)	88
14.8	Full Row Rank ($r = m$)	89
14.9	$r = m = n$ — Tersinir	89
14.10	Bu Dersin Özeti	89
14.11	Kontrol Soruları	90
14.12	Egzersizler	90
14.13	Sonraki Ders İçin Hazırlık	91
14.14	Anahtar Kavramlar (Cheat Sheet)	91
14.15	ML Bağlantıları Özeti	91
15	Lineer Bağımsızlık, Baz ve Boyut	93
15.1	Bu Derste Ne Var?	93
15.2	$n > m \implies$ Null Uzayı Boş Değil	94
15.3	Lineer Bağımsızlık	94
15.4	Matris Dili — Bağımsızlık $\iff N(A) = \{\mathbf{0}\}$	94
15.5	Span	94
15.6	Baz (Basis)	95
15.7	Baz Tek Değil — Ama Boyut Sabit	95
15.8	Boyut	95
15.9	Temel Teorem — rank-nullity	96
15.10	Bu Dersin Özeti	96
15.11	Kontrol Soruları	97
15.12	Egzersizler	97
15.13	Sonraki Ders İçin Hazırlık	98
15.14	Anahtar Kavramlar (Cheat Sheet)	98
15.15	ML Bağlantıları Özeti	99
16	Dört Temel Alt-Uzay	101
16.1	Bu Derste Ne Var?	101
16.2	Dört Alt-Uzay — Yaşadıkları Uzaylar	101
16.3	Satır Uzayı $C(A^T)$	103
16.4	Sol Null Uzayı $N(A^T)$	103
16.5	Satır Rank = Kolon Rank — Harika Gerçek	104
16.6	Bazlar — Satır, Sol Null	104
16.7	İki Çift — \mathbb{R}^n ve \mathbb{R}^m Bölünmesi	105
16.8	Matris Uzayları	105
16.9	Bu Dersin Özeti	105
16.10	Kontrol Soruları	106
16.11	Egzersizler	107
16.12	Sonraki Ders İçin Hazırlık	107
16.13	Anahtar Kavramlar (Cheat Sheet)	107

16.14ML Bağlantıları Özeti	108
17 Matris Uzayları, Rank 1 ve Küçük Dünya Grafikleri	109
17.1 Bu Derste Ne Var?	109
17.2 Matris Uzayı M	110
17.3 Alt-Uzaylar — Simetrik, Üst Üçgensel, Diagonal	110
17.4 Boyut Formülü — Kesişim ve Toplam	110
17.5 Fonksiyon Uzayları — ODE Çözümleri	111
17.6 Rank 1 Matrisler — $A = \mathbf{uv}^T$	111
17.7 Rank 1 = Yapı Taşı	112
17.8 Sabit Rank Alt-Uzay Değil	112
17.9 Toplamı Sıfır Alt-Uzayı	112
17.10Dört Alt-Uzay — $A = (1, 1, 1, 1)$ Örneği	113
17.11Küçük Dünya Grafikleri	113
17.12Bu Dersin Özeti	113
17.13Kontrol Soruları	114
17.14Egzersizler	115
17.15Sonraki Ders İçin Hazırlık	115
17.16Anahtar Kavramlar (Cheat Sheet)	115
17.17ML Bağlantıları Özeti	116
18 Graflar, Ağlar ve Incidence Matrisleri	117
18.1 Bu Derste Ne Var?	117
18.2 Graf ve Incidence Matrisi	117
18.3 Null Uzayı — Potansiyeller ve Grounding	118
18.4 Sol Null Uzayı — Kirchhoff Akım Yasası	119
18.5 Loop Akımları — Bazlar	119
18.6 Ağaç (Tree)	119
18.7 Euler Formülü	120
18.8 A^TCA Çerçevesi — Uygulamalı Matematiğin Temeli	120
18.9 Graph Laplacian = A^TA	120
18.10Bu Dersin Özeti	121
18.11Kontrol Soruları	122
18.12Egzersizler	122
18.13Sonraki Ders İçin Hazırlık	123
18.14Anahtar Kavramlar (Cheat Sheet)	123
18.15ML Bağlantıları Özeti	123
19 Quiz 1 İncelemesi	125
19.1 Bu Derste Ne Var?	125
19.2 Span ve Boyut	125
19.3 Blok Matrisler ve Rank	127
19.4 Dört Alt-Uzayın Boyutları	127
19.5 Ters Mühendislik — Çözümünden Matrise	127
19.6 Çözülebilirlik	128
19.7 Doğru/Yanlış	128
19.8 $B^2 = 0$ ve Çarpım Null Uzayı	128
19.9 Satır vs Kolon Uzayı	128

19.10	Satır Operasyonları	129
19.11	Satır Uzayı \perp Null Uzayı — Ders 14 Önizleme	129
19.12	Bu Tekrarın Özeti	129
19.13	Kontrol Soruları	130
19.14	Egzersizler	131
19.15	Sonraki Ders İçin Hazırlık	131
19.16	Sınav Formülleri (Cheat Sheet)	131
19.17	ML Bağlantıları Özeti	132
20	Ortogonal Vektörler ve Alt-Uzaylar	133
20.1	Bu Derste Ne Var?	133
20.2	Ortogonal Vektörler — $\mathbf{x}^T \mathbf{y} = 0$	133
20.3	Pisagor ve Uzunluk	135
20.4	Ortogonal Alt-Uzaylar	135
20.5	Satır Uzayı \perp Null Uzayı	135
20.6	Ortogonal Tümlleyen	136
20.7	Temel Teorem 2. Kısım	136
20.8	En İyi Çözüm Problemi — Least Squares Motivasyonu	136
20.9	$A^T A$ Özellikleri	137
20.10	Bu Dersin Özeti	137
20.11	Kontrol Soruları	138
20.12	Egzersizler	138
20.13	Sonraki Ders İçin Hazırlık	139
20.14	Anahtar Kavramlar (Cheat Sheet)	139
20.15	ML Bağlantıları Özeti	139
21	Alt-Uzaylara Projeksiyon	141
21.1	Bu Derste Ne Var?	141
21.2	Doğruya Projeksiyon	142
21.3	Projeksiyon Matrisi P (1-D)	143
21.4	İki İmza Özelliği — $P^T = P, P^2 = P$	143
21.5	Çözümsüz $Ax = b \rightarrow$ Projeksiyon	143
21.6	Alt-Uzaya Projeksiyon — Normal Denklemler	143
21.7	Üç Formül	144
21.8	A Kare ise $P = I$	144
21.9	Doğru Fitleme — Sayısal Örnek	144
21.10	Least Squares Geometrisi	145
21.11	Bu Dersin Özeti	145
21.12	Kontrol Soruları	145
21.13	Egzersizler	146
21.14	Sonraki Ders İçin Hazırlık	146
21.15	Anahtar Kavramlar (Cheat Sheet)	147
21.16	ML Bağlantıları Özeti	147
22	Projeksiyon Matrisleri ve Least Squares	149
22.1	Bu Derste Ne Var?	149
22.2	Projeksiyon Recap — İki Uç	150
22.3	$b = p + e$ ve $I - P$	150

22.4 Doğru Fitleme — Sayısal Çözüm	150
22.5 İki Resim — Noktalar vs Vektörler	150
22.6 Least Squares = Kalkülüs	151
22.7 Outlier Uyarısı	151
22.8 $e \perp$ Kolon Uzayı — Doğrulama	152
22.9 $A^T A$ Tersinir \Leftrightarrow Bağımsız Kolonlar	152
22.10 Ortonormal Vektörler — $A^T A = I$	152
22.11 Bu Dersin Özeti	153
22.12 Kontrol Soruları	153
22.13 Egzersizler	154
22.14 Sonraki Ders İçin Hazırlık	154
22.15 Anahtar Kavramlar (Cheat Sheet)	154
22.16 ML Bağlantıları Özeti	155
23 Ortogonal Matrisler ve Gram-Schmidt	157
23.1 Bu Derste Ne Var?	157
23.2 Ortonormal Vektörler ve Q	158
23.3 $Q^T Q = I$	158
23.4 Ortogonal Matris (Kare) — $Q^T = Q^{-1}$	158
23.5 Projeksiyon $P = Q Q^T$	158
23.6 $\hat{x} = Q^T b$ — Normal Denklemler Çöker	159
23.7 Gram-Schmidt — Temel Fikir	159
23.8 $A = QR$ Ayrışımı	159
23.9 QR ile Least Squares — Sayısal Üstünlük	160
23.10 Bu Dersin Özeti	160
23.11 Kontrol Soruları	161
23.12 Egzersizler	161
23.13 Sonraki Ders İçin Hazırlık	162
23.14 Anahtar Kavramlar (Cheat Sheet)	162
23.15 ML Bağlantıları Özeti	162
24 Determinant ve Özellikleri	165
24.1 Bu Derste Ne Var?	165
24.2 Özellik 1, 2 — $\det(I) = 1$, Satır Takası	166
24.3 Özellik 3 — Satır-Lineerlik	166
24.4 Özellik 4 — Eşit Satır $\rightarrow \det = 0$	166
24.5 Özellik 5, 6 — Eliminasyon ve Sıfır Satır	167
24.6 Özellik 7 — Üçgensel = Pivot Çarpımı	167
24.7 Özellik 8 — $\det = 0 \Leftrightarrow$ Singular	167
24.8 Özellik 9 — $\det(AB) = \det(A) \cdot \det(B)$	167
24.9 Özellik 10 — $\det(A^T) = \det(A)$	168
24.10 Determinant = Hacim	168
24.11 Bu Dersin Özeti	168
24.12 Kontrol Soruları	169
24.13 Egzersizler	169
24.14 Sonraki Ders İçin Hazırlık	170
24.15 Anahtar Kavramlar (Cheat Sheet)	170
24.16 ML Bağlantıları Özeti	170

25	Determinant Formülleri ve Kofaktörler	173
25.1	Bu Derste Ne Var?	173
25.2	2×2 'yi Özelliklerden Türetme	173
25.3	Survivor'lar ve Büyük Formül	175
25.4	Permütasyon İşareti	175
25.5	Çok Sıfır = Az Survivor	175
25.6	Kofaktör Açılımı	176
25.7	Üç Formülün Karşılaştırması	176
25.8	Tridiagonal Örnek — Recursion ve Periyot 6	177
25.9	Bu Dersin Özeti	177
25.10	Kontrol Soruları	178
25.11	Egzersizler	178
25.12	Sonraki Ders İçin Hazırlık	179
25.13	Anahtar Kavramlar (Cheat Sheet)	179
25.14	ML Bağlantıları Özeti	179
26	Cramer Kuralı, Ters Matris ve Hacim	181
26.1	Bu Derste Ne Var?	181
26.2	Ters Matris Formülü	182
26.3	Cramer Kuralı	182
26.4	Determinant = Hacim	183
26.5	Hacim = Determinant İspatı	183
26.6	Alan Formülü — Paralelkenar ve Üçgen	183
26.7	Bu Dersin Özeti	184
26.8	Kontrol Soruları	184
26.9	Egzersizler	185
26.10	Sonraki Ders İçin Hazırlık	185
26.11	Anahtar Kavramlar (Cheat Sheet)	185
26.12	ML Bağlantıları Özeti	186
27	Özdeğerler ve Özvektörler	187
27.1	Bu Derste Ne Var?	187
27.2	$Ax = \lambda x$ — Özvektör	188
27.3	$\lambda = 0 \iff$ Singular	188
27.4	Geometriden Özdeğer Okuma	188
27.5	Trace = $\sum \lambda$, det = $\prod \lambda$	189
27.6	Karakteristik Denklem	189
27.7	Tam Örnek — $A = \begin{pmatrix} 3 & 1 \\ 1 & 3 \end{pmatrix}$	189
27.8	$A + cI$ ve $A + B$	190
27.9	Kompleks Özdeğerler — Rotasyon	190
27.10	Tekrarlı Özdeğer — Degenerate	190
27.11	Bu Dersin Özeti	191
27.12	Kontrol Soruları	191
27.13	Egzersizler	192
27.14	Sonraki Ders İçin Hazırlık	192
27.15	Anahtar Kavramlar (Cheat Sheet)	192
27.16	ML Bağlantıları Özeti	193

28 Diagonalization ve A^k	195
28.1 Bu Derste Ne Var?	195
28.2 $AS = SA$ ve $A = SAS^{-1}$	196
28.3 $A^k = SA^kS^{-1}$	196
28.4 Kararlılık	197
28.5 Multiplicity	197
28.6 Fark Denklemleri ve Spektral Çözüm	197
28.7 Fibonacci Örneği	198
28.8 Bu Dersin Özeti	198
28.9 Kontrol Soruları	198
28.10Egzersizler	199
28.11Sonraki Ders İçin Hazırlık	199
28.12Anahtar Kavramlar (Cheat Sheet)	200
28.13ML Bağlantıları Özeti	200
29 Diferansiyel Denklemler ve e^{At}	201
29.1 Bu Derste Ne Var?	201
29.2 $du/dt = Au$ ve Üstel Çözümler	202
29.3 Örnek — Steady State	202
29.4 Kararlılık — $\text{Re}(\lambda) < 0$	202
29.5 Kompleks Özdeğerler — Osilasyon	202
29.6 2×2 Kararlılık — Trace ve Determinant	203
29.7 Uncoupling — $u = Sv$	203
29.8 Matris Üstel — e^{At}	203
29.9 $e^{At} = Se^{At}S^{-1}$	203
29.10İkinci-Mertebe \rightarrow Sistem (Companion)	204
29.11Bu Dersin Özeti	204
29.12Kontrol Soruları	204
29.13Egzersizler	205
29.14Sonraki Ders İçin Hazırlık	205
29.15Anahtar Kavramlar (Cheat Sheet)	205
29.16ML Bağlantıları Özeti	206
30 Markov Matrisleri ve Fourier Serisi	207
30.1 Bu Derste Ne Var?	207
30.2 Markov Matrisi	208
30.3 $\lambda = 1$ Garantili	208
30.4 Diğer λ ve Steady State	208
30.5 A ve A^T Aynı Özdeğer	208
30.6 Markov Örneği — CA/MA	209
30.7 Ortonormal Bazda Projeksiyon	209
30.8 Fourier Serisi	209
30.9 Fonksiyon İç Çarpımı	210
30.10Fourier Katsayıları	210
30.11Bu Dersin Özeti	210
30.12Kontrol Soruları	211
30.13Egzersizler	211
30.14Sonraki Ders İçin Hazırlık	211

30.15 Anahtar Kavramlar (Cheat Sheet)	212
30.16 ML Bağlantıları Özeti	212
31 Quiz 2 İncelemesi	213
31.1 Bu Derste Ne Var?	213
31.2 Kapsam	213
31.3 Projeksiyon Özdeğerleri	215
31.4 P + Fark Denklemi	215
31.5 Least Squares — İki Resim	215
31.6 Gram-Schmidt	215
31.7 Özdeğer Cebri	216
31.8 Tridiagonal Recurrence ve Kompleks Özdeğer	216
31.9 Trace Kontrolü — Strang'ın Disiplini	216
31.10 Bu Tekrarın Özeti	216
31.11 Kontrol Soruları	217
31.12 Egzersizler	217
31.13 Sonraki Ders İçin Hazırlık	218
31.14 Sınav Formülleri (Cheat Sheet)	218
31.15 ML Bağlantıları Özeti	218
32 Simetrik Matrisler ve Pozitif Tanımlılık	221
32.1 Bu Derste Ne Var?	221
32.2 İki Ana Gerçek	221
32.3 Spektral Teorem — $A = Q\Lambda Q^T$	222
32.4 Gerçek Özdeğer İspatı	222
32.5 Spektral Ayrışım	223
32.6 Pivot İşareti = Özdeğer İşareti	223
32.7 Pozitif Tanım — Tanım ve Testler	223
32.8 Örnek — $A = ((2, 1), (1, 2))$	224
32.9 Kursun Birleştiği Nokta	224
32.10 Bu Dersin Özeti	224
32.11 Kontrol Soruları	225
32.12 Egzersizler	225
32.13 Sonraki Ders İçin Hazırlık	225
32.14 Anahtar Kavramlar (Cheat Sheet)	226
32.15 ML Bağlantıları Özeti	226
33 Kompleks Matrisler ve FFT	227
33.1 Bu Derste Ne Var?	227
33.2 Kompleks Uzunluk — $\mathbf{z}^H \mathbf{z}$	228
33.3 Hermitian Matris — $A^H = A$	228
33.4 Unitary Matris — $Q^H Q = I$	228
33.5 Fourier Matrisi F_n	228
33.6 Ters Matris	229
33.7 FFT Fikri — F_{2n} ile F_n	229
33.8 $n^2 \rightarrow \frac{1}{2}n \log_2 n$	229
33.9 Bu Dersin Özeti	230
33.10 Kontrol Soruları	230

33.11Egzersizler	230
33.12Sonraki Ders İçin Hazırlık	231
33.13Anahtar Kavramlar (Cheat Sheet)	231
33.14ML Bağlantıları Özeti	231
34 Pozitif Tanımlı Matrisler ve Minimumlar	233
34.1 Bu Derste Ne Var?	233
34.2 Dört Eşdeğer Test	234
34.3 Kuadratik Form	234
34.4 Üç Örnek	234
34.5 Kareye Tamamlama = Eliminasyon	235
34.6 Hessian ve Minimum	235
34.7 Strang'ın 3×3 Favori	235
34.8 Geometri — Ellipsoid	236
34.9 Bu Dersin Özeti	236
34.10Kontrol Soruları	236
34.11Egzersizler	237
34.12Sonraki Ders İçin Hazırlık	237
34.13Anahtar Kavramlar (Cheat Sheet)	237
34.14ML Bağlantıları Özeti	238
35 Benzer Matrisler ve Jordan Formu	239
35.1 Bu Derste Ne Var?	239
35.2 $A^T A$ Pozitif Tanım	240
35.3 Pozitif Tanım Ek Gerçekler	240
35.4 Benzer Matris	240
35.5 Aynı Özdeğer İspatı	240
35.6 Aile Örneği — Özdeğer 3, 1	240
35.7 Tekrarlı Özdeğer — İki Aile	241
35.8 Jordan Formu	241
35.9 Jordan Teoremi	241
35.10Bu Dersin Özeti	241
35.11Kontrol Soruları	242
35.12Egzersizler	242
35.13Sonraki Ders İçin Hazırlık	243
35.14Anahtar Kavramlar (Cheat Sheet)	243
35.15ML Bağlantıları Özeti	243
36 Tekil Değer Ayrışımı (SVD)	245
36.1 Bu Derste Ne Var?	245
36.2 SVD — $A = U\Sigma V^T$	245
36.3 Geometrik Hedef	247
36.4 Matris Diline — $AV = U\Sigma$	247
36.5 U 'ları Yok Etme — $A^T A = V\Sigma^2 V^T$	247
36.6 Hesap Reçetesi	247
36.7 Örnek 1 — $A = \begin{pmatrix} 4 & 4 \\ -3 & 3 \end{pmatrix}$	248
36.8 Örnek 2 — Rank 1	248

36.9 SVD ve Dört Temel Alt-Uzay	248
36.10 Simetrik PD — Özel Durum	249
36.11 Bu Dersin Özeti	249
36.12 Kontrol Soruları	249
36.13 Egzersizler	250
36.14 Sonraki Ders İçin Hazırlık	250
36.15 Anahtar Kavramlar (Cheat Sheet)	250
36.16 ML Bağlantıları Özeti	251
37 Lineer Dönüşümler ve Matrisleri	253
37.1 Bu Derste Ne Var?	253
37.2 Lineer Dönüşüm Tanımı	254
37.3 Örnekler ve Karşı-Örnekler	254
37.4 $T(\mathbf{v}) = A\mathbf{v}$	254
37.5 Baz Yeter	254
37.6 İki Baz — Girdi + Çıktı	254
37.7 Matris Kurma Kuralı	255
37.8 Projeksiyon — İyi Baz vs Standart Baz	255
37.9 Türev Lineer Dönüşümdür	255
37.10 Bu Dersin Özeti	256
37.11 Kontrol Soruları	256
37.12 Egzersizler	256
37.13 Sonraki Ders İçin Hazırlık	257
37.14 Anahtar Kavramlar (Cheat Sheet)	257
37.15 ML Bağlantıları Özeti	258
38 Baz Değişimi ve Görüntü Sıkıştırma	259
38.1 Bu Derste Ne Var?	259
38.2 Görüntü = Vektör; Standart Baz Kötü	260
38.3 JPEG — Fourier Bazı, 8×8 Blok	260
38.4 Wavelet Bazı	260
38.5 Baz Değişimi — $\mathbf{x} = W\mathbf{c}$	260
38.6 Benzer Matrisler — $B = M^{-1}AM$	261
38.7 Bu Dersin Özeti	261
38.8 Kontrol Soruları	261
38.9 Egzersizler	262
38.10 Sonraki Ders İçin Hazırlık	262
38.11 Anahtar Kavramlar (Cheat Sheet)	262
38.12 ML Bağlantıları Özeti	263
39 Quiz 3 İncelemesi	265
39.1 Bu Derste Ne Var?	265
39.2 Kapsam	265
39.3 Problem 1: Anti-Simetrik ODE	267
39.4 Problem 2: “Hangi c ?”	267
39.5 SVD İşaret Tuzağı	267
39.6 SVD’den Matris Okuma	267
39.7 Simetrik + Ortogonal — $\lambda = \pm 1$	268

39.8	$\frac{1}{2}(A + I)$ Projeksiyon mu?	268
39.9	Özet — Parmak İzi Tablosu	268
39.10	Kontrol Soruları	269
39.11	Egzersizler	269
39.12	Sonraki Ders İçin Hazırlık	269
39.13	Anahtar Kavramlar (Cheat Sheet)	270
39.14	ML Bağlantıları Özeti	270
40	Sol ve Sağ Tersler, Pseudoinverse	271
40.1	Bu Derste Ne Var?	271
40.2	Dört Durum	271
40.3	Sol Ters — Tam Kolon Rank	272
40.4	Sağ Ters — Tam Satır Rank	272
40.5	Yanlış Tarafa \rightarrow Projeksiyon	272
40.6	Pseudoinverse — Satır \leftrightarrow Kolon Mucizesi	272
40.7	SVD'den A^+	274
40.8	Σ^+ ve Projeksiyonlar	274
40.9	İstatistik / Least Squares Bağlantısı	274
40.10	Bu Dersin Özeti	274
40.11	Kontrol Soruları	275
40.12	Egzersizler	275
40.13	Sonraki Ders İçin Hazırlık	276
40.14	Anahtar Kavramlar (Cheat Sheet)	276
40.15	ML Bağlantıları Özeti	276
41	Final İncelemesi	277
41.1	Bu Derste Ne Var?	277
41.2	Soru 1: Rank ve Çözülebilirlik	278
41.3	Soru 1 T/F	278
41.4	$A^T y = c$ — Varlık + Çokluk	278
41.5	Soru 2: Kolon Kombinasyonu	278
41.6	Soru 3: Markov Kararlı Durum	278
41.7	Soru 4: 2×2 Hızlı	279
41.8	Soru 5: Least Squares	279
41.9	Final Stratejisi	279
41.10	Bu Dersin Özeti	279
41.11	Kontrol Soruları	280
41.12	Egzersizler	280
41.13	Kurs Tamamlandı 🎓	280
41.14	Cheat Sheet (Kurs Geneli)	281
41.15	ML Bağlantıları (Kapanış)	281

1 Önsöz

2 Bu kitap nedir?

Bu, MIT 18.06 — **Linear Algebra** dersinin (Gilbert Strang, 35 ders) Türkçe öğretim notlarıdır. Hedef: Strang'i izlerken paralel okunabilecek; tek başına da yeterli olabilecek bir referans seti üretmek.


Her ders bir **Builder Notu** katmanı taşır: kavramın **makine öğrenmesi** ile köprüsü. Matris çarpımı → forward pass; özdeğer/özvektör → PCA, attention'da scale; projeksiyon → en küçük kareler (least squares regression); SVD → boyut indirgeme, LoRA; null space → degenerate çözümler, dropout; LU/QR → sayısal stabilite. Lineer cebiri “tek başına matematik” olarak değil, ML'i türeten alet kutusu olarak okuyoruz.

Kaynak

- **Video dizisi:** [MIT 18.06SC Linear Algebra, Fall 2011](#) — YouTube playlist: [PL221E2BBF13BECF6C](#)
- **Yazar:** Gilbert Strang — [MIT Mathematics](#)
- **Kitap:** *Introduction to Linear Algebra* (Wellesley-Cambridge Press)
- **Çeviri ve genişletme:** Phase 1 (TR + ML köprüleri)

3 Nasıl Okumalı

Strang'ın sırası 50+ yıl test edilmiş — onu izle. Her ders bir öncekinin **kelimelerini** kullanır: Ders 6'daki “kolon uzayı”, Ders 10'daki “dört temel alt-uzay”ın ön sözüdür; Ders 21-22 (“özdeğer/diagonalization”) olmadan Ders 29 (SVD) anlamsızdır.

 Pratik bir tavsiye

Strang'ın **videosunu önce izle** (≈ 50 dk). Sonra Türkçe dersi oku — kafanda netleşmeyen yerler için. **Kontrol sorularını** toggle açmadan cevapla. Python egzersizleri ML sezgisini parmaklarına yerleştirir; aynı numpy kodunu yıllarca farklı kılıklarda yazacaksın.

4 35 Ders

#	Ders	Ana Fikir	ML Köprüsü
1	Lineer Denklemlerin Geometrisi	Row + column picture	Vektör uzayında veri
2	Matrislerle Eliminasyon	Gauss eliminasyonu	Sayısal stabilite
3	Matris Çarpımı ve Ters	5 çarpma yorumu	Forward pass
4	A = LU Faktörizasyonu	Eliminasyonu çarpan olarak yaz	Nümerik LA
5	Transpoz, Permütasyon, \mathbb{R}^n	Vektör uzayı tanımı	Latent space
6	Kolon Uzayı ve Sıfır Uzayı	C(A) ve N(A)	Çözüm kümesi
7	Ax = 0: Pivot Değişkenler	Rref + özel çözümler	Dropout / degenerate
8	Ax = b: Tam Çözüm ve Rank	Tutarlılık + rank	Underdetermined sistem
9	Lineer Bağımsızlık, Baz, Boyut	Boyut kavramı	Feature redundancy
10	Dört Temel Alt-Uzay	C(A), N(A), C(A ^T), N(A ^T)	Forward + backward
11	Matris Uzayları, Rank 1	Rank 1 matrisin gücü	Düşük-rank yaklaşım
12	Graflar ve Incidence	A ^T A = Laplacian	Graph neural net
13	Quiz 1 İncelemesi	Eliminasyon + 4 alt-uzay	—
14	Ortogonal Vektörler	Ortogonallık	Bağımsız feature'lar
15	Alt-Uzaylara Projeksiyon	$p = A\hat{a}$	Regresyon prosedürü
16	Projeksiyon ve Least Squares	$\hat{x} = (A^T A)^{-1} A^T b$	Lineer regresyon
17	Ortogonal Matrisler ve Gram-Schmidt	Q, QR	İstikrarlı bazlar
18	Determinant ve Özellikleri	det'in 10 özelliği	Hacim, çakışma

#	Ders	Ana Fikir	ML Köprüsü
19	Determinant Formülleri	Big-formula, kofaktör	—
20	Cramer, Ters, Hacim	det ile geometri	—
21	Özdeğerler ve Özvektörler	$Ax = \lambda x$	Power method, PageRank
22	Diagonalization ve A^k	$A = SAS^{-1}$	Hızlı matris üs alma
23	$e^{(At)}$	Diferansiyel sistem çözümü	RNN, Neural ODE
24	Markov ve Fourier	Stochastic, ortogonal baz	Diffusion, Transformer
24b	Quiz 2 İncelemesi	Özdeğer odaklı	—
25	Simetrik ve Pozitif Tanım	$A = Q\Lambda Q^T$	Covariance, Hessian
26	Kompleks Matrisler ve FFT	Unitary, FFT	Sinyal işleme
27	Pozitif Tanım ve Minimum	Konvekslik testi	Optimizasyon
28	Benzer Matrisler ve Jordan	$B = M^{-1}AM$	Nadir, ama kavramsal
29	SVD	$A = U\Sigma V^T$	LoRA, PCA, low-rank
30	Lineer Dönüşümler	T: $V \rightarrow W$	Bazsız tanımlama
31	Baz Değişimi ve Sıkıştırma	Wavelet, JPEG	Görüntü compress
32	Quiz 3 İncelemesi	SVD + LS	—
33	Sol/Sağ Tersler, Pseudoinverse	A^+	Min-norm çözüm
34	Final İncelemesi	Hepsi	—

5 Notasyon

- **Vektör:** \mathbf{v} veya \vec{v} — kolon vektörü kabul edilir
- **Matris:** A, B — büyük harf
- **Transpoz:** A^T — satır \leftrightarrow kolon
- **Ters:** A^{-1} — sadece kare ve singular değilse
- **İç çarpım:** $\mathbf{x} \cdot \mathbf{y} = \mathbf{x}^T \mathbf{y}$
- **Norm:** $\|\mathbf{x}\| = \sqrt{\mathbf{x}^T \mathbf{x}}$
- **4 alt-uzay:** $C(A)$ kolon uzayı, $N(A)$ sıfır uzayı, $C(A^T)$ satır uzayı, $N(A^T)$ sol sıfır uzayı
- **Özdeğer:** λ , **özvektör:** \mathbf{x} veya \mathbf{v}
- **SVD:** $A = U\Sigma V^T$ — U, V ortogonal, Σ singular değerler köşegeni

Tüm matematik [KaTeX](#) ile render ediliyor.

6 Türkçe Terminoloji

Türkçe matematik literatüründe lineer cebir kelime dağarcığı yerleşmemiş. Bu set boyunca tutarlı çeviriler:

İngilizce	Türkçe	Not
Vector	Vektör	—
Matrix	Matris	—
Linear combination	Lineer kombinasyon	“Doğrusal birleşim” de var; biz lineer
Row	Satır	—
Column	Kolon	Strang’le eşleşsin diye “kolon”
Row/column picture	Satır/kolon görünümü	—
Span	Geriş / kapsama	Konuya göre
Linear independence	Lineer bağımsızlık	—
Rank	Rank	Çeviri yok
Null space	Sıfır uzayı	$N(A)$
Column space	Kolon uzayı	$C(A)$
Pivot	Pivot	—
Eigenvalue/eigenvector	Özdeğer/özvektör	—
SVD	Tekil değer ayrışımı (SVD)	Kısaltma İngilizce
Determinant	Determinant	—
Dot product	İç çarpım / dot product	İkisi de geçer
Least squares	En küçük kareler	—
Projection	Projeksiyon	“İzdüşüm” de var; biz projeksiyon
Embedding	Gömme / embedding	ML terimi, İngilizce yaygın
Forward pass	İleri geçiş / forward pass	—

! Bir tek şey

Lineer cebir, **vektör uzayları ve onlar arasındaki lineer dönüşümlerin** matematiğidir. Bir matris A , bir lineer dönüşümdür; onun **dört temel alt-uzayı** ($C(A)$, $N(A)$, $C(A^T)$, $N(A^T)$) tüm hikayeyi anlatır. Özdeğer/özvektör ve SVD bu hikayenin en güçlü iki anahtarıdır — birincisi “ A ’yı kendi yönlerine ayırıştır”, ikincisi “ A ’yı en iyi doğrularla yaklaştır” der. Hemen hemen tüm modern ML buraya bağlanır.

7 Linear Denklemlerin Geometrisi

Satır, kolon, matris — üç bakış, bir denklem

i Bölüm bilgisi

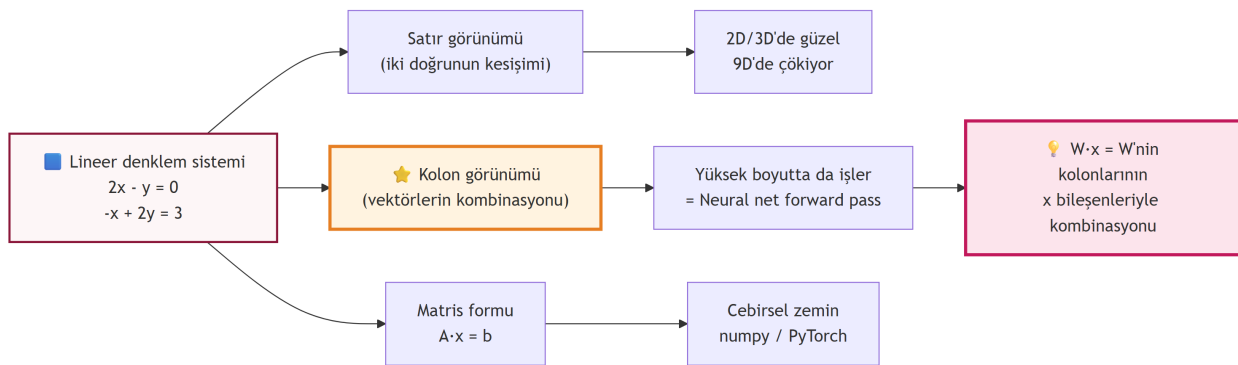
- **Strang'ın videosu:** [YouTube — Lecture 1: The Geometry of Linear Equations](#) (≈40 dk)
- **OCW sayfası:** [MIT 18.06SC — Lecture 1](#)
- **Okuma süresi:** ≈30 dk

7.1 Bu Derste Ne Var?

Bu lineer cebir kursunun ilk dersi. Strang'ın tüm sonraki 33 dersin üstüne bina edeceği üç bakış açısını koyuyor: aynı denklem sistemine **satır görünümü**, **kolon görünümü** ve **matris formu** gözüyle bakmak. Üç bakış aynı sayıyı verir; üçü farklı şeyler öğretir.

Üç bakış açısı:

1. **Satır görünümü (Row picture)** — Her denklemi geometrik olarak bir doğru/düzlem olarak çiz, kesişim noktasını ara. Liseden tanıdık.
2. **Kolon görünümü (Column picture)** — Matrisin kolonlarına vektör olarak bak; hangi kombinasyon b 'ye ulaşır? Strang dersin **en kritik kavramı** olduğunu vurguluyor.
3. **Matris formu** — Sistem $Ax = b$. Cebirsel/programatik yaklaşım, BLAS'tan PyTorch'a kadar her yerde.



Şekil 7.1: Aynı problem, üç bakış — ve modern ML için neden kolon görünümü kritik.

“Pay attention, this is the key point.” — Strang, 8:39

💡 Builder Notu — ML Köprüleri

- **$Ax =$ kolonların kombinasyonu.** Her neural network katmanı $y = Wx + b$ bir matris \times vektör. Kolon yorumu sana embedding’lerin, attention head’lerin, LoRA decomposition’unun çalışma biçimini anlatır. Satır (dot product) yorumu sadece “hesap nasıl yapıldı”yı söyler.
- **Kolon uzayı = ulaşılabilir çıkışlar.** W ’nin kolonlarının kombinasyonları, ağ katmanının üretebileceği tüm vektörler. Bottleneck katmanı bu kavramla tasarlanır.
- **Singular = düşük rank = LoRA fırsatı.** Strang “kolonlar bağımlıysa trouble” diyor; ML’de “ ΔW zaten düşük rank, $A \cdot B^T$ olarak parametrize edelim” deyip LoRA çıkıyor.
- **Yüksek boyut sezgisi.** 768D, 12288D’de çizemezsin ama “9D’de pretend” alışkanlığı tüm ML çalışmasının zihinsel zemini.

7.2 İki Denklem, İki Bilinmeyen

Hadi somutla başlayalım:

$$\begin{aligned} 2x - y &= 0 \\ -x + 2y &= 3 \end{aligned}$$

Hangi (x, y) değerleri *her iki* denklemi sağlar?

Matris formuna çevirelim:

$$\underbrace{\begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}}_A \underbrace{\begin{pmatrix} x \\ y \end{pmatrix}}_x = \underbrace{\begin{pmatrix} 0 \\ 3 \end{pmatrix}}_b$$

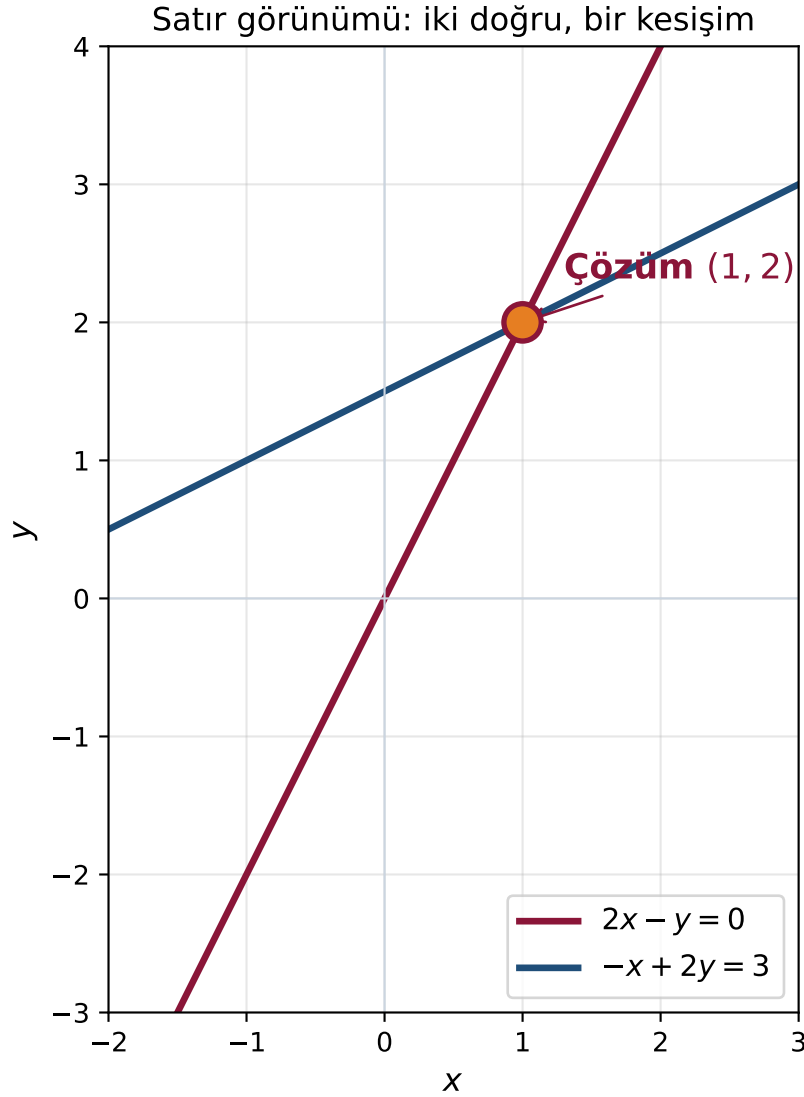
- A = katsayı matrisi, her satır bir denklemin katsayıları.
- x = bilinmeyenler vektörü.
- b = sağ taraf vektörü.

Bu format dersin geri kalanında ve gerçek hayatta (numpy, PyTorch, MATLAB) sürekli karşına çıkacak.

7.3 Birinci Bakış: Satır Görünümü

Lise matematiği: her denklem bir doğru, çözüm = kesişim noktası.

- $2x - y = 0 \rightarrow y = 2x$ (eğim 2, orijinden geçer)
- $-x + 2y = 3 \rightarrow y = (x + 3)/2$ (eğim 1/2, orijinden geçmez)



Şekil 7.2: Satır görünümü: iki doğrunun kesişimi (1, 2). 2D’de güzel; 3D’de düzlemlerin kesişimi gözle bulunması zor; 9D’de imkânsız.

İki denklem, iki bilinmeyen için bu güzel. 3 boyutta her denklem bir **düzlem**, üç düzlemin kesişimi bir nokta — hâlâ çizilebilir ama zor. 9 boyutta her denklem 8 boyutlu bir “hyperplane” — zihinde canlandırılmaz. Bu yüzden satır görünümü iyi bir başlangıç; ölçeklenmek için kolon görünümüne geçeceğiz.

7.4 İkinci Bakış: Kolon Görünümü

Strang dersin **kalbi**. Aynı sistemi farklı yaz:

$$x \begin{pmatrix} 2 \\ -1 \end{pmatrix} + y \begin{pmatrix} -1 \\ 2 \end{pmatrix} = \begin{pmatrix} 0 \\ 3 \end{pmatrix}$$

7 Lineer Denklemlerin Geometrisi

Doğrula:

- 1. bileşen: $x \cdot 2 + y \cdot (-1) = 2x - y \checkmark$
- 2. bileşen: $x \cdot (-1) + y \cdot 2 = -x + 2y \checkmark$

Matematiksel hiçbir şey değişmedi — **bakış açısı** değişti. Şimdi iki kolon vektörü var:

- $\mathbf{c}_1 = (2, -1)$ — A 'nın birinci kolonu
- $\mathbf{c}_2 = (-1, 2)$ — A 'nın ikinci kolonu

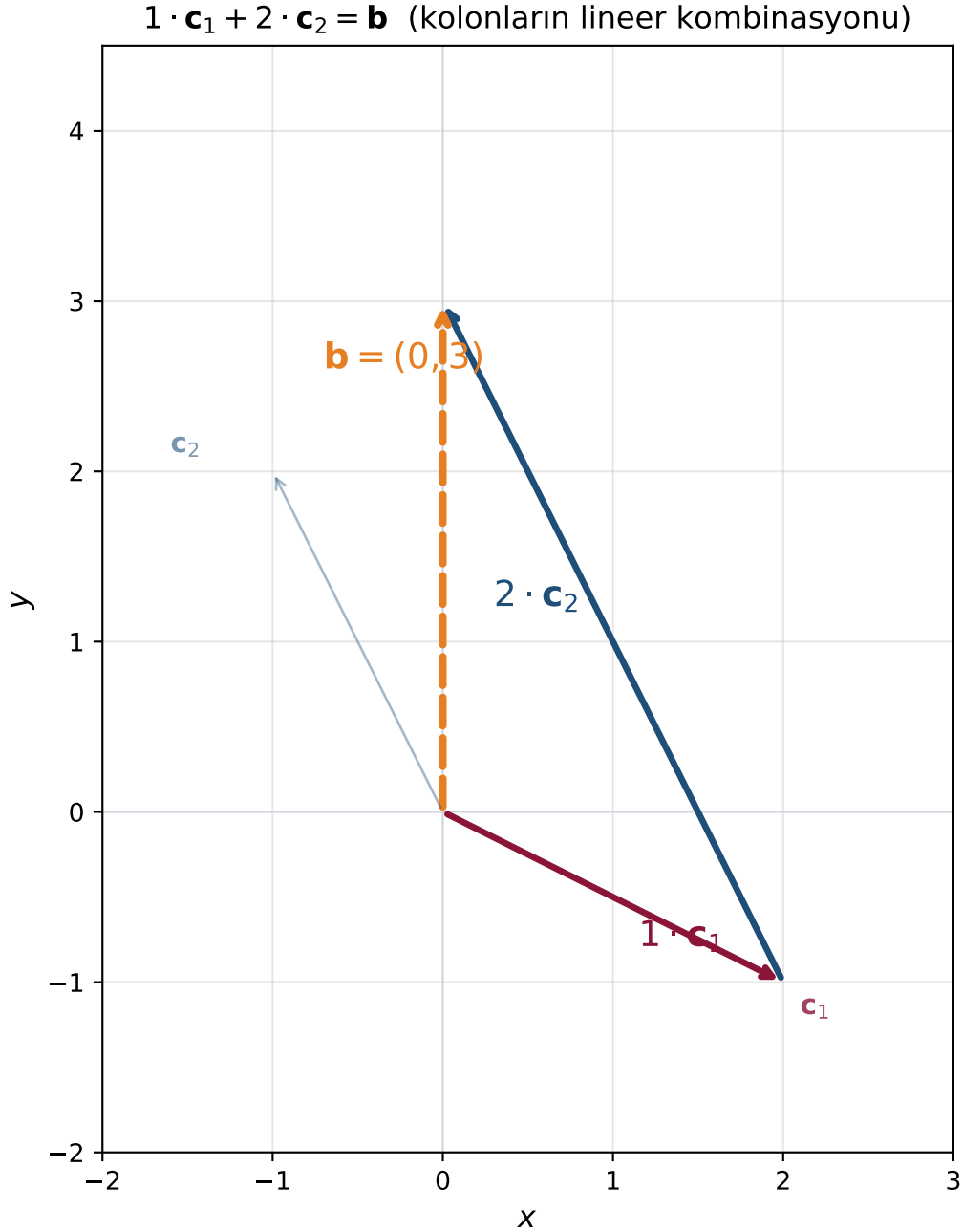
Soru artık şu: \mathbf{c}_1 ve \mathbf{c}_2 vektörlerini **hangi oranda karıştırırsan** $\mathbf{b} = (0, 3)$ vektörüne ulaşırsın?

“It’s a linear combination of the columns. That’s the most fundamental operation in the whole course.” — Strang, 9:51

Satır görünümünden bulduğumuz $x = 1, y = 2$ ile doğrulayalım:

$$1 \cdot \begin{pmatrix} 2 \\ -1 \end{pmatrix} + 2 \cdot \begin{pmatrix} -1 \\ 2 \end{pmatrix} = \begin{pmatrix} 2 \\ -1 \end{pmatrix} + \begin{pmatrix} -2 \\ 4 \end{pmatrix} = \begin{pmatrix} 0 \\ 3 \end{pmatrix} = \mathbf{b}$$

Birinci kolondan **bir tane**, ikinci kolondan **iki tane** al, topla $\rightarrow \mathbf{b}$.



Şekil 7.3: Kolon görünümü: \mathbf{c}_1 vektörünün 1 katı + \mathbf{c}_2 vektörünün 2 katı = \mathbf{b} . Vektörleri uç uca topla.

💡 Builder Notu — Forward Pass'ın Anlamı

Bir neural network katmanı $y = W @ x + b$. İki yorum:

- **Satır yorumu (dot product):** $y_i = \sum_j W_{ij} x_j$. Kod kıvamı.
- **Kolon yorumu:** $y = \sum_j x_j W_{:,j}$ — W 'nin kolonlarının x bileşenleriyle kombinasyonu. **Geometrik kıvamı.**

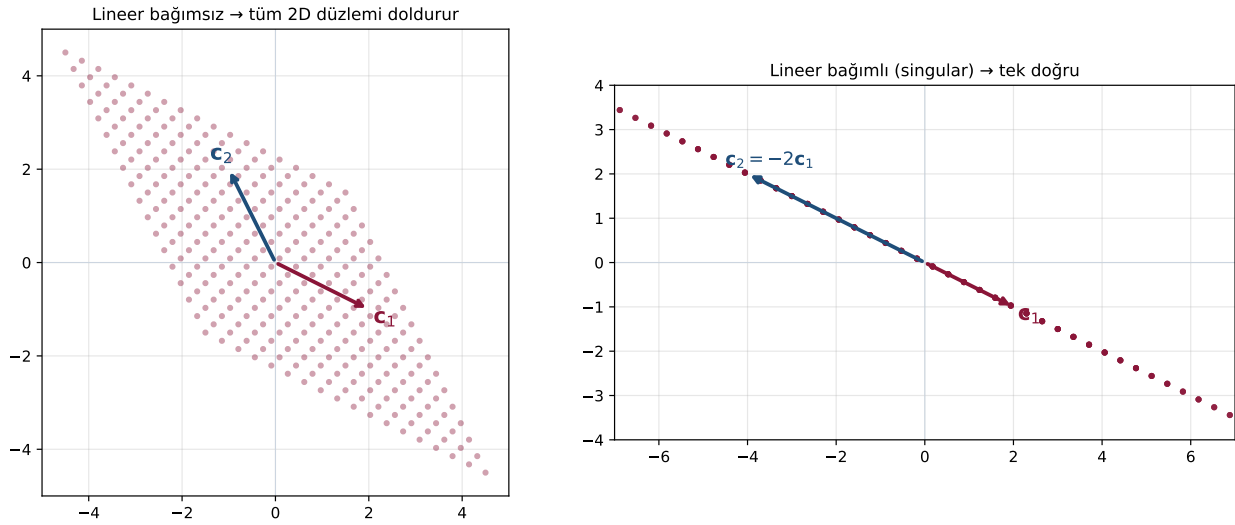
İkisi aynı sayıyı verir, ama farklı zihinsel model. Kolon yorumu sana **embedding'lerin** yorumunu (her token'in embedding'i, embedding matrisinin bir kolonudur), attention'da V kolonlarının skor ağırlıklarıyla karışmasını, LoRA'nın $\Delta W = AB^T$ ayrışımını anlatır. Satır yorumu sadece çarpımın aritmetiğini.

ML için ağırlık kolon yorumunda.

7.5 Tüm Kombinasyonlar Neyi Doldurur?

Sezgisel soru: x ve y **tüm gerçek sayıları** alabilse, $x\mathbf{c}_1 + y\mathbf{c}_2$ ne tür vektörler üretir?

Cevap: Tüm 2D düzlem. Hangi hedef vektörü \mathbf{b} seçersen seç, \mathbf{c}_1 ile \mathbf{c}_2 'nin doğru bir kombinasyonu seni oraya götürür. Bu, matrisin **non-singular** (tekil olmayan, tersi alınabilir) olması demek.



Şekil 7.4: \mathbf{c}_1 ve \mathbf{c}_2 vektörlerinin bağımsız olduğu zaman, $\alpha\mathbf{c}_1 + \beta\mathbf{c}_2$ kombinasyonları tüm düzlemi tarar (sol). Bağımlı kolonlar (sağ) sadece bir doğruyu tarar — singular durum.

Sağdaki **singular** durumda $\mathbf{c}_2 = -2\mathbf{c}_1$ olduğu için \mathbf{c}_2 “yeni bilgi” getirmiyor. Kombinasyonlar 2D'nin tamamı yerine sadece bir doğru üzerinde kalır. O doğrunun **dışındaki** her \mathbf{b} için sistemin çözümü yoktur.

7.6 Üç Boyuta Çıkış: 3×3 Sistem

İkinci örnek:

$$\begin{aligned} 2x - y &= 0 \\ -x + 2y - z &= -1 \\ -3y + 4z &= 4 \end{aligned}$$

Matris formu:

$$A = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -3 & 4 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 0 \\ -1 \\ 4 \end{pmatrix}$$

Satır görünümü 3D’de: Her denklem 3D uzayda bir **düzlem**. Üç düzlemin ortak kesişimi = çözüm.

“the row picture is getting a little hard to see” — Strang, 21:11

Strang itiraf ediyor: üç düzlemin nerede kesiştiğini gözle bulmak zor. 4D’de imkânsız. Yine kolon görünümüne geçelim.

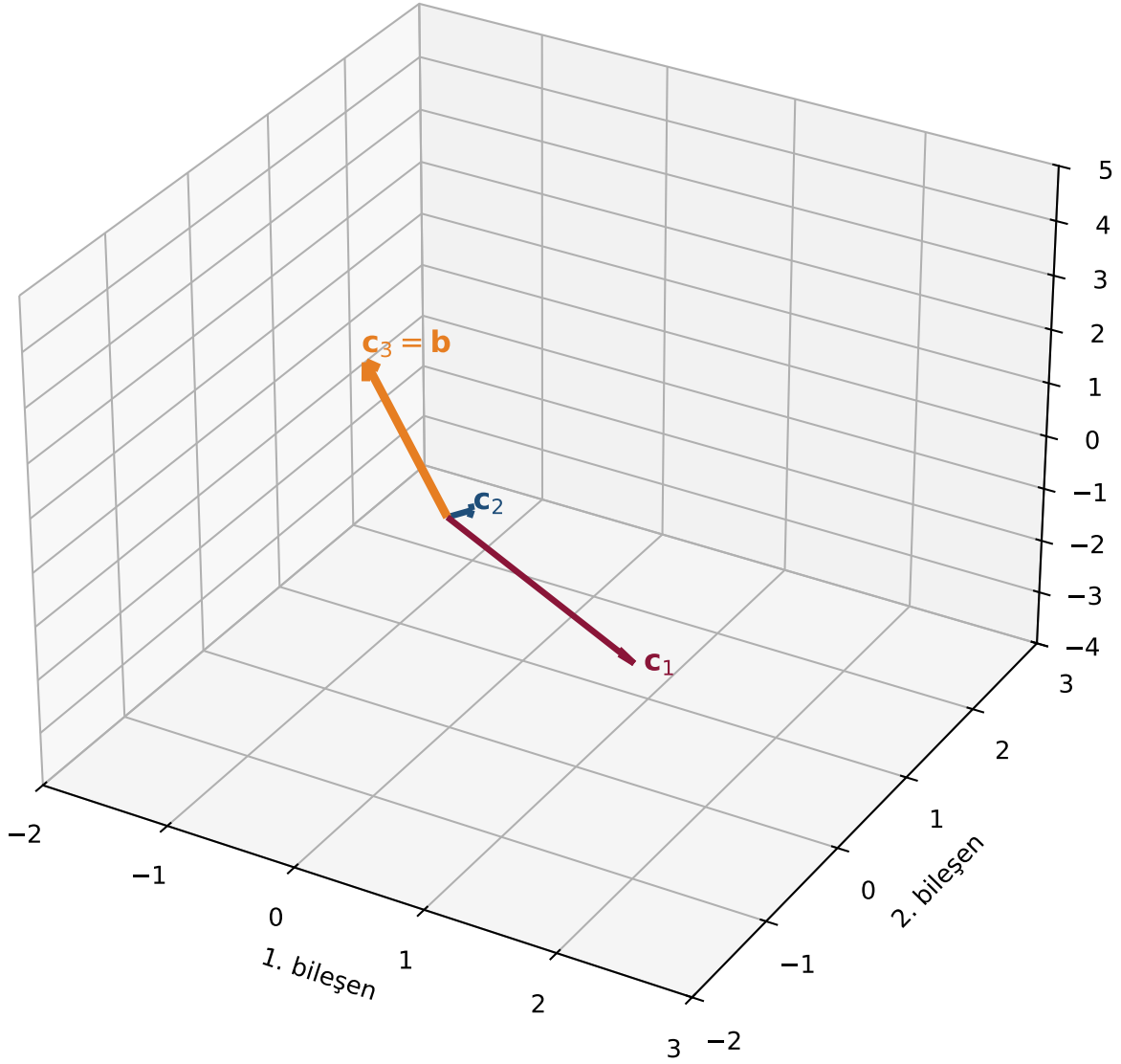
Kolon görünümü:

$$x \begin{pmatrix} 2 \\ -1 \\ 0 \end{pmatrix} + y \begin{pmatrix} -1 \\ 2 \\ -3 \end{pmatrix} + z \begin{pmatrix} 0 \\ -1 \\ 4 \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \\ 4 \end{pmatrix}$$

Strang **hile** yaptı: $\mathbf{b} = (0, -1, 4)$ tam olarak \mathbf{c}_3 ’e eşit. O zaman çözüm bir bakışta görünüyor: $x = 0, y = 0, z = 1$ — sadece üçüncü kolondan bir tane al.

“I made it work that way just so we would get an answer.” — Strang, 25:15

$x = 0, y = 0, z = 1$ — \mathbf{b} direkt \mathbf{c}_3 ile elde edilir



Şekil 7.5: 3D kolon görünümü: \mathbf{b} tam olarak \mathbf{c}_3 'e denk seçildi, çözüm $(0, 0, 1)$ — sadece 3. kolondan bir tane al.

7.7 Büyük Soru: Her \mathbf{b} İçin Çözüm Var mı?

Dersin merkez sorusu:

Cebirsel hâli: Verilen A matrisi için, her sağ taraf \mathbf{b} için $A\mathbf{x} = \mathbf{b}$ denkleminin çözümü var mı?

Kombinasyon dili: A 'nın kolonlarının lineer kombinasyonları **tüm uzayı** dolduruyor mu?

Strang vurguluyor: bu **aynı** soru, sadece farklı kelimelerle. Cevap “evet” ise A **non-singular** (full rank, invertible) — eşdeğer isimler.

Cevap “hayır” olduğunda: Kolonlar bağımlı, “tüm uzay” yerine bir alt-uzayı dolduruyorlar. O alt-uzayın dışında kalan \mathbf{b} 'lere ulaşamaz.

💡 Builder Notu — LoRA ve Düşük Rank

Strang “singular = trouble” diyor. ML’de aynı geometri **araç** oluyor.

LoRA (Low-Rank Adaptation): Bir LLM’in ağırlık matrisini fine-tune ederken, güncelleme ΔW 'nin **tam rank** olması gerekli değil. Pratikte ΔW zaten düşük ranklı — kolonları birbirine bağlı.

O yüzden ΔW yerine $\Delta W = A \cdot B^T$ yazılır; A, B ince dikdörtgen matrisler ($r \ll d$). Tipik $d = 768$, $r = 8$: $768 \times 768 = 589,824$ parametre yerine $2 \times 768 \times 8 = 12,288$ — $\sim 48\times$ tasarruf.

Strang’ın “kolonlar bağımlıysa kayıp” dediği özellik, LoRA’nın **temel sezgisi**. Aynı geometri, farklı yorum.

7.8 Yüksek Boyutlarda Düşünmek

“*Shall we take a little shot at thinking about nine dimensions?*” — Strang, 32:25

9 denklem, 9 bilinmeyen = 9 kolon, her biri 9 boyutlu. Aynı soru: 9 kolonun kombinasyonları, 9 boyutlu uzayın tamamını doldurur mu?

Çizemezsin. Strang da “I don’t pretend to do it” diyor. Ama mantığı işletebilirsin:

1. 2D: 2 bağımsız kolon → tüm düzlem.
2. 3D: 3 bağımsız kolon → tüm 3D.
3. **Genelleme:** n boyutta n bağımsız kolon → tüm n -boyutlu uzay.

Rastgele seçilirse:

“*If I picked a random matrix, ... I guarantee it would be good.*” — Strang, 33:33

Rastgele vektörlerin tam olarak bağımlı çıkması olasılığı **sıfır** (ölçü sıfır). Patolojik durumlar bilinçli kurular.

💡 Builder Notu — Yüksek Boyut Sezgisi

ML hep yüksek boyutta yaşar:

- BERT embedding: 768 boyut
- GPT-3 embedding: 12288 boyut
- LLaMA-3 70B bazı katmanlar: 8192

Çizemezsin. Ama 2D/3D’de öğrendiğin **mantığın** yüksek boyutta da geçerli olduğuna alışmalısın. Yapay zekayı anlamak büyük ölçüde “yüksek boyutta sezgisel düşünmek” demek. Strang’ın “pretend 9D” alıştırmaları bunun temeli.

7.9 Matris × Vektör: İki Hesaplama Yolu

Somut soru: Ax nasıl hesaplanır?

$$A = \begin{pmatrix} 2 & 5 \\ 1 & 3 \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

Yöntem 1: Kolon kombinasyonu (Strang'ın tercihi)

$$A\mathbf{x} = 1 \cdot \begin{pmatrix} 2 \\ 1 \end{pmatrix} + 2 \cdot \begin{pmatrix} 5 \\ 3 \end{pmatrix} = \begin{pmatrix} 12 \\ 7 \end{pmatrix}$$

Yöntem 2: Satır × vektör (dot product)

$$y_1 = (2, 5) \cdot (1, 2) = 12, \quad y_2 = (1, 3) \cdot (1, 2) = 7$$

Aynı sonuç. Ama farklı zihinsel model:

“A times x is a combination of the columns of A. That’s how I hope you will think of A times x when we need it.” — Strang, 38:24

```
Kolon kombinasyonu: [12  7]
Satır dot product : [12  7]
A @ x (numpy)      : [12  7]
```

Üç farklı kod, aynı sonuç. Modern kütüphaneler (numpy, PyTorch, JAX) bunu BLAS/cuBLAS ile çok daha hızlı yapar, ama altta yatan mantık bu.

7.10 Bu Dersin Özeti

1. Lineer denklem sistemi: $A\mathbf{x} = \mathbf{b}$.
2. Üç bakış: **satır görünümü** (doğrular/düzlemler), **kolon görünümü** (vektörler), **matris formu** (cebirsel).
3. **Kolon görünümü kritik**, çünkü yüksek boyutta da işliyor; satır görünümü çöker.
4. **Lineer kombinasyon**: $\alpha_1 \mathbf{v}_1 + \alpha_2 \mathbf{v}_2 + \dots$ — dersin en temel işlemi.
5. **Büyük soru**: Kolonların kombinasyonları tüm uzayı doldurur mu? Evet ise A **non-singular**.
6. **Singular durum**: kolonlar bağımlıysa kombinasyonlar bir alt-uzayı doldurur, bazı \mathbf{b} erişilmezdir.
7. Yüksek boyutta aynı mantık çalışır — sadece çizemezsin.
8. Ax : kolon yöntemi (ML için doğru zihin modeli) ya da dot product (kod aritmetiği).

! Tek bir cümle

Ax ifadesi, A matrisinin kolonlarının \mathbf{x} bileşenleriyle ağırlıklandırılmış **lineer kombinasyonudur** — sonraki 33 ders bu cümlemin üzerine inşa edilir.

i Soru 1: $A = ((1,2),(3,4))$ ve $x = (5,6)$ için Ax 'i (a) satır yöntemi, (b) kolon yöntemi ile hesapla.

7.11 Kontrol Soruları

(a) Satır yöntemi (dot product):

7.11 Kontrol Soruları

- $y_1 = 1 \cdot 5 + 2 \cdot 6 = 17$
- $y_2 = 3 \cdot 5 + 4 \cdot 6 = 39$
- $y = (17, 39)$

(b) Kolon yöntemi (lineer kombinasyon):

$$Ax = 5 \cdot \begin{pmatrix} 1 \\ 3 \end{pmatrix} + 6 \cdot \begin{pmatrix} 2 \\ 4 \end{pmatrix} = \begin{pmatrix} 5 \\ 15 \end{pmatrix} + \begin{pmatrix} 12 \\ 24 \end{pmatrix} = \begin{pmatrix} 17 \\ 39 \end{pmatrix}$$

İki yöntem aynı sonucu verir. ✓

i Soru 2: Aşağıdaki matrisin kolonları bağımlı mı bağımsız mı? $Ax = b$ her b için çözülebilir mi?

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 1 & 1 & 2 \end{pmatrix}$$

Cevap: Lineer **bağımlı**.

Kolonlar: $c_1 = (1, 2, 1)$, $c_2 = (2, 4, 1)$, $c_3 = (3, 6, 2)$.

Dikkat: $c_3 = c_1 + c_2$ ($1+2=3$, $2+4=6$, $1+1=2$). Üçüncü kolon yeni bilgi getirmiyor. Matris **singular**.

Her b için çözüm yok — sadece c_1 ve c_2 kombinasyonu olarak yazılabilen b vektörleri çözülebilir (kolon uzayı 3D yerine 2D'lik bir düzlem).

i Soru 3: 'A non-singular' ile 'A'nın kolonlarının kombinasyonları tüm uzayı doldurur' aynı şeyi mi söylüyor?

Cevap: Evet, aynı şey.

- "Non-singular" = $Ax = b$ her b için çözülebilir.
- Çözüm = b vektörünü A 'nın kolonlarının kombinasyonu olarak yazma.
- "Her b için çözüm var" = "her b kolonların kombinasyonu olarak yazılabilir".
- "Her b yazılabilir" = "kombinasyonlar tüm uzayı doldurur".

Bu eşdeğerlik dersin temel mesajlarından biri (Strang 27:46–30:35).

i Soru 4: (Builder) 768×768 W matrisinin 200 kolonu diğer kolonların kombinasyonu. Bu ne anlama gelir?

Cevap: W 'nin etkin rank'i en fazla $768 - 200 = 568$. W 768 boyutlu çıkış alanında çalışmasına rağmen, ulaşabildiği vektörler 568 boyutluk bir alt-uzayda yatıyor.

Sonuç:

- Temsil gücü tam 768 boyutlu değil; 200 boyutluk “kayıp” var.
- Düşük-rank temsille daha verimli ifade edilebilir: $W \approx UV^T$, $U, V \in \mathbb{R}^{768 \times r}$.
- **LoRA fine-tuning’de** tipik $r = 4-32$. $768 \times 768 = 589,824$ parametre yerine $2 \times 768 \times r$.
- $r = 32$ için $\approx 12 \times$ bellek tasarrufu.

LoRA’nın temel sezgisi: fine-tuning için tam rank gerekli değil, çünkü ağırlık güncellemeleri zaten düşük ranklıdır.

7.12 Egzersizler

Egzersiz 1. Aşağıdaki sistemi (a) satır yöntemi (çizerek), (b) kolon yöntemi (vektörleri çizerek), (c) matris formunda yaz:

$$\begin{aligned} 3x + 2y &= 7 \\ x - y &= 1 \end{aligned}$$

Egzersiz 2. $A = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}$ matrisinin kolonlarının kombinasyonlarının tüm 2D düzlemi doldurduğunu göster. Rastgele bir \mathbf{b} seç, \mathbf{x} ’i bul.

Egzersiz 3. $A = \begin{pmatrix} 1 & 2 \\ 2 & 4 \end{pmatrix}$ matrisinin kolonları hangi alt-uzayı tarar? Hangi \mathbf{b} vektörleri için çözüm var, hangileri için yok? (İpucu: $\mathbf{c}_2 = 2\mathbf{c}_1$.)

Egzersiz 4. (Python — sayısal doğrulama) 5×5 rastgele bir sistem kur, çözümünü hem kolon hem dot-product yöntemiyle elle hesapla ve `np.linalg.solve` ile karşılaştır.

```
x          : [ 1.639 -0.162 -0.229  0.988  0.169]
|Ax_kolon - b|: 3.2487068343022356e-16
|Ax_satir - b|: 3.2487068343022356e-16
```

Egzersiz 5. A matrisi $n \times n$, kolonları lineer bağımsız. $A\mathbf{x} = \mathbf{0}$ denkleminin kaç çözümü vardır? (Bu sorunun cevabı Ders 6’da **sıfır uzayı** $N(A)$ kavramının temelini oluşturacak.)


7.13 Sonraki Ders İçin Hazırlık

Ders 2: Matrislerle Eliminasyon

Bu derste sezgi seviyesinde kaldık. Genel bir 3×3 sistem verilse hâlâ sistematik çözüm yöntemimiz yok — kolon görünümünde sezgi bazı özel durumlarda çalışır, genel durumda yetmez.

“Every bit of software, too — production, large-scale software would solve the equations.” — Strang, 25:35

Ders 2’de **Gauss eliminasyonunun** mekaniği, hata durumları (sıfır pivot, satır deęiş tokuşu), ve **LU ayrışımına** giden yol. `numpy.linalg.solve` arka planda tam olarak bunu yapar.

 Ders 2 öncesi yapılacak

- Egzersizleri çöz — özellikle **4** (Python doğrulama) ve **5** (sıfır çözüm = bağımsız kolonlar).
- Bu cümleyi içselleştir: “*Ax* ifadesi, *A*’nın kolonlarının lineer kombinasyonudur.”
- Strang’in PS1’inden 2-3 problem dene (OCW 18.06SC sayfası).

7.14 Anahtar Kavramlar (Cheat Sheet)

Kavram	Tanım	Strang’da
Lineer denklem sistemi	$A\mathbf{x} = \mathbf{b}$ formundaki denklemler kümesi	1:02
Matris	Sayıların dikdörtgen düzeni; lineer dönüşümü temsil eder	2:30
Vektör	n bileşenli sıralı sayı listesi; geometrik nesne	3:21
Satır görünümü	Her denklemini geometrik doğru/düzlem olarak görme	4:20
Kolon görünümü	$A\mathbf{x}$ ’i kolonların kombinasyonu olarak görme	8:39
Lineer kombinasyon	$\alpha_1 \mathbf{v}_1 + \alpha_2 \mathbf{v}_2 + \dots$ — dersin temel işlemi	9:51
Non-singular matris	Her \mathbf{b} için $A\mathbf{x} = \mathbf{b}$ ’nin çözümü olan matris	30:04
Singular matris	Kolonları bağımlı, bazı \mathbf{b} için çözüm yok	32:09
Lineer bağımsızlık	Hiçbir vektörün diğerlerinin kombinasyonu olarak yazılamaması	34:00 (örtük)
Dot product	$\mathbf{u} \cdot \mathbf{v} = \sum_i u_i v_i$	37:52

7.15 ML Bağlantıları Özeti

💡 5 köprü

1. $W_{\mathbf{x}} = W$ 'nin kolonlarının kombinasyonu → Her nöral ağ katmanı bu işlem. Embedding'ler, attention'da V , MLP'lerin gizli temsilleri — hepsi kolon yorumuyla anlam kazanır.
2. **Kolon uzayı** $C(W)$ → Katmanın “ulaşabildiği” çıkışlar. Bottleneck/encoder katmanı tasarımı bu kavramla.
3. **Lineer bağımlılık** → **Rank** → **LoRA** → Düşük rank kabul edip $\Delta W = AB^T$ ile parametrize etme; LoRA'nın temel sezgisi.
4. **Singular = null vektör** → **dropout / pruning** → Bazı yön sinyallerini sıfırlayan operasyonlar.
5. **Yüksek boyut sezgisi** → Tüm ML 768D+ boyutlarda yaşar. 2D/3D'den 768D'ye taşınmış sezgi alışkanlığı, bu dersin en kalıcı kazancı.

❗ Tek bir şey alıp gideceksen

$A\mathbf{x}$ ifadesi A 'nın kolonlarının kombinasyonudur. Bu cümle, sonraki 33 ders, ve neredeyse tüm modern ML'in zihinsel iskeletidir. Üç kez okuduğunda “evet, kolonların kombinasyonu” diye içinde otursa, bu dersin amacına ulaşmış olur.

8 Matrislerle Eliminasyon

Gauss eliminasyonu, pivot, E matrisleri ve $EA = U$

i Bölüm bilgisi

- **Strang'ın videosu:** [YouTube — Lecture 2: Elimination with Matrices](#) (≈ 47 dk)
- **OCW sayfası:** [MIT 18.06SC — Lecture 2](#)
- **Okuma süresi:** ≈ 40 dk

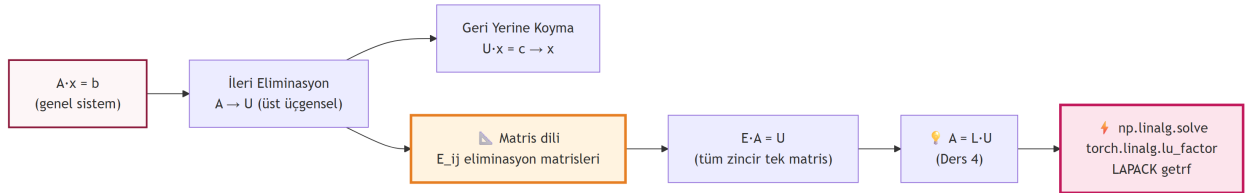
8.1 Bu Derste Ne Var?

Ders 1'de “lineer sistemin geometrisi ne?” sorusunu işledik. Şimdi pratik soru: **bu sistemi nasıl çözeriz?** Strang'ın açış cümlesi sert — “*every software package solves equations*” yöntemiyle, yani **eliminasyon** ile. numpy, MATLAB, LAPACK, PyTorch — hangi sayısal kütüphaneye baksan, $Ax = b$ tipi sistemleri çözerken arka planda yapılan budur. Determinant değil; eliminasyon.

Üç ana parça:

1. **İleri eliminasyon** — A 'yı sistematik satır işlemleriyle üst üçgensel U matrisine dönüştür.
2. **Geri yerine koyma** — Üst üçgensel $Ux = c$ 'den çözümü alttan yukarı oku.
3. **Matris dili** — Her satır işlemini bir eliminasyon matrisi E_{ij} olarak yaz; tüm zincir $EA = U$ olur.

Üçüncü parça en kritik olanı; Strang dersin yarısını buna ayırıyor. Sebep: **LU ayrışımı** (Ders 4), **ters matris** (Ders 3) ve modern numerik LA hep bu E matrisleri üzerine kurulu.



Şekil 8.1: Gauss eliminasyonu üç katmanlı: hesaplama → matris dili → ML köprüleri.

💡 Builder Notu — Eliminasyon Her Yerde

Gauss eliminasyonunu elle yazmazsın. Ama her yerde:

- `np.linalg.solve(A, b)` arka planda LAPACK `getrf` çağırır — partial pivoting’li Gauss eli-

minasyonu.

- **Linear regression normal equations, ridge regression, Newton's method, Kalman filter** — hepsi solve çağırır.
- **Matris çarpımının “satır mı, kolon mu” sezgisi** (bu derste netleşecek) attention mekanizmasından dense layer'a kadar her yerde lazım.
- **Çarpımın birleşmeliği (associative)** Flash Attention'ın temeli — aynı sonuç, farklı parantez, dramatik bellek tasarrufu.

8.2 Problem — Strang'ın 3×3 Sistemi

Somut sistem:

$$\begin{aligned}x + 2y + z &= 2 \\3x + 8y + z &= 12 \\4y + z &= 2\end{aligned}$$

(Üçüncü denklemden x katsayısı 0.) Matris formunda:

$$A = \begin{pmatrix} 1 & 2 & 1 \\ 3 & 8 & 1 \\ 0 & 4 & 1 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 2 \\ 12 \\ 2 \end{pmatrix}$$

Strateji: A 'yı satır işlemleriyle **üst üçgensel** U 'ya dönüştür. Üst üçgensel sistemler kolay çözülür — alttan yukarı, her satırda tek bilinmeyen kalır.

8.3 İleri Eliminasyon — Adım Adım

Hedef: A 'nın köşegen altındaki tüm girdileri sıfırlamak.

Adım 1: (2,1) sıfırla. İlk pivot = sol üst köşe **1**. İkinci satırdaki 3'ü silmek için **3 × satır 1'i, satır 2'den çıkar.**

- Yeni satır 2 = $(3, 8, 1) - 3 \cdot (1, 2, 1) = (0, 2, -2)$
- Çarpan $\ell_{21} = 3$.

$$\begin{pmatrix} \boxed{1} & 2 & 1 \\ 3 & 8 & 1 \\ 0 & 4 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 2 & 1 \\ 0 & 2 & -2 \\ 0 & 4 & 1 \end{pmatrix}$$

Adım 2: (3,1) sıfırla. Üçüncü satırın 1. elemanı zaten 0. $\ell_{31} = 0$, geç.

Adım 3: (3,2) sıfırla. İkinci pivot = satır 2'deki yeni **2**. Üçüncü satırdaki 4'ü silmek için **2 × satır 2'yi, satır 3'ten çıkar.**

- Yeni satır 3 = $(0, 4, 1) - 2 \cdot (0, 2, -2) = (0, 0, 5)$
- Çarpan $\ell_{32} = 2$.

$$U = \begin{pmatrix} \boxed{1} & 2 & 1 \\ 0 & \boxed{2} & -2 \\ 0 & 0 & \boxed{5} \end{pmatrix}$$

Pivotlar: 1, 2, 5.

A (orijinal)	$E_{21}A$ ($3-r1 \rightarrow r2$)	$U = E_{32}E_{21}A$ ($2-r2 \rightarrow r3$)																											
<table border="1" style="width: 100%; text-align: center; border-collapse: collapse;"> <tr><td style="width: 33.33%; height: 40px;">1</td><td style="width: 33.33%;">2</td><td style="width: 33.33%;">1</td></tr> <tr><td style="height: 40px;">3</td><td>8</td><td>1</td></tr> <tr><td style="height: 40px;">0</td><td>4</td><td>1</td></tr> </table>	1	2	1	3	8	1	0	4	1	<table border="1" style="width: 100%; text-align: center; border-collapse: collapse;"> <tr><td style="width: 33.33%; height: 40px;">1</td><td style="width: 33.33%;">2</td><td style="width: 33.33%;">1</td></tr> <tr><td style="height: 40px;">0</td><td style="background-color: #f9cb9c;">2</td><td style="background-color: #f9cb9c;">-2</td></tr> <tr><td style="height: 40px;">0</td><td>4</td><td>1</td></tr> </table>	1	2	1	0	2	-2	0	4	1	<table border="1" style="width: 100%; text-align: center; border-collapse: collapse;"> <tr><td style="width: 33.33%; height: 40px;">1</td><td style="width: 33.33%;">2</td><td style="width: 33.33%;">1</td></tr> <tr><td style="height: 40px;">0</td><td style="background-color: #f9cb9c;">2</td><td style="background-color: #f9cb9c;">-2</td></tr> <tr><td style="height: 40px;">0</td><td>0</td><td style="background-color: #f9cb9c;">5</td></tr> </table>	1	2	1	0	2	-2	0	0	5
1	2	1																											
3	8	1																											
0	4	1																											
1	2	1																											
0	2	-2																											
0	4	1																											
1	2	1																											
0	2	-2																											
0	0	5																											

Şekil 8.2: Eliminasyonun üç anlık görüntüsü: orijinal $A \rightarrow E_{21}$ sonrası $\rightarrow E_{32}$ sonrası U . Renkli hücreler değişimi gösterir; sıfır pozisyonları beyaza döner.

“This matrix is great. It gave me three pivots, I didn’t have to do anything special, I just followed the rules.” — Strang, 10:00

Pivot Notu:

- Üç pivot, hiçbiri sıfır değil $\rightarrow A$ invertible (Ders 3 detay).
- Determinant = pivotların çarpımı = $1 \times 2 \times 5 = 10$ (Strang: “I never want to know” — determinanta nadiren bakacağız, ama burada bedava çıktı).
- Pivot işaretleri matrisin **pozitif tanımlılığ** hakkında ipucu (Ders 25).

8.4 Başarısızlık Modları — Pivot Sıfır

Geçici başarısızlık — Satır takasıyla kurtuluş:

Pivot pozisyonunda 0 var. Aşağıdaki bir satırla yer değiştir, eliminasyona devam.

“I will never be heard to utter those words, ‘0 pivot’. But if there’s a 0 in the pivot position, I would try to exchange for a lower equation.” — Strang, 11:25

Strang’in örneği: A ’daki 8 yerine 6 olsaydı, ikinci pivot $6 - 3 \cdot 2 = 0$ olurdu. Satır 2 ile satır 3’ü takas et — kurtuluş.

Tam başarısızlık — Kurtuluş yok:

8 Matrislerle Eliminasyon

Pivot 0 ve aşağıdaki tüm girdiler de 0. O zaman matris **singular** — ya sonsuz çözüm ya hiç çözüm. Ders 1'deki $\mathbf{c}_3 = \mathbf{c}_1 + \mathbf{c}_2$ örneğindeki durum.

💡 Builder Notu — Numerik Stabilite ve Partial Pivoting

Pratikte (numpy, LAPACK) sadece “sıfır mı değil mi” diye bakılmaz. **Çok küçük pivot** da problemlidir — float aritmetikte küçük sayıyla bölme = patlayan hata.

Gerçek implementasyonlar **partial pivoting** yapar: her adımda kolondaki en büyük (mutlak değer) elemanı pivot olarak seçer, ona göre satır takası. `np.linalg.solve` arka planda bunu yapar; sen “küçük sayılar pivotta olmasın” diye düşünmek zorunda değilsin. Ama bilmek lazım: pivot seçimi sayısal sağlamlık için kritik.

ML'de bu sezgi şuralarda karşına çıkar: - **Batch normalization** — küçük varyansa bölmenin patlamasını önlemek için $\sqrt{\sigma^2 + \epsilon}$. - **L2 norm, RMSNorm** — aynı motivasyon. - **Cholesky factorization** — pozitif tanımlı matrisler için “pivoting-free” alternatif.

8.5 Geri Yerine Koyma

Eliminasyon $A'yı U'ya$ dönüştürdü. Sağ tarafa da aynı işlemleri uygulayalım. **Augmented matrix** $[A | \mathbf{b}]$:

$$[A | \mathbf{b}] = \left(\begin{array}{ccc|c} 1 & 2 & 1 & 2 \\ 3 & 8 & 1 & 12 \\ 0 & 4 & 1 & 2 \end{array} \right)$$

Aynı çarpanları $\mathbf{b}'ye$ uygula:

- Yeni $b_2 = 12 - 3 \cdot 2 = 6$
- Yeni $b_3 = 2 - 2 \cdot 6 = -10$

$$[U | \mathbf{c}] = \left(\begin{array}{ccc|c} 1 & 2 & 1 & 2 \\ 0 & 2 & -2 & 6 \\ 0 & 0 & 5 & -10 \end{array} \right)$$

Alttan yukarı çöz:

- $5z = -10 \Rightarrow z = -2$
- $2y - 2z = 6 \Rightarrow 2y + 4 = 6 \Rightarrow y = 1$
- $x + 2y + z = 2 \Rightarrow x + 2 - 2 = 2 \Rightarrow x = 2$

Çözüm: $(x, y, z) = (2, 1, -2)$.

Orijinalde doğrula: $-2 + 2 - 2 = 2 \checkmark$ $-6 + 8 - 2 = 12 \checkmark$ $-4 - 2 = 2 \checkmark$

8.6 Matris Dili — E_{ij} ile Yazmak

Strang dersin ikinci yarısında devasa bir adım: **yukarıdaki tüm satır işlemlerini matris çarpımı olarak yaz.**

“All the key ideas get expressed as matrix operations, not as words.” — Strang, 2:16

İki bakış lazım:

Bakış 1 (Ders 1’den): Matris \times kolon vektör = matrisin **kolonlarının** lineer kombinasyonu.

Bakış 2 (YENİ): Satır vektör \times matris = matrisin **satırlarının** lineer kombinasyonu.

$$(1 \ 2 \ 7) \begin{pmatrix} - & \mathbf{r}_1 & - \\ - & \mathbf{r}_2 & - \\ - & \mathbf{r}_3 & - \end{pmatrix} = 1\mathbf{r}_1 + 2\mathbf{r}_2 + 7\mathbf{r}_3$$

! Anahtar Prensiptir

Soldan çarpma \rightarrow satır işlemleri.

Sağdan çarpma \rightarrow kolon işlemleri.

Eliminasyon satır işlemleri olduğu için, eliminasyon matrisleri A ’nın **soluna** çarpılır.

8.7 Eliminasyon Matrisleri

E_{21} — **İlk adım:** “Satır 1 değişmesin, satır 3 değişmesin, satır 2 \rightarrow satır 2 $-$ 3·satır 1.”

$$E_{21} = \begin{pmatrix} 1 & 0 & 0 \\ -3 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Her satırı “satır vektör \times matris” mantığıyla yorumla:

- Satır 1 $(1, 0, 0)$: $1\mathbf{r}_1 \rightarrow$ değişmedi ✓
- Satır 2 $(-3, 1, 0)$: $-3\mathbf{r}_1 + \mathbf{r}_2 \rightarrow$ istediğimiz ✓
- Satır 3 $(0, 0, 1)$: $\mathbf{r}_3 \rightarrow$ değişmedi ✓

Genel kural: E_{ij} identity matristen (i, j) konumuna **çarpanın negatifi** yazılarak elde edilir.

E_{32} — **İkinci adım:**

$$E_{32} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{pmatrix}$$

Tüm zincir: $E_{32}(E_{21}A) = U$.

8 Matrislerle Eliminasyon

E (toplu eliminasyon):

$$\begin{bmatrix} 1. & 0. & 0. \\ -3. & 1. & 0. \\ 6. & -2. & 1. \end{bmatrix}$$

E @ A:

$$\begin{bmatrix} 1. & 2. & 1. \\ 0. & 2. & -2. \\ 0. & 0. & 5. \end{bmatrix}$$

E32 @ (E21 @ A):

$$\begin{bmatrix} 1. & 2. & 1. \\ 0. & 2. & -2. \\ 0. & 0. & 5. \end{bmatrix}$$

Fark: 0.0

8.8 Birleşme Yasası — Parantezleri Kaydır

“You can move the parentheses... this is highly important. So many proofs of main facts come from just moving the parentheses.” — Strang, 35:36

$$E_{32}(E_{21}A) = (E_{32}E_{21})A$$

İki taraf aynı. Bu yüzden tek bir $E = E_{32}E_{21}$ matrisi tanımlayıp tüm eliminasyonu bir hamlede uygulayabiliriz.

DİKKAT — Değişme yasası yok: $AB \neq BA$ genelde. Eliminasyonda sıra önemli — önce E_{21} (sağda, ilk uygulanır), sonra E_{32} (solda, sonra uygulanır). Sırayı karıştırırsan farklı matris çıkar.

💡 Builder Notu — Flash Attention ve Associative Law

Attention formülü:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right) V$$

Sequence uzunluğu n büyüdükçe QK^T matrisi $n \times n$ — bellek katlar ($n = 8192$ için ~ 256 MB tek bir attention map için).

Flash Attention (Dao et al., 2022) associative law ve softmax'ın çevrimiçi (online) güncellenebilirliğini kullanarak çarpım sırasını yeniden düzenler — ara matrisi hiç oluşturmadan sonucu hesaplar. Aynı matematik, farklı parantez, çok daha az HBM erişimi.

Bu derste öğrendiğin “parantezleri kaydır” sezgisi, transformer optimizasyonunun temel taşı.

8.9 Permütasyon Matrisleri

Pivot sıfır geldiyse satır takası gerek. Bu da bir matris çarpımı:

$$P = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad P \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} c & d \\ a & b \end{pmatrix}$$

(Soldan çarpma → satır işlemi → satırları takas.)

Sağdan çarptığında **kolonları** takas eder. Pratik implementasyonlarda her zaman P, L, U üçlüsü saklanır:

$$PA = LU$$

Bu Ders 4'ün ana sonucu olacak — numerik LA'nın en yaygın ayrışımı.

8.10 Ters Matrise İlk Bakış

Strang sonu sonraki dersin köprüsüyle bağlıyor. E_{21} ne yaptıysa, hangi matris geri alır?

$$E_{21} = \begin{pmatrix} 1 & 0 & 0 \\ -3 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \rightarrow E_{21}^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Sadece işaret değişti: $-3 \rightarrow +3$ (eliminasyonun tersi: “3 katı çıkardıysan, 3 katı ekle”).

Ana özellik: $E_{21}^{-1} \cdot E_{21} = I$.

$A = LU$ — Ders 4'ün önizlemesi

A	L (alt üçgensel, köşegen = 1)	U (üst üçgensel, pivotlar = 1, 2, 5)
1	1	1
2	0	2
1	0	1
3	3	0
8	1	2
1	0	-2
0	0	0
4	2	0
1	1	5

Şekil 8.3: $A = LU$ ayrışımı: L alt üçgensel (eliminasyon çarpanları), U üst üçgensel (pivot + üst tarafı). Köşegen L 'de 1'ler, U 'da pivotlar.

8 Matrislerle Eliminasyon

$L @ U =$

$$\begin{bmatrix} 1. & 2. & 1. \\ 3. & 8. & 1. \\ 0. & 4. & 1. \end{bmatrix}$$

$A =$

$$\begin{bmatrix} 1. & 2. & 1. \\ 3. & 8. & 1. \\ 0. & 4. & 1. \end{bmatrix}$$

Fark: 0.0

8.11 Bu Dersin Özeti

1. **Gauss eliminasyonu:** $Ax = b$ 'yi sistematik çözmenin algoritması.
2. **İleri eliminasyon:** $A \rightarrow U$ (üst üçgensel).
3. **Pivot:** her kolonda referans eleman, sıfır olamaz.
4. **Başarısızlık modları:** pivot sıfır \rightarrow satır takası (kurtuluş) ya da singular (kurtuluş yok).
5. **Geri yerine koyma:** $Ux = c$ 'yi alttan yukarı çöz.
6. **Matris dili:** Her satır işlemi bir E_{ij} . Soldan çarpma = satır işlemi.
7. **Birleşme yasası:** $E_{32}(E_{21}A) = (E_{32}E_{21})A = EA = U$.
8. **Permütasyon:** P satır takası matrisi $\rightarrow PA = LU$ (Ders 4).
9. **Ters matris:** her E için E^{-1} — Ders 3'ün konusu.

! Tek bir cümle

Eliminasyon, A 'yı bir dizi E_{ij} matrisiyle soldan çarparak üst üçgensel U 'ya dönüştürür: $EA = U$.
Sonraki derslerde bu denklem $A = LU$ olarak ters yüz olacak ve numerik LA'nın omurgası çıkacak.

8.12 Kontrol Soruları

i Soru 1: $A = ((2,1),(6,8))$ için ileri eliminasyon — pivot, çarpan, U .

İlk pivot: 2. Çarpan: $\ell_{21} = 6/2 = 3$.
Yeni satır 2: $(6, 8) - 3 \cdot (2, 1) = (0, 5)$.

$$U = \begin{pmatrix} 2 & 1 \\ 0 & 5 \end{pmatrix}$$

Pivotlar: 2, 5. Determinant = 10.

i Soru 2: Aynı örnek için E_{21} matrisini yaz ve $E_{21} A = U$ olduğunu doğrula.

$$E_{21} = \begin{pmatrix} 1 & 0 \\ -3 & 1 \end{pmatrix}$$

(2, 1) konumunda çarpanın negatifi -3 . Çarpım:

$$E_{21}A = \begin{pmatrix} 1 & 0 \\ -3 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 \\ 6 & 8 \end{pmatrix} = \begin{pmatrix} 2 & 1 \\ 0 & 5 \end{pmatrix} = U \checkmark$$

i Soru 3: Aşağıdaki matriste eliminasyon problem çıkarır — satır takası yardım eder mi?

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 1 \\ 0 & 1 & 5 \end{pmatrix}$$

İlk adım: $\ell_{21} = 2$, satır 2 $\rightarrow (0, 0, -5)$.

$$\rightarrow \begin{pmatrix} 1 & 2 & 3 \\ 0 & 0 & -5 \\ 0 & 1 & 5 \end{pmatrix}$$

(2, 2) pozisyonunda **0** var. Ama altında satır 3'te 1 var. Satır 2 \leftrightarrow satır 3 takas:

$$\rightarrow \begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & 5 \\ 0 & 0 & -5 \end{pmatrix} = U$$

Pivotlar 1, 1, -5 . Matris singular değil — pivot sırasını düzeltmek gerekti. Buna **partial pivoting** denir.

i Soru 4: (Builder) `np.linalg.solve(A, b)` ne yapar — neden `np.linalg.inv(A) @ b` yazmıyoruz?

İki ayrı sebep:

1. **Sayısal sağlamlık.** `solve` Gauss + partial pivoting kullanır. `inv` tüm tersi hesaplar — çok daha fazla işlem ve hata birikimi. Sadece $Ax = b$ 'yi çözüyorsan, tüm tersi bilmeye gerek yok.
2. **Hız.** `solve` $\approx n^3/3$ flop. `inv` ardından `inv @ b` $\approx n^3$ flop. Yaklaşık $3\times$ yavaş.

Genel kural: $A^{-1}b$ yazmak istediğin her yerde `np.linalg.solve(A, b)` yaz. Newton's method, normal equations, Cholesky tabanlı işlemler, hepsinde geçerli. PyTorch: `torch.linalg.solve`.

8.13 Egzersizler

Egzersiz 1. Aşağıdaki sistemde ileri eliminasyon + geri yerine koyma yap. U , çarpanlar, x .

$$\begin{aligned} 2x + y &= 5 \\ 4x + 7y &= 11 \end{aligned}$$

8 Matrislerle Eliminasyon

Egzersiz 2. 3×3 sistemini eliminasyonla çöz; E_{21}, E_{31}, E_{32} matrislerini yaz; $E = E_{32}E_{31}E_{21}$ çarpımını bul.

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 2 & 3 & 4 \\ 1 & 0 & 2 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 6 \\ 20 \\ 8 \end{pmatrix}$$

Egzersiz 3. Hangi A matrisinde eliminasyon başarısız olur ve satır takası da kurtaramaz? En basit 2×2 örneği yaz.

Egzersiz 4. (Python — LU görselleştirme) Strang'ın örneğinde `scipy.linalg.lu` ile P, L, U ayrışımını üret ve pivotları kontrol et.

```
x = [ 2.  1. -2.]
Pivotlar (diag U): [3.          4.          0.83333333]
P =
[[0. 0. 1.]
 [1. 0. 0.]
 [0. 1. 0.]]
L =
[[ 1.          0.          0.          ]
 [ 0.          1.          0.          ]
 [ 0.33333333 -0.16666667  1.          ]]
U =
[[3.          8.          1.          ]
 [0.          4.          1.          ]
 [0.          0.          0.83333333]]
```

Egzersiz 5. Çarpanların işaretlerini değiştirip E_{ij}^{-1} matrislerini ters çevirerek $L = E_{21}^{-1}E_{31}^{-1}E_{32}^{-1}$ matrisini bul (Strang'ın 3×3 örneği için). Sonuç alt üçgensel olmalı. **Bu L , LU ayrışımının L' 'idir.**

8.14 Sonraki Ders İçin Hazırlık

Ders 3: Matris Çarpımı ve Ters Matrisler

- Matris çarpımının dört bakışı: kolon \times kolon, satır \times satır, hücre, blok.
- Ters matrisin tam tanımı ve Gauss–Jordan ile hesabı.
- $(AB)^{-1} = B^{-1}A^{-1}$ — sıra ters çevriliyor.

⚠ Ders 3 öncesi yapılacak

- Bu dersin egzersizlerini çöz, özellikle **4** (LU görselleştirme) ve **5** (L 'i ters eliminasyonla inşa).
- Bu cümleyi içselleştir: “Eliminasyon A 'yı E matrisleri ile çarparak U 'ya dönüştürür: $EA = U$.”
- Python'da `scipy.linalg.lu` ile birkaç farklı matris dene; pivot pozisyonlarını anlamaya çalış.

8.15 Anahtar Kavramlar (Cheat Sheet)

Kavram	Tanım	Strang'da
Eliminasyon (Gauss)	A 'yı satır işlemleriyle U 'ya dönüştürme	0m30
Pivot	Her kolonda referans eleman, sıfır olamaz	4m20
Çarpan ℓ_{ij}	Eliminasyon adımının oranı	4m50
Üst üçgensel U	Köşegen altı sıfır olan matris	9m25
İleri eliminasyon	$A \rightarrow U$ dönüşümü	5m18
Geri yerine koyma	$U\mathbf{x} = \mathbf{c}$ çözümü, alttan yukarı	17m50
Eliminasyon matrisi E_{ij}	(i, j) konumunu sıfırlayan elementer matris	29m08
Birleşme yasası	$A(BC) = (AB)C$ — parantez kaydırılabilir	35m36
Permütasyon P	Satır takası matrisi	37m35
Ters matris E^{-1}	$E \cdot E^{-1} = I$	44m00
Singular	Pivotu sıfırlanan ve takasla kurtulmayan matris	14m00

8.16 ML Bağlantıları Özeti

💡 7 köprü

1. **np.linalg.solve = Gauss eliminasyonu** → Linear regression normal equations, Newton's method, Kalman filter — hepsi bu rutini çağırır.
2. **LU decomposition (Ders 4)** → Eliminasyonu saklamanın yolu. `torch.linalg.lu_factor` — aynı A için birden çok \mathbf{b} çözüleceğinde cache.
3. **Matris çarpımı sıra önemli** → $AB \neq BA$. Attention'da QK^T vs K^TQ farklı; backprop'ta gradient zincirinde sıra hayati.
4. **Birleşme yasası → Flash Attention** → Aynı sonuç, farklı parantez; HBM erişimi $O(n^2)$ 'den $O(n)$ 'a iner.
5. **Partial pivoting → numerik sağlamlık** → Batch/layer norm, ϵ ile bölme, Cholesky alternatifi — hepsi aynı motivasyon.
6. **Singular matrisler** → Ill-conditioned sistemler, gradient explosion/vanishing, düşük-rank kayıplar; Ders 1'in LoRA hikayesinin numerik versiyonu.
7. **Permütasyon matrisleri P** → Token shuffling, batch reordering, dropout maskelerinde implicit P .

8 Matrislerle Eliminasyon

! Tek bir şey alıp gideceksen

$EA = U$. Eliminasyon = sol-çarpan E matrislerinin zinciri = A 'yı üst üçgensel yapma. Bu denklem, sayısal lineer cebirin temel taşı.

9 Matris Çarpımı ve Ters Matrisler

Beş bakış, bir sonuç — ve Gauss–Jordan ile A^{-1}

i Bölüm bilgisi

- **Strang’in videosu:** [YouTube — Lecture 3: Multiplication and Inverse Matrices](#) (≈47 dk)
- **OCW sayfası:** [MIT 18.06SC — Lecture 3](#)
- **Okuma süresi:** ≈40 dk

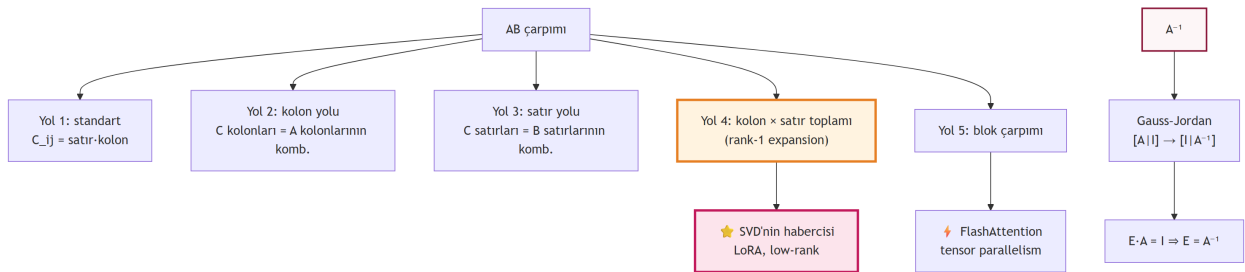
9.1 Bu Derste Ne Var?

Ders 2’de eliminasyon matrislerini sol-çarpan olarak kullandık: $EA = U$. Strang şimdi geri dönüp diyor ki matris çarpımının **beş ayrı yolu** var — hepsi aynı sonucu verir, ama her biri farklı bir açı kazandırır.

İki büyük tema:

1. **Matris çarpımının beş yolu** — Standart (sıra × kolon), kolon yolu, sıra yolu, kolon × sıra toplamı (rank-1 ayrışımı), blok çarpımı.
2. **Ters matris** — Tanımı, var olma koşulu (üç farklı bakış), ve Gauss–Jordan ile hesabı.

İkinci kısım, Ders 2’deki “her eliminasyon adımının bir tersi var” sezgisini formalleştirir.



Şekil 9.1: Matris çarpımının beş yolu ve ters matrisin Gauss–Jordan ile inşası.

💡 Builder Notu — Modern ML’de Beş Yol Ne İşe Yarar?

- **Yol 4 (kolon × sıra = rank-1 toplamı)** → SVD’nin sezgisel temeli (Ders 29). $A = \sum_k \sigma_k \mathbf{u}_k \mathbf{v}_k^T$ — her terim rank-1. **LoRA** bu temele dayanır: büyük ΔW yerine $A \cdot B^T$ (ince matrisler).
- **Yol 5 (blok çarpımı)** → Distributed GPU matmul, **FlashAttention** chunking, tensor parallelism.

Büyük matrisleri belleğe sığdırmanın yolu.

- **Singular vs invertible** → ML’de ill-conditioned problem teşhisi; regularization (L2/weight decay) tam olarak “matrisi singular olmaktan uzaklaştırma”.
- $(AB)^{-1} = B^{-1}A^{-1}$ → Backprop’ta zincirleme türev sırasının matematiksel temeli; normalizing flows / invertible networks.

9.2 Matris Çarpımı — Beş Yol

İki matris: $C = AB$. Boyutlar: $A (m \times n)$, $B (n \times p)$, $C (m \times p)$. **A’nın kolon sayısı = B’nin satır sayısı** olmazsa çarpım tanımsız.

“There are many ways you can do it, and they all give the same answer. And they’re all important.”
— Strang, 0:24

9.2.1 Yol 1: Standart (sattır × kolon)

$$C_{ij} = \sum_{k=1}^n a_{ik} \cdot b_{kj}$$

Klasik okul yöntemi. Hesaplama için pratik; sezgi vermez.

9.2.2 Yol 2: Kolon Yolu

C ’nin her kolonu $A \cdot (B$ ’nin ilgili kolonu):

$$AB = A \begin{pmatrix} | & | & \dots & | \\ \mathbf{b}_1 & \mathbf{b}_2 & \dots & \mathbf{b}_p \\ | & | & \dots & | \end{pmatrix} = \begin{pmatrix} | & | & \dots & | \\ A\mathbf{b}_1 & A\mathbf{b}_2 & \dots & A\mathbf{b}_p \\ | & | & \dots & | \end{pmatrix}$$

C ’nin her kolonu, A ’nın kolonlarının bir lineer kombinasyonudur — katsayılar B ’nin ilgili kolonundan.

9.2.3 Yol 3: Satır Yolu

C ’nin her satırı (A ’nın ilgili satırı) $\cdot B$:

C ’nin her satırı, B ’nin satırlarının bir lineer kombinasyonudur — katsayılar A ’nın ilgili satırından.

Yol 2 + 3 sezgisi: Soldan ne koyarsan C ’nin **satırlarını** belirler; sağdan ne koyarsan **kolonlarını**. Ders 2’deki “soldan = satır işlemi” kuralının başka bir yüzü.

9.2.4 Yol 4: Kolon × Satır Toplamı (rank-1 expansion)

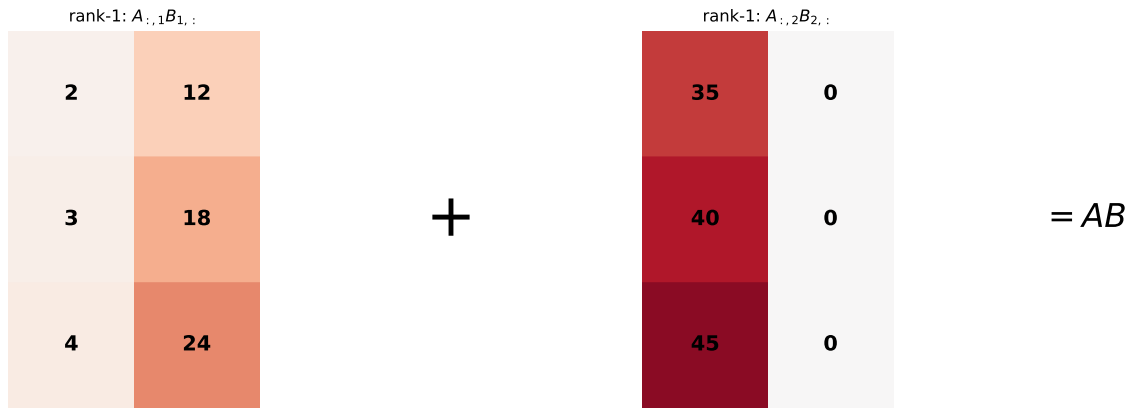
Bir **kolon** ($m \times 1$) ile bir **satır** ($1 \times p$) çarp $\rightarrow m \times p$ matris. Örnek:

$$\begin{pmatrix} 2 \\ 3 \\ 4 \end{pmatrix} (1 \ 6) = \begin{pmatrix} 2 & 12 \\ 3 & 18 \\ 4 & 24 \end{pmatrix}$$

Bu matrisin **tüm kolonları** $(2, 3, 4)^T$ doğrultusunda, **tüm satırları** $(1, 6)$ doğrultusunda — bu **rank-1 matris** demektir.

Çarpımı rank-1 toplamı olarak yaz:

$$AB = \sum_{k=1}^n (A\text{'nin } k. \text{ kolonu}) \cdot (B\text{'nin } k. \text{ satırı})$$



Şekil 9.2: Yol 4: AB çarpımı, n tane rank-1 matrisin toplamı. Bu sezgi doğrudan SVD'ye (Ders 29) ve LoRA'ya götürür.

$$\begin{aligned} R1 + R2 &= \\ &[[37. \ 12.]] \\ &[[43. \ 18.]] \\ &[[49. \ 24.]] \\ A @ B &= \\ &[[37. \ 12.]] \\ &[[43. \ 18.]] \\ &[[49. \ 24.]] \end{aligned}$$

“All those rows lie on the line through $(1, 6)$. All the columns lie on the line through $(2, 3, 4)$. So this is a really minimal matrix.” — Strang, 17:18

9.2.5 Yol 5: Blok Çarpımı

Matrisleri uygun şekilde bloklara böl, blok-blok çarp:

$$\begin{pmatrix} A_1 & A_2 \\ A_3 & A_4 \end{pmatrix} \begin{pmatrix} B_1 & B_2 \\ B_3 & B_4 \end{pmatrix} = \begin{pmatrix} A_1B_1 + A_2B_3 & A_1B_2 + A_2B_4 \\ A_3B_1 + A_4B_3 & A_3B_2 + A_4B_4 \end{pmatrix}$$

Her blok pozisyonunda “sıra blokları × kolon blokları” toplamı — büyük çarpımın aynı kuralı, sadece elemanlar yerine alt-matrisler.

💡 Builder Notu — Blok Çarpımı = AI Altyapısı

- **GPU memory:** Matris GPU’ya sığmıyorsa blok blok çarp.
- **FlashAttention (Dao 2022):** QK^T ’yi blok blok hesapla, sadece **softmax çıktısını** HBM’e yaz. Aynı matematik, $O(n)$ bellek erişimi.
- **Tensor parallelism:** Büyük weight matrisi GPU’lar arasında bloklara dağıtılır; her GPU bir bloğu tutar.
- **Mixed precision:** Bazı blokları FP32, bazılarını FP16 hesapla; blok çarpım kuralı birleştirir.

9.3 Ters Matris — Tanım ve Var Olma Koşulu

A kare ($n \times n$). A ’nın **tersi** A^{-1} , eğer varsa:

$$A \cdot A^{-1} = I \quad \text{ve} \quad A^{-1} \cdot A = I$$

Square matrisler için bedava bonus: sol ters = sağ ters (rectangular için değil, Ders 33).

İki sınıf:

- **Invertible (non-singular):** A^{-1} var.
- **Singular:** A^{-1} yok.

9.4 Ters Olmayan Matrisler — Üç Bakış

$$A = \begin{pmatrix} 1 & 3 \\ 2 & 6 \end{pmatrix}$$

Bakış 1: Determinant. $\det A = 1 \cdot 6 - 3 \cdot 2 = 0 \rightarrow$ singular.

Bakış 2: Kolon bağımlılığı. $c_2 = 3c_1$. İki kolon aynı doğrultuda — kolon uzayı bir doğru. AX ’in kolonları hep o doğru üzerinde, identity üretilemez.

Bakış 3: $Ax = 0$ için sıfırdan farklı x (en güçlü test):

$$A \begin{pmatrix} 3 \\ -1 \end{pmatrix} = \begin{pmatrix} 3 - 3 \\ 6 - 6 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Çelişki: Eğer A^{-1} olsaydı, her iki yanı soldan A^{-1} ile çarp: $X = A^{-1} \cdot \mathbf{0} = \mathbf{0}$. Ama $X = (3, -1) \neq \mathbf{0}$.
Çelişki $\rightarrow A^{-1}$ yok.

“The matrix can’t have an inverse if some combination of the columns gives nothing.” — Strang, 30:02

! Singular Testin Özü

A^{-1} var $\Leftrightarrow Ax = \mathbf{0}$ 'in tek çözümü $x = \mathbf{0}$ 'dır.

9.5 Gauss–Jordan: A^{-1} 'i Bulma Algoritması

$$A = \begin{pmatrix} 1 & 3 \\ 2 & 7 \end{pmatrix} \quad (\det = 1, \text{invertible})$$

Strateji: A^{-1} 'i bir matris olarak yaz: kolonları $\mathbf{x}_1, \mathbf{x}_2$. Yol 2'ye göre $A \cdot A^{-1} = I$ demek:

- $A\mathbf{x}_1 = \mathbf{e}_1 = (1, 0)^T$
- $A\mathbf{x}_2 = \mathbf{e}_2 = (0, 1)^T$

İki sistem, aynı A , farklı sağ taraflar. Jordan'ın fikri: **ikisini birlikte çöz** — augmented matris $[A \mid I]$.

Augmented:

$$[A \mid I] = \left[\begin{array}{cc|cc} 1 & 3 & 1 & 0 \\ 2 & 7 & 0 & 1 \end{array} \right]$$

Aşama 1 (ileri eliminasyon): $2 \cdot r_1 \rightarrow r_2$:

$$\rightarrow \left[\begin{array}{cc|cc} 1 & 3 & 1 & 0 \\ 0 & 1 & -2 & 1 \end{array} \right]$$

Aşama 2 (geri eliminasyon): $3 \cdot r_2 \rightarrow r_1$:

$$\rightarrow \left[\begin{array}{cc|cc} 1 & 0 & 7 & -3 \\ 0 & 1 & -2 & 1 \end{array} \right]$$

Sol taraf I oldu, sağ taraf:

$$A^{-1} = \begin{pmatrix} 7 & -3 \\ -2 & 1 \end{pmatrix}$$

Doğrula: $A \cdot A^{-1} = I \checkmark$.

9 Matris Çarpımı ve Ters Matrisler

Manuel A^{-1} :

$$\begin{bmatrix} 7 & -3 \\ -2 & 1 \end{bmatrix}$$

numpy A^{-1} :

$$\begin{bmatrix} 7 & -3 \\ -2 & 1 \end{bmatrix}$$

$A @ A^{-1} =$

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

9.6 Gauss–Jordan Neden Çalışıyor?

Her eliminasyon adımı = soldan bir E matrisi ile çarpma. Tüm adımları birleştir: $E = E_k \cdots E_2 E_1$.

Augmented matris $[A \mid I]$ üzerine etki:

$$E \cdot [A \mid I] = [EA \mid EI] = [EA \mid E]$$

Algoritma sonunda sol taraf I oldu, yani $EA = I$. Demek ki:

$$E = A^{-1}$$

Ve sağ tarafta kalan da $E = A^{-1}$. Algoritma “şanstan” çalışmıyor; tüm eliminasyonu yapan kombo matris, tam olarak A^{-1} .

“If E times A is the identity, then E must be the inverse of A . And on the right-hand side, where we just smartly tucked on the identity, it’s turning into A inverse.” — Strang, 45:35

9.7 Bu Dersin Özeti

1. **Matris çarpımının beş yolu** — Standart, kolon, satır, rank-1 toplamı, blok.
2. **Ters matris:** $A \cdot A^{-1} = A^{-1} \cdot A = I$ (square için).
3. **Tersi olmayan matrisler** üç bakış: determinant 0, kolonlar bağımlı, $Ax = \mathbf{0}$ için sıfırdan farklı x .
4. **Gauss–Jordan:** $[A \mid I] \rightarrow [I \mid A^{-1}]$.
5. **Neden çalışır:** Tüm eliminasyon = E kombo; $EA = I \Rightarrow E = A^{-1}$.

! Tek bir cümle

Matris çarpımının beş yolu farklı bakışlar verir; ters matris $[A \mid I] \rightarrow [I \mid A^{-1}]$ ile bulunur — çünkü tüm eliminasyonun kombosu E , $EA = I$ koşulunu sağladığında $E = A^{-1}$.

9.8 Soru 2: $A = ((4,8),(2,4))$ invertible mi? Üç bakışla göster.

Det: $4 \cdot 4 - 8 \cdot 2 = 0 \rightarrow$ singular.

Kolon bağımlılığı: $c_2 = 2c_1$. Aynı doğrultu \rightarrow identity üretilemez.

$Ax = 0$ testi: $x = (2, -1)^T$:

$$Ax = \begin{pmatrix} 8 - 8 \\ 4 - 4 \end{pmatrix} = \mathbf{0}$$

Sıfırdan farklı x var \rightarrow singular.

9.8 Soru 3: Gauss-Jordan ile $A = ((2,1),(5,3))$ tersini bul.

Augmented:

$$\left[\begin{array}{cc|cc} 2 & 1 & 1 & 0 \\ 5 & 3 & 0 & 1 \end{array} \right]$$

$l_{21} = 5/2, r_2 \rightarrow r_2 - (5/2)r_1$:

$$\left[\begin{array}{cc|cc} 2 & 1 & 1 & 0 \\ 0 & 1/2 & -5/2 & 1 \end{array} \right]$$

Pivotları normalize ($r_2 \cdot 2$, sonra $r_1 \rightarrow r_1 - r_2$, sonra $r_1/2$):

$$\rightarrow \left[\begin{array}{cc|cc} 1 & 0 & 3 & -1 \\ 0 & 1 & -5 & 2 \end{array} \right]$$

$$A^{-1} = \begin{pmatrix} 3 & -1 \\ -5 & 2 \end{pmatrix}$$

Doğru! $A \cdot A^{-1} = I \checkmark$.

i Soru 4: $(AB)^{-1} = B^{-1}A^{-1}$ — neden sıra ters? Backprop'ta nerede?

Niye sıra ters:

$$(AB)(B^{-1}A^{-1}) = A(BB^{-1})A^{-1} = A \cdot I \cdot A^{-1} = I \checkmark$$

Eğer $A^{-1}B^{-1}$ deneseydin: $A(BA^{-1})B^{-1}$ — burada BA^{-1} sadeleşmez ($BA \neq AB$ genelde).

Backprop'ta: Forward $\mathbf{x} \rightarrow A\mathbf{x} \rightarrow B(A\mathbf{x}) = (BA)\mathbf{x}$. Invertible flow modellerinde “geri yön” $(BA)^{-1} = A^{-1}B^{-1}$.

Zincirleme türevde aynı sıra: $\frac{\partial L}{\partial \mathbf{x}} = \frac{\partial L}{\partial \mathbf{y}} \cdot \frac{\partial \mathbf{y}}{\partial \mathbf{x}}$ — Jacobian sırası çıktıdan girdiye. **Backward pass'in matematiksel iskeleti.**

9.9 Egzersizler

Egzersiz 1. $A = ((1, 2), (3, 4))$, $B = ((5, 6), (7, 8))$ çarpımını **Yol 4** (rank-1 toplamı) ile yaz; iki rank-1 matris.

Egzersiz 2. $A = ((1, 2, 3), (2, 4, 6), (3, 6, 9))$ invertible mi? Soru 2'deki üç bakışla göster. (İpucu: tüm satırlar $(1, 2, 3)$ 'ün katları.)

Egzersiz 3. (Python) `np.linalg.inv` ile `np.linalg.solve` performansını karşılaştır.

```
solve : 2.25 ms
inv@b : 133.00 ms (~ 59.0x yavas)
|fark| : 2.01e-11
```

Egzersiz 4. 4×4 bir matrisi 2×2 bloklara böl, blok çarpımı ile sonucu bul, standart yöntemle karşılaştır.

Egzersiz 5. $A = ((1, 2), (0, 3))$, $B = ((2, 0), (1, 1))$. A^{-1} , B^{-1} Gauss–Jordan ile bul, sonra $(AB)^{-1}$ 'i iki yoldan: (a) AB 'nin doğrudan tersi, (b) $B^{-1}A^{-1}$ çarpımı. Aynı çıkmalı.

9.10 Sonraki Ders İçin Hazırlık

Ders 4: $A = LU$ Faktörizasyonu

- Her invertible matris için $A = LU$ ayrışımı.
- L (alt üçgensel) = ters eliminasyon adımlarının çarpımı.
- U (üst üçgensel) = ileri eliminasyonun sonucu.
- Sayısal LA'nın en yaygın ayrışımı; `scipy.linalg.lu`, `torch.linalg.lu_factor` arka planı.

⚠ Ders 4 öncesi yapılacak

- Egzersizleri çöz, özellikle 3 (solve vs inv) ve 4 (blok çarpımı).
- Python'da `scipy.linalg.lu` ile birkaç matris dene.
- Ana cümleyi tekrar oku.

9.11 Anahtar Kavramlar (Cheat Sheet)

Kavram	Tanım	Strang'da
Yol 1: Standart	$C_{ij} = \sum_k a_{ik} b_{kj}$	0:33
Yol 2: Kolon	C kolonları = A kolonlarının komb.	6:11
Yol 3: Satır	C satırları = B satırlarının komb.	10:09
Yol 4: rank-1 toplamı	$AB = \sum_k \mathbf{a}_k^{\text{col}} \mathbf{b}_k^{\text{row}}$	12:06
Yol 5: Blok çarpımı	Bloklara böl, blok-blok çarp	18:34
Rank-1 matris	Tüm kolonlar/satırlar tek doğrultuda	17:18
Ters matris	$AA^{-1} = A^{-1}A = I$ (square)	21:22
Singular test	$A\mathbf{x} = \mathbf{0}$ için $\mathbf{x} \neq \mathbf{0}$ varsa singular	28:00
Gauss-Jordan $(AB)^{-1}$	$[A I] \rightarrow [I A^{-1}]$ $= B^{-1}A^{-1}$ (sıra ters)	36:13 47:00

9.12 ML Bağlantıları Özeti

💡 6 köprü

- Yol 4 (rank-1 toplamı)** → SVD/LoRA → Her matris en az r rank-1 matrisin toplamı; LoRA bu temele dayanır.
- Yol 5 (blok çarpımı)** → FlashAttention / tensor parallelism → Büyük matrisleri belleğe sığdırmanın matematiksel iskeleti.
- $(AB)^{-1} = B^{-1}A^{-1}$ → **backprop sırası** → Normalizing flows, invertible NNs, zincirleme türev.
- np.linalg.inv vs np.linalg.solve** → Inverse hesaplamak nadiren gerekli; $A^{-1}\mathbf{b}$ yerine solve(A, b) (3× hızlı, sağlam).
- Singular = ill-conditioned** → Gradient descent yakınsamıyorsa condition number yüksek; weight decay = “matrisi singular’dan uzaklaştır”.
- Kolon uzayı sezgisi** → **embedding spaces** → Word/sentence embeddings = vektör uzayında “anlam yönleri”; ulaşılabilir vs ulaşılmaz vektör kavramı.

! Tek bir şey alıp gideceksen

Matris çarpımının beş yolu farklı bakışlardır, aynı sonucu verir. Gauss-Jordan $[A | I]$ 'dan $[I | A^{-1}]$ 'a giderek A^{-1} 'i kurar — çünkü eliminasyon kombo matrisi E , $EA = I$ koşulunda tam olarak A^{-1} 'dir.

10 A = LU Faktörizasyonu

Eliminasyonun doğru matris formu — ve numerik LA'nın temel taşı

i Bölüm bilgisi

- **Strang'ın videosu:** [YouTube — Lecture 4: A = LU Factorization](#) (≈48 dk)
- **OCW sayfası:** [MIT 18.06SC — Lecture 4](#)
- **Okuma süresi:** ≈40 dk

10.1 Bu Derste Ne Var?

Ders 3'te ters matrisi ve Gauss–Jordan'ı gördük. Ders 4'ün iki teması:

1. **Kısa hazırlık:** $(AB)^{-1} = B^{-1}A^{-1}$ ve $(A^T)^{-1} = (A^{-1})^T$.
2. **Asıl mesele — $A = LU$:** Ders 2'de $EA = U$ gördük. Şimdi ters çevirip $A = LU$ yazıyoruz. Sebep: L sezgisel olarak güzel bir matris (eliminasyon çarpanları doğrudan yerleşir), E ise değildir (çarpanlar birbirine karışır).

“ $A = LU$ is the most basic factorization of a matrix.” — Strang, 7:13

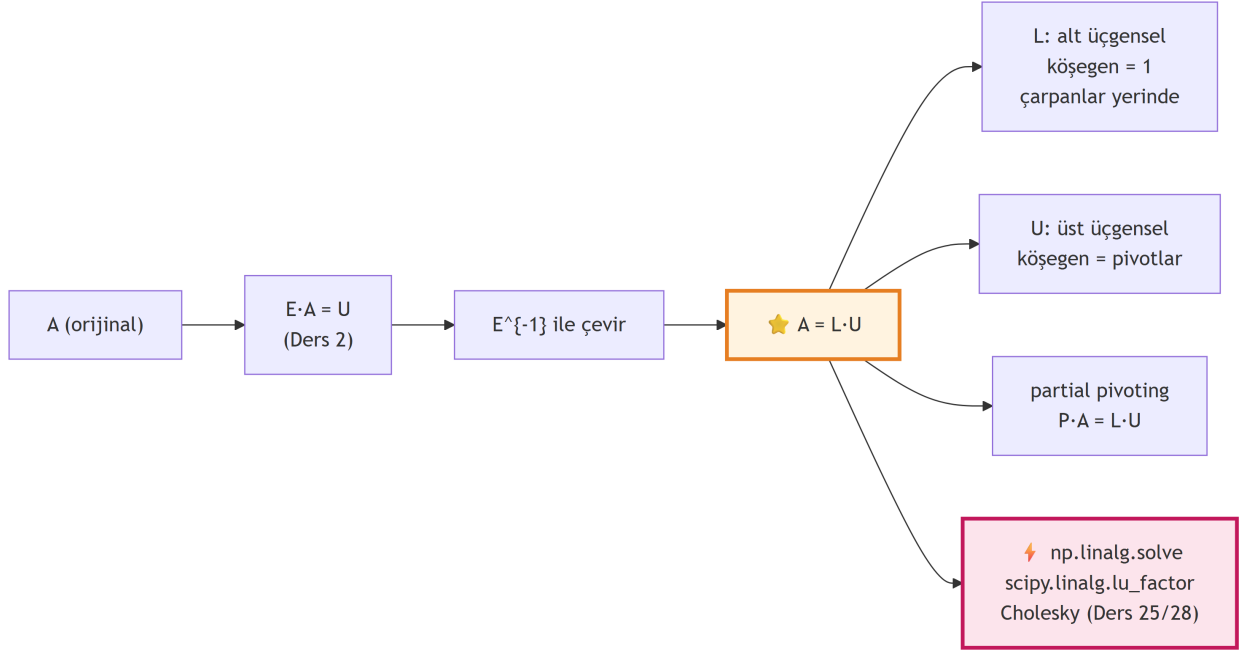
💡 Builder Notu — Neden LU Her Yerde

- **np.linalg.solve(A, b)** arka planda: $A = LU$ faktörize et $\rightarrow Ly = b$ (forward sub) $\rightarrow Ux = y$ (back sub).
- **Aynı A ile birden çok b çöz:** LU'yu **bir kez** hesapla, sonra her **b** için sadece $O(n^2)$ öde, $O(n^3)$ değil. Kalman filter, Newton's method, batch çözümlerin altyapısı.
- **Cholesky decomposition** (Ders 25/28) LU'nun simetrik pozitif tanımlı özel hali — Bayesian inference, Gaussian processes, normal equations.
- **Maliyet** $\approx n^3/3$ — GPU TFLOPS sayıları bu hesapla anlam kazanır.

10.2 $(AB)^{-1} = B^{-1}A^{-1}$ — Çarpımın Tersisi

$$(AB)(B^{-1}A^{-1}) = A(BB^{-1})A^{-1} = A \cdot I \cdot A^{-1} = I \checkmark$$

Sıra **ters dönüyor**. Strang'ın metaforu:



Şekil 10.1: Eliminasyon $\rightarrow A = LU$. L çarpanları, U pivotları taşır; partial pivoting gerektiğinde $PA = LU$.

“You take off your shoes, you take off your socks; the good way to invert that process is socks back on first, then shoes.” — Strang, 3:21

Çorap önce, ayakkabı sonra; tersi: ayakkabı önce çıkar, çorap sonra.

10.3 $(A^T)^{-1} = (A^{-1})^T$ — Transpoz ve Ters Yer Değiştirir

$AA^{-1} = I$ 'yı transpozla:

$$(AA^{-1})^T = I^T \implies (A^{-1})^T A^T = I$$

Demek ki $(A^T)^{-1} = (A^{-1})^T$. Transpoz ve ters alma işlemleri **commute eder**.

ML'de: Normal equations $(A^T A)^{-1}$ simetri korur. Kovaryans Σ^{-1} simetrik kalır. Backprop'ta forward $y = Wx \rightarrow$ gradient W^T ile zincirlenir.

10.4 $A = LU$ — Basit 2x2 Örneği

Ders 2'den: $EA = U$. Tersini al:

$$A = E^{-1}U \iff A = LU \quad (L := E^{-1})$$

Strang'in örneği:

$$A = \begin{pmatrix} 2 & 1 \\ 8 & 7 \end{pmatrix}$$

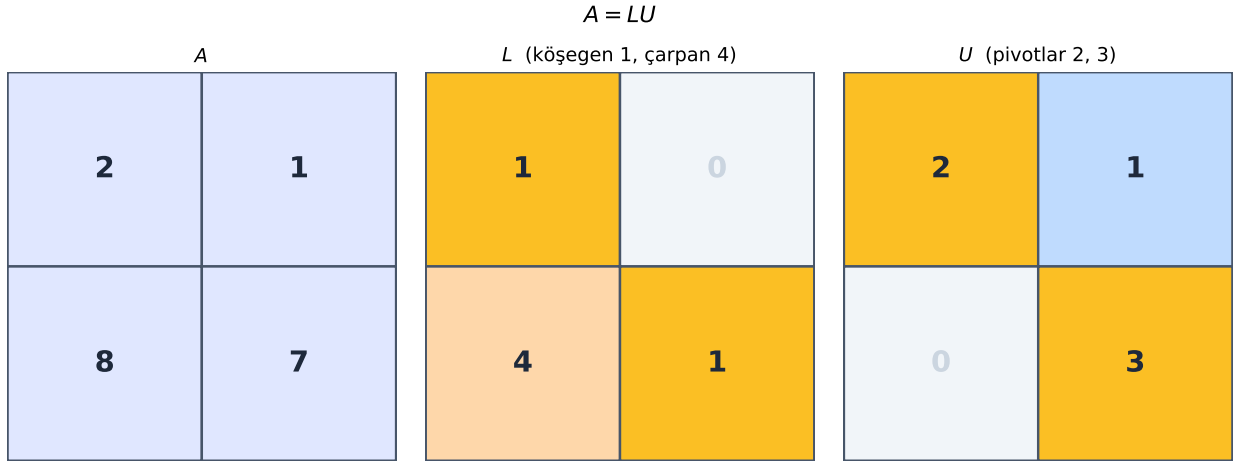
Çarpan $\ell_{21} = 8/2 = 4$:

$$E_{21} = \begin{pmatrix} 1 & 0 \\ -4 & 1 \end{pmatrix}, \quad E_{21}A = \begin{pmatrix} 2 & 1 \\ 0 & 3 \end{pmatrix} = U$$

$L = E_{21}^{-1}$ — sadece işaret değişir ($-4 \rightarrow +4$):

$$L = \begin{pmatrix} 1 & 0 \\ 4 & 1 \end{pmatrix}, \quad LU = \begin{pmatrix} 1 & 0 \\ 4 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 \\ 0 & 3 \end{pmatrix} = \begin{pmatrix} 2 & 1 \\ 8 & 7 \end{pmatrix} = A \checkmark$$

L 'nin (2,1) konumundaki 4, eliminasyondaki çarpan ℓ_{21} 'in kendisi. Rastlantı değil.



Şekil 10.2: $A = LU$ ayrışımı: L alt üçgensel (köşegen 1, çarpanlar yerinde), U üst üçgensel (köşegen = pivotlar).

10.5 $A = LDU$ — Pivotları Köşegene Çekme

L ve U arasında bir asimetri var. Pivotları ayrı bir köşegen matrise çek:

$$U = \begin{pmatrix} 2 & 1 \\ 0 & 3 \end{pmatrix} = \underbrace{\begin{pmatrix} 2 & 0 \\ 0 & 3 \end{pmatrix}}_D \underbrace{\begin{pmatrix} 1 & 1/2 \\ 0 & 1 \end{pmatrix}}_{U'}$$

$$A = LDU'$$

Hem L hem U' köşegende 1'ler taşır; D pivotları. **Simetrik matrisler için** $A = LDL^T$ olacak (Ders 25, Cholesky'nin temeli).

10.6 3×3 — Çarpanlar L’de Doğrudan Oturuyor

3×3’te üç eliminasyon adımı: $E_{32}E_{31}E_{21}A = U$, yani $A = (E_{21}^{-1}E_{31}^{-1}E_{32}^{-1})U = LU$.

Strang’ın büyük gözlemi: **Çarpanlar L’ye doğrudan yerleşir, hiçbir matris çarpımı gerekmez.**

“The multipliers go directly into L. You can throw away A as you create LU.” — Strang, 23:52

$E = E_{32}E_{21}$	— (3,1)’de **10** karışıklığı									
<table border="1" style="border-collapse: collapse; width: 100px; height: 100px;"> <tr><td style="text-align: center;">1</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td></tr> <tr><td style="text-align: center;">-2</td><td style="text-align: center;">1</td><td style="text-align: center;">0</td></tr> <tr><td style="text-align: center;">10</td><td style="text-align: center;">-5</td><td style="text-align: center;">1</td></tr> </table>	1	0	0	-2	1	0	10	-5	1	
1	0	0								
-2	1	0								
10	-5	1								

$L = E_{21}^{-1}E_{32}^{-1}$	— çarpanlar (2, 5) yerlerinde									
<table border="1" style="border-collapse: collapse; width: 100px; height: 100px;"> <tr><td style="text-align: center;">1</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td></tr> <tr><td style="text-align: center;">2</td><td style="text-align: center;">1</td><td style="text-align: center;">0</td></tr> <tr><td style="text-align: center;">0</td><td style="text-align: center;">5</td><td style="text-align: center;">1</td></tr> </table>	1	0	0	2	1	0	0	5	1	
1	0	0								
2	1	0								
0	5	1								

Şekil 10.3: 3×3 için çarpanlar L’ye doğrudan yerleşir, E’de karışır. $\ell_{21}=2$ ve $\ell_{32}=5$ olsun: E’nin (3,1) hücresinde $2 \cdot 5 = 10$ karışıklığı çıkar, L’de hiç olmaz.

Neden L’de karışmıyor? Ters sırada çarpıyoruz; üst matris (E_{21}^{-1}) zaten satır 2’yi etkilemiş, alt matris (E_{32}^{-1}) sadece satır 3’e dokunuyor. Üst-alt etkileşim olmadan, her çarpan kendi yerine yerleşir.

💡 Builder Notu — LU Cache

`scipy.linalg.lu_factor(A)` L ve U’yu compact formatta döndürür. Sonra `lu_solve(LU_factors, b)` her **b** için $O(n^2)$ — çok daha ucuz. Newton’s method, recursive Bayesian filter, PDE çözücüler, hepsi bu pattern.

10.7 Eliminasyon Maliyeti — $n^3/3$

İlk adım: $(n-1) \times (n-1)$ blok değişir $\rightarrow \sim n^2$ işlem. İkinci: $\sim (n-1)^2$. Toplam:

$$T(n) = \sum_{k=1}^n k^2 \approx \int_0^n x^2 dx = \frac{n^3}{3}$$

“It’s about one third of n cubed. That’s the magic operation count.” — Strang, 36:35

Sağ taraf maliyeti: \mathbf{b} için $\sim n^2$ — ihmal edilebilir.

Boyut	LU faktörize	Her \mathbf{b} için çözüm
$n = 100$	$\sim 10^5$ flop	$\sim 10^4$ flop
$n = 1000$	$\sim 10^9$ flop (~ 1 sn)	$\sim 10^6$ flop (~ 1 ms)
$n = 10000$	$\sim 10^{12}$ flop (dk)	$\sim 10^8$ flop (~ 100 ms)

n 'yi $2\times$ alırsan, maliyet $8\times$. Bu, neden modern ML'in tam matris terslerinden kaçındığını açıklıyor: $1B \times 1B$ parametre için 10^{27} flop — imkansız. Onun yerine SGD, Adam, KFAC gibi yaklaşık metotlar.

10.8 Permütasyon Matrisleri ve $PA = LU$

Pivot pozisyonunda sıfır gelirse satır takası gerek. **Permütasyon matrisi** $P = \text{identity}$ 'nin satırları yer değişmiş hali. Soldan çarpma \rightarrow satır takası.

Genel form: $PA = LU$. `scipy.linalg.lu(A)` her zaman (P, L, U) üçlüsü döner.

Numerik sağlamlık (partial pivoting): Pratikte LAPACK her adımda kolondaki en büyük (mutlak) elemanı pivot olarak seçer. Küçük pivota bölme = float hata patlaması.

$n \times n$ permütasyon sayısı = $n!$:

- $n = 3 \rightarrow 6$
- $n = 4 \rightarrow 24$
- $n = 10 \rightarrow 3.6M$

Bu hızlı büyüme, “her permütasyonu dene” algoritmalarının neden çalışmadığını (TSP, vb.) açıklıyor.

10.9 $P^{-1} = P^T$ — Permütasyonların Sihirli Özelliği

Permütasyon matrisleri **ortogonal**:

$$P^{-1} = P^T$$

Geometri: P bir takas, geri almak = aynı takası tekrar. Ortogonal matrisler Ders 17'de detaylı işlenecek.

Grup yapısı: $n!$ permütasyon bir matematiksel grup oluşturur — kapalı çarpım, identity, terslere sahip. ML'de gruplar simetri = **equivariant networks** (GNN, transformer permütasyon-equivariance, group convolutions).

10.10 Bu Dersin Özeti

1. $(AB)^{-1} = B^{-1}A^{-1}$ — sıra ters.
2. $(A^T)^{-1} = (A^{-1})^T$ — transpoz ve ters yer değiştirir.
3. $A = LU$ — eliminasyonun doğru matris formu.
4. **Çarpanlar L 'ye doğrudan oturur** — eliminasyon sırasında sadece ℓ_{ij} kayıtları, L bedava.
5. $A = LDU$ — alternatif simetrik form (D pivotları taşır).
6. **Maliyet** $\sim n^3/3$ (\mathbf{b} için $\sim n^2$).
7. **Permütasyon matrisleri** P — satır takası, $n \times n$ için $n!$ tane.
8. $P^{-1} = P^T$ — ortogonalite.

! Tek bir cümle

Eliminasyon = $A = LU$ demektir. L eliminasyon çarpanlarını yerinde taşır; U pivotlu üst üçgensel. Numerik LA'nın en temel ayrışımı — solve çağrısının arkasında bu çalışıyor.

10.11 Kontrol Soruları

i Soru 1: $A = ((3,6),(9,14))$ için LU bul.

$$\ell_{21} = 9/3 = 3. E_{21}A = U:$$

$$U = \begin{pmatrix} 3 & 6 \\ 0 & -4 \end{pmatrix}$$

L : çarpan +3, (2,1)'e:

$$L = \begin{pmatrix} 1 & 0 \\ 3 & 1 \end{pmatrix}$$

Pivotlar: 3, -4. $LU = A$ ✓.

i Soru 2: $A = ((1,2),(0,1))$, $B = ((1,0),(3,1))$. $(AB)^{-1}$ 'i iki yoldan.

Yol 1 — direkt: $AB = \begin{pmatrix} 7 & 2 \\ 3 & 1 \end{pmatrix}$, $\det = 1$, $(AB)^{-1} = \begin{pmatrix} 1 & -2 \\ -3 & 7 \end{pmatrix}$.

Yol 2 — formül: $A^{-1} = \begin{pmatrix} 1 & -2 \\ 0 & 1 \end{pmatrix}$, $B^{-1} = \begin{pmatrix} 1 & 0 \\ -3 & 1 \end{pmatrix}$. Sıra ters:

$$B^{-1}A^{-1} = \begin{pmatrix} 1 & -2 \\ -3 & 7 \end{pmatrix} \checkmark$$

$A^{-1}B^{-1}$ yapsaydık yanlış çıkardı.

i Soru 3: $A = ((0,1,1),(1,0,1),(1,1,0))$ için satır takası gerekli — $PA = LU$ yap.

$(1,1)$ 'de $\mathbf{0}$ var \rightarrow satır 1 \leftrightarrow satır 2 takas, $P = P_{12}$:

$$PA = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

Eliminasyon: $\ell_{31} = 1, \ell_{32} = 1$. U :

$$U = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & -2 \end{pmatrix}$$

L (çarpanlar):

$$L = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

Pivotlar 1, 1, -2 — sıfır yok, A invertible.

i Soru 4: solve niye inv @ \mathbf{b} 'den hem hızlı hem doğru? (LU bağlantısı.)

Hız:

- $\text{inv}(A)$: $LU + n$ tane RHS çözümü ($n \cdot n^2 = n^3$) + $\text{inv} \cdot \mathbf{b}$ (n^2). Toplam $\sim 4n^3/3$.
- $\text{solve}(A, \mathbf{b})$: tek LU ($n^3/3$) + tek forward/back sub (n^2). Toplam $\sim n^3/3$. $\sim 4 \times$ hızlı.

Sağlamlık: inv tüm hücreleri hesaplar \rightarrow her hücrede float hata birikir; sonra $\text{inv} \cdot \mathbf{b}$ hataları büyütür. solve partial pivoting'li LU — stabilite optimize.

İstisna: A^{-1} 'i çok kere kullanacaksan inv hesaplayıp cache'le. Modern Bayesian inference/Kalman: lu_factor + lu_solve (tam ters yerine LU sakla).

10.12 Egzersizler

Egzersiz 1. $A = \begin{pmatrix} 2 & 1 & 0 \\ 6 & 4 & 1 \\ 4 & 5 & 3 \end{pmatrix}$ için LU — çarpanları kayıtlı et, L 'yi doğrudan inşa et.

Egzersiz 2. Egzersiz 1'in A 'sı için $\mathbf{Ax} = (1, 7, 14)^T$:

- Forward sub: $L\mathbf{y} = \mathbf{b}$
- Back sub: $U\mathbf{x} = \mathbf{y}$

Egzersiz 3. Tridiagonal $A = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 2 & 3 & 1 & 0 \\ 0 & 4 & 6 & 1 \\ 0 & 0 & 5 & 8 \end{pmatrix}$ için sadece L 'nin köşegen-altı elemanlarını bul. (Tridiagonal

10 A = LU Faktörizasyonu

→ LU çok hızlı.)

Egzersiz 4. (Python) `scipy.linalg.lu` ile partial pivoting'li ayrışım.

```
P =
[[0. 0. 1.]
 [1. 0. 0.]
 [0. 1. 0.]]
L =
[[ 1.          0.          0.          ]
 [ 0.66666667  1.          0.          ]
 [ 0.33333333 -0.14285714  1.          ]]
U =
[[ 6.00000000e+00  4.00000000e+00  1.00000000e+00]
 [ 0.00000000e+00  2.33333333e+00  2.33333333e+00]
 [ 0.00000000e+00  0.00000000e+00 -5.55111512e-17]]
Pivotlar (diag U): [ 6.00000000e+00  2.33333333e+00 -5.55111512e-17]
P @ L @ U =
[[ 2.00000000e+00  1.00000000e+00 -1.18952467e-17]
 [ 6.00000000e+00  4.00000000e+00  1.00000000e+00]
 [ 4.00000000e+00  5.00000000e+00  3.00000000e+00]]
```

Egzersiz 5. İspat: A simetrik ve LU ayrışımı (sıralı takasız) varsa, $A = LDL^T$ (Cholesky'nin temeli, Ders 25).

İpucu: $A = LU = LDU'$ ve $A = A^T$. Eşitlikten $L = U'^T$ yani $A = LDL^T$ çıkar.

10.13 Sonraki Ders İçin Hazırlık

Ders 5: Transpoz, Permütasyon, Vektör Uzayları \mathbb{R}^n

- Transpoz kuralları, simetrik matrisler ($A = A^T$).
- Permütasyon matrislerinin grup yapısı.
- **Vektör uzayları** $\mathbb{R}^2, \mathbb{R}^3, \mathbb{R}^n$ ve **alt-uzaylar** — Chapter 3'ün açılışı.

Ders 5 öncesi yapılacak

- Egzersizleri çöz, özellikle **4** (`scipy.linalg.lu`) ve **5** (Cholesky temeli).
- `solve` vs `inv` performansını $n = 1000$ ve $n = 2000$ için ölç.
- Ana cümleyi tekrar oku: “Eliminasyon = $A = LU$. Çarpanlar L 'de yerinde, U 'da pivotlar.”

10.14 Anahtar Kavramlar (Cheat Sheet)

Kavram	Tanım	Strang'da
$(AB)^{-1}$	$B^{-1}A^{-1}$ — sıra ters	0:31
$(A^T)^{-1}$	$(A^{-1})^T$ — transpoz/ters yer değiştirir	4:01
$A = LU$	Eliminasyonun matris formu	7:06
L (lower)	Alt üçgensel, köşegen 1, çarpanlar yerinde	12:02
U (upper)	Üst üçgensel, köşegen pivotlar	8:10
$A = LDU$	L, U 'da 1'ler, D 'de pivotlar	12:22
3×3'te 10 problemi	E 'de çarpanlar karışır, L 'de karışmaz	18:24
Operation count	$n^3/3 + n^2$ per RHS	36:35
Partial pivoting	Nümerik stabilite için en büyüğü pivot	38:50
Permütasyon P	Identity satır takası	39:35
$PA = LU$	Genel form	40:30
$P^{-1} = P^T$	Ortogonallık	47:06
$n!$ permütasyon	$n \times n$ için tam sayı	47:19

10.15 ML Bağlantıları Özeti

💡 7 köprü

1. $A = LU = \text{np.linalg.solve}$ → Her solve çağrısının arkasında.
2. **LU cache** → `lu_factor` + `lu_solve` → Kalman/Newton/Bayesian inference.
3. $n^3/3$ **maliyeti** → Tam matris terslerinden kaçınma; SGD/Adam/KFAC.
4. **Cholesky** $A = LL^T$ → Bayesian inference, Gaussian processes, normal equations.
5. P **permütasyonu** → **equivariant networks** → GNN, transformer permütasyon-equivariance, group convolutions.
6. $(AB)^{-1} = B^{-1}A^{-1}$ → Backward pass, normalizing flows (Glow, RealNVP).
7. $(A^T)^{-1} = (A^{-1})^T$ → Normal equations simetri koruması; ridge, kernel methods, GP.

! Tek bir şey alıp gideceksen

Eliminasyon = $A = LU$. L çarpanları yerinde taşır; U pivotlu üst üçgensel. Nümerik LA'nın temel taşı — solve çağırдыңın her yerde arka planda bu çalışıyor.

11 Transpoz, Permütasyon, Vektör Uzayları \mathbb{R}^n

Vektörlerden uzaylara — ve kolon uzayının doğuşu

Bölüm bilgisi

- **Strang'ın videosu:** [YouTube — Lecture 5: Transposes, Permutations, Spaces \$\mathbb{R}^n\$](#) (≈ 48 dk)
- **OCW sayfası:** [MIT 18.06SC — Lecture 5](#)
- **Okuma süresi:** ≈ 40 dk

11.1 Bu Derste Ne Var?

Ders 4'te $A = LU$ ve permütasyon matrislerine ilk bakışı gördük. Ders 5 iki yarıdan oluşuyor:

1. **Bölüm 2.7'nin tamamlanması:** Genel $PA = LU$ + transpoz kuralları + simetrik matrisler.
2. **Asıl olay — vektör uzayları:** Strang'ın deyişiyle “gerçek lineer cebirin başlangıcı”. Artık tek tek vektörlerle değil, **vektör uzayları** ve **alt-uzayları** ile düşünüyoruz.

“We’re coming to the beginning of real linear algebra, which is seeing a bigger picture with vector spaces — not just vectors, but spaces of vectors and sub-spaces of those spaces.” — Strang, 0:57

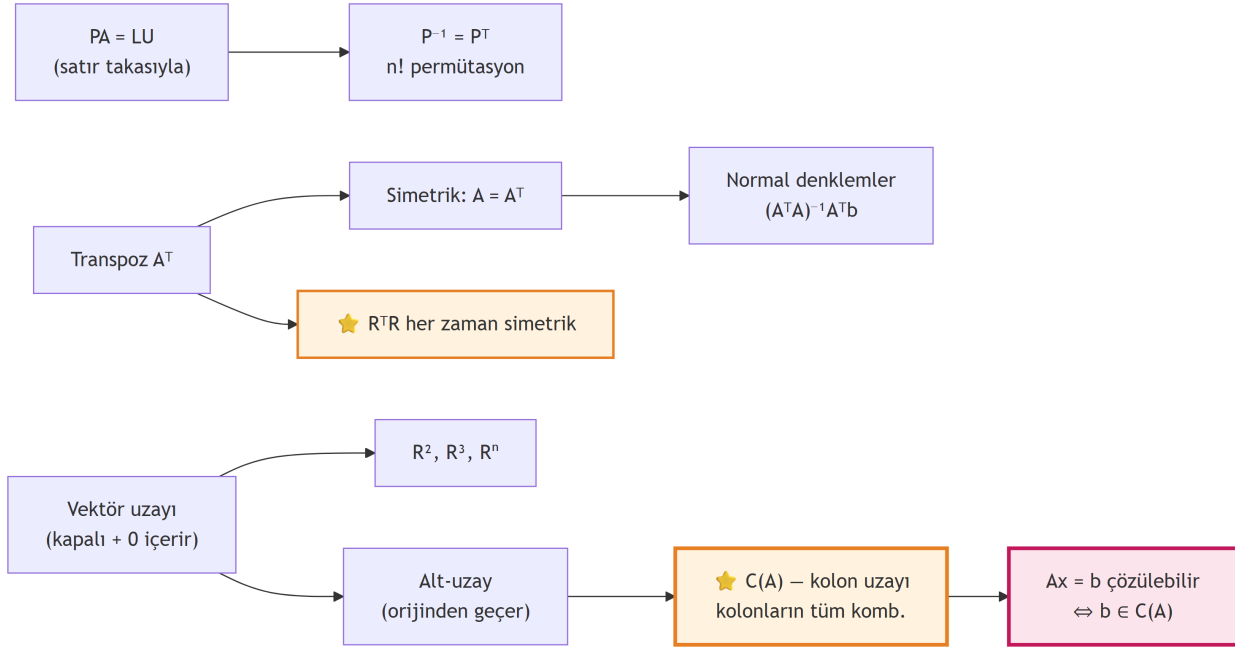
Builder Notu — Vektör Uzayı Neden ML'in Tam Merkezinde

- **Embedding uzayları** birer vektör uzayıdır. Kelime/görüntü/kullanıcı \mathbb{R}^n 'de bir nokta; “anlam” = geometri.
- **Alt-uzay = düşük boyutlu yapı.** Yüksek boyutlu veri çoğunlukla daha düşük boyutlu bir alt-uzaya yakın oturur — **PCA, SVD, LoRA** hep bunu sömürür.
- **Kapalılık** = lineer kombinasyon uzaydan çıkmaz. Bir lineer katmanın bias'sız kısmı çıktıyı tam $C(W)$ 'ye taşır.
- **Bias = afin = orijinden kayma.** PCA'da merkezleme tam da “alt-uzaylar orijinden geçer” kuralına uydurmak için.
- **A^T ve simetri** backprop Jacobian transpozlarının ve kovaryans matrislerinin iskeleti.

11.2 Genel $PA = LU$

Eliminasyon sırasında pivotta sıfır gelirse \rightarrow satır takası $\rightarrow P$ matrisi. Faktörizasyon:

11 Transpoz, Permütasyon, Vektör Uzayları R^n



Şekil 11.1: Transpoz/permütasyon kapanışı → vektör uzayı tanımı → alt-uzay → $C(A)$.

$$PA = LU$$

Pratikte neredeyse her matris için $P = I$, ama gerektiğinde takas önceden yapılıp L, U tıpkı Ders 4'teki gibi temiz çıkar.

“It checks: is that pivot big enough? Pivots close to zero are numerically bad.” — Strang, 4:02

Numerik sağlamlık: LAPACK her zaman partial pivoting yapar — sıfır olmasa bile küçük pivotlardan kaçınır. Cebir “takas gereksiz” dese de doğruluk “gerekli” der.

11.3 Permütasyon Matrislerinin Özellikleri

$P = \text{identity}$ 'nin satırları yeniden sıralanmış hâli. $n!$ tane $n \times n$ permütasyon vardır:

n	$n!$
3	6
4	24
5	120
10	~3.6M

Hızlı büyür — “her permütasyonu dene” TSP’de neden işlemez bunu söylüyor.

Sihirli özellik: $P^{-1} = P^T$ (ortogonal). P bir takas; geri almak yine bir takas — ve bu transpoz.

“*P transpose P is the identity. That tells me this is the inverse of that.*” — Strang, 8:39

Grup yapısı: $n!$ permütasyon bir matematiksel grup oluşturur (kapalı, terslere sahip, identity). ML’de gruplar = **equivariant networks** (transformer permütasyon-equivariance, GNN, group convolutions).

11.4 Transpoz

Tanım: $(A^T)_{ij} = A_{ji}$. Satır ve kolon numaraları yer değiştirir.

$$A = \begin{pmatrix} 1 & 2 & 4 \\ 3 & 3 & 1 \end{pmatrix} \rightarrow A^T = \begin{pmatrix} 1 & 3 \\ 2 & 3 \\ 4 & 1 \end{pmatrix}$$

$A (2 \times 3) \rightarrow A^T (3 \times 2)$. Boyutlar yer değiştirir.

11.5 Simetrik Matrisler

$$A^T = A \quad (\text{yalnız kare matrisler için mümkün})$$

Örnek:

$$S = \begin{pmatrix} 2 & 1 & 7 \\ 1 & 3 & 9 \\ 7 & 9 & 5 \end{pmatrix}, \quad S^T = S$$

Köşegen üstü/altı ayna gibi yansır. **Gözle anında tanınır.**

💡 Builder Notu — Simetri Her Yerde

- **Kovaryans** Σ : $\text{cov}(x, y) = \text{cov}(y, x) \rightarrow$ simetrik. PCA bu simetriye dayanır.
- **Gram matrisi** $G = X^T X$ ve **kernel matrisi** $K_{ij} = \langle x_i, x_j \rangle$ simetrik.
- Simetrik matrislerin özdeğerleri **gerçel**, özvektörleri **ortogonal** (Ders 25). Spektral yöntemlerin temeli.
- Simetrik pozitif tanımlı için $A = LU \rightarrow A = LL^T$ — **Cholesky** (Ders 28).

11.6 $R^T R$ Her Zaman Simetrik

Herhangi bir dikdörtgen R için $R^T R$ **her zaman simetriktir**. İki satırlık ispat:

$$(R^T R)^T = R^T (R^T)^T = R^T R \checkmark$$

(Çarpımın transpozunda sıra ters döner; $(R^T)^T = R$.)

```
R^T R =
[[10. 11.  7.]
 [11. 13. 11.]
 [ 7. 11. 17.]]
Simetrik mi? True
```

Builder Notu — Normal Denklemler

- **Least squares** çözümü $\hat{\mathbf{x}} = (A^T A)^{-1} A^T \mathbf{b}$ — $A^T A$ simetrik, $(A^T A)^{-1}$ de simetrik kalır.
- **Linear regression, ridge, kernel methods** (Ders 15–17) — hepsi bu pattern.
- $A^T A$ ayrıca **pozitif yarı-tanımlı** — SVD ve PCA'nın matematiği tam buradan.

11.7 Vektör Uzayları — \mathbb{R}^2 , \mathbb{R}^3 , \mathbb{R}^n

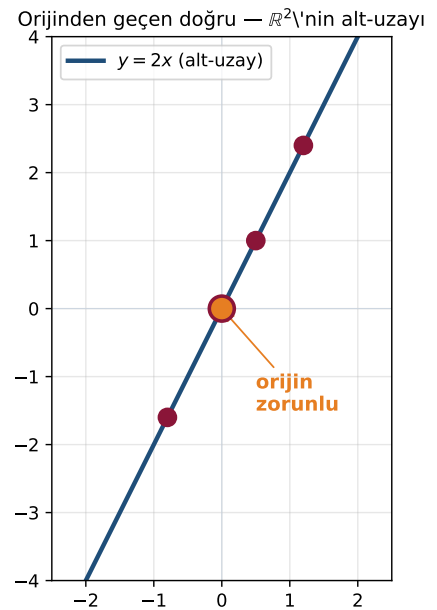
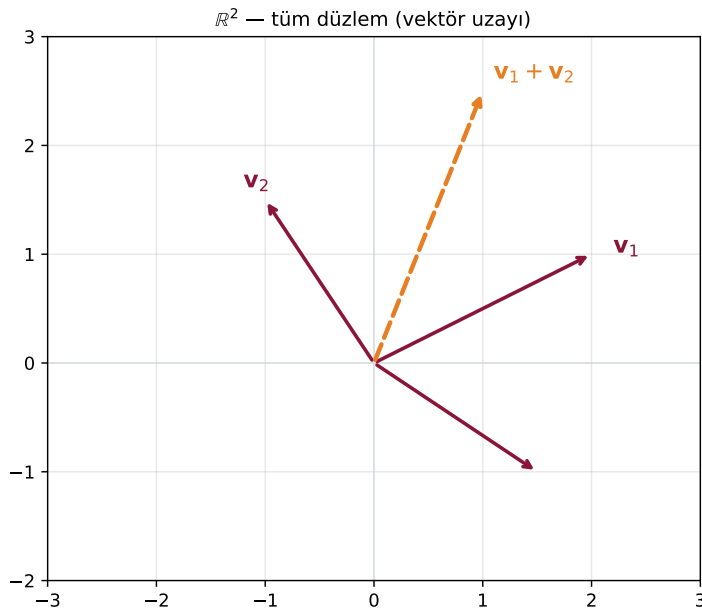
Vektör uzayı: Vektörlerin iki temel işlemi (toplama + skalerle çarpma) altında **kapalı** olan vektör kümesi.

İkisi birleşince: **linear kombinasyon** $c\mathbf{v} + d\mathbf{w}$ her zaman uzayda olmalı. Bu “içeride kalma” özelliği = **kapalılık (closure)**.

En tanıdık örnek: $\mathbb{R}^2 =$ sıradan xy -düzlemi:

$$\mathbb{R}^2 = \left\{ \begin{pmatrix} x \\ y \end{pmatrix} : x, y \in \mathbb{R} \right\}$$

Genel: $\mathbb{R}^n = n$ bileşenli tüm reel kolon vektörleri. Bu kursta vektörler kolon ve bileşenler reel.



Şekil 11.2: Vektör uzayında kapalılık: iki vektörü toplayınca veya skalerle çarpınca uzaydan çıkmazsın. (Üst) \mathbb{R}^2 düzlemi — sınırsız uzay. (Alt) $y = 2x$ doğrusu — alt-uzay, orijinden geçer.

11.8 Sıfır Vektör Zorunlu

Her vektör uzayı **sıfır vektörünü içermek zorunda**. Sebep: kapalılıktan gelir. Uzaydaki herhangi \mathbf{v} 'yi 0 ile çarpınca $\mathbf{0}$ çıkar, ve uzayda kalmalı.

“No way I can do without the origin. Every vector space has got that zero vector in it.” — Strang, 26:29

Karşı-örnek — Pozitif çeyrek ($x \geq 0, y \geq 0$): Toplama altında kapalı (her ikisi de pozitif). Ama $(3, 2) \times (-5) = (-15, -10)$ çeyrekte çıkar \rightarrow **kapalı değil** \rightarrow vektör uzayı değil.

11.9 Alt-Uzaylar (Subspace)

Uzay içinde uzay. İki kapalılık kuralı yeterli (ve dolayısıyla $\mathbf{0}$ içerir).

\mathbb{R}^2 'deki **orijinden geçen doğru bir alt-uzaydır**. İçindeki bir vektörü çarpınca/toplayınca üstte kalırsın.

Kritik şart — orijinden geçmeli. Orijinden geçmeyen doğru alt-uzay **değildir**: bir noktayı 0 ile çarpınca $\mathbf{0}$ çıkar, doğrudan değildir \rightarrow kapalılık ihlali.

$$\mathbf{v} \in S \implies c\mathbf{v} \in S \quad \forall c \in \mathbb{R}$$

Builder Notu — Lineer vs Afin

- $\mathbf{y} = W\mathbf{x}$ (bias yok) \rightarrow çıktı tam bir alt-uzayda (orijinden geçer).
- $\mathbf{y} = W\mathbf{x} + \mathbf{b}$ (bias var) \rightarrow **afin** (orijinden kaymış); teknik olarak alt-uzay değil.
- Bias modele “orijinden kayma” özgürlüğü verir.
- **PCA’da merkezleme** (veri ortalamasını çıkarma) tam da bu kayma’yı sıfırlamak için zorunlu.

11.10 \mathbb{R}^2 'nin ve \mathbb{R}^3 'ün Tüm Alt-Uzayları

\mathbb{R}^2 'nin alt-uzayları (tam üç tür):

1. Tüm \mathbb{R}^2
2. Orijinden geçen herhangi doğru
3. Yalnız $\{\mathbf{0}\}$ — en küçük

\mathbb{R}^3 'ün alt-uzayları (dört tür):

- Tüm \mathbb{R}^3
- Orijinden geçen düzlemler
- Orijinden geçen doğrular
- $\{\mathbf{0}\}$

11 Transpoz, Permütasyon, Vektör Uzayları \mathbb{R}^n

Bu hiyerarşi, **boyut** kavramının habercisi (Ders 9).

Dikkat: Orijinden geçen doğru \mathbb{R}^1 gibi *görünür*, ama \mathbb{R}^1 **değildir** — vektörlerinin iki bileşeni var, \mathbb{R}^1 'inkilerin bir.

11.11 Kolon Uzayı $C(A)$ — Matristen Doğan Alt-Uzay

Dersin doruk noktası. Bir matrisin **kolon uzayı** = kolonlarının **tüm lineer kombinasyonları**.

$$A = \begin{pmatrix} 1 & 3 \\ 2 & 3 \\ 4 & 1 \end{pmatrix}$$

İki kolon \mathbb{R}^3 'te. Tüm kombinasyonları:

$$C(A) = \{c_1(1, 2, 4)^T + c_2(3, 3, 1)^T : c_1, c_2 \in \mathbb{R}\}$$

Geometrik olarak: iki bağımsız vektörün kombinasyonları \mathbb{R}^3 'te **orijinden geçen bir düzlem** doldurur. Bir doğrudan fazlası, tüm \mathbb{R}^3 değil.

“When I take all their combinations, I fill out a whole plane, and it's through the origin.” — Strang, 45:05

! $C(A)$ 'nın Anlamı

$A\mathbf{x} = \mathbf{b}$ çözülebilir $\iff \mathbf{b} \in C(A)$.

Çünkü $A\mathbf{x}$ tam olarak A 'nın kolonlarının \mathbf{x} ağırlıklarıyla kombinasyonudur (Ders 1!). ML'de $C(W)$ bir lineer katmanın “erişebileceği çıktılar” kümesi — modelin temsil sınırı.

11.12 Bu Dersin Özeti

1. $PA = LU$ — satır takasıyla genel form.
2. **Permütasyon:** $n!$ tane, $P^{-1} = P^T$, grup yapısı.
3. **Transpoz:** $(A^T)_{ij} = A_{ji}$.
4. **Simetrik:** $A^T = A$, sadece kare matrislerde.
5. $R^T R$ **simetrik** — iki satırlık ispat; normal denklemlerin temeli.
6. **Vektör uzayı** — kapalılık + $\mathbf{0}$.
7. **Alt-uzay** — uzay içinde uzay, orijinden geçer.
8. $C(A)$ — kolonların tüm kombinasyonları; matristen doğan ilk büyük alt-uzay.

! Tek bir cümle

Vektör uzayı, lineer kombinasyon altında **kapalı** ve $\mathbf{0}$ 'ı içerendir. Bir matrisin kolonlarının tüm kombinasyonları **kolon uzayı** $C(A)$ 'yı doldurur; $A\mathbf{x} = \mathbf{b}$ ancak $\mathbf{b} \in C(A)$ olduğunda çözülebilir.

11.13 Kontrol Soruları

i Soru 1: $B = ((1,4),(2,5),(3,6))$ için B^T — simetrik mi?

$$B^T = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$

Simetrik **değil**. B 3×2 , B^T 2×3 — boyutlar aynı bile değil. Simetri yalnız kare matrislerde mümkün.

i Soru 2: $R = ((1,2),(0,1),(3,0))$ için $R^T R$ 'i hesapla.

$$R^T R = \begin{pmatrix} 1 & 0 & 3 \\ 2 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ 0 & 1 \\ 3 & 0 \end{pmatrix} = \begin{pmatrix} 10 & 2 \\ 2 & 5 \end{pmatrix}$$

$(1, 2) = (2, 1) = 2 \rightarrow$ simetrik \checkmark . **Niye her zaman:** $(R^T R)^T = R^T (R^T)^T = R^T R$.

i Soru 3: \mathbb{R}^2 'nin alt-uzayı: (a) $y = 2x$, (b) $y = 2x + 1$, (c) $\{0\}$, (d) birinci çeyrek.

- (a) $y = 2x \rightarrow$ **EVET**. Orijinden geçen doğru.
- (b) $y = 2x + 1 \rightarrow$ **HAYIR**. $(0, 0)$ doğrudan değil \rightarrow orijinden geçmiyor.
- (c) $\{0\} \rightarrow$ **EVET**. $0 + 0 = 0$, $c \cdot 0 = 0$.
- (d) **Birinci çeyrek** \rightarrow **HAYIR**. $(3, 2) \cdot (-1) = (-3, -2)$ çeyrekte çıkar.

Test: “Orijini içeriyor mu?”

i Soru 4: $y = Wx + b$ 'nin görüntüsü neden alt-uzay değil? $b = 0$ olursa?

$x = 0 \rightarrow y = b$. $b \neq 0$ ise çıktı orijinden geçmez \rightarrow **afin**, alt-uzay değil.

$b = 0$ olursa: $y = Wx$, $x = 0 \rightarrow y = 0$, görüntü tam olarak $C(W) =$ gerçek alt-uzay.

PCA'da merkezleme veri ortalamasını çıkararak afin'i lineer'e dönüştürür.

11.14 Egzersizler

Egzersiz 1. $P = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$ için P^{-1} 'i transpozla bul, $P^T P = I$ doğrula.

Egzersiz 2. $A = \begin{pmatrix} 1 & 2 \\ 2 & 4 \\ 3 & 6 \end{pmatrix}$. $C(A)$ doğru mu düzlem mi? (İpucu: kolonlar bağımsız mı?)

Egzersiz 3. Hangileri \mathbb{R}^3 alt-uzayı?

- (a) $x + y + z = 0$ olan tüm (x, y, z)
- (b) $x + y + z = 1$

11 Transpoz, Permütasyon, Vektör Uzayları R^n

- (c) $(x, y, 0)$ tipi vektörler
- (d) $z \geq 0$ olan vektörler

Egzersiz 4. (Python)

```
R^T =
[[1. 0. 3.]
 [2. 1. 0.]]
R^T R =
[[10. 2.]
 [ 2. 5.]]
Simetrik mi? True
P^-1 == P^T? True
```

Egzersiz 5. İspatla: Her kare A için $A + A^T$ simetriktir. (İpucu: $(A + A^T)^T$.) Bu, her matrisin simetrik + antisimetrik parçaya ayrılabilmesinin (Ders 25 habercisi) ilk adımı.

11.15 Sonraki Ders İçin Hazırlık

Ders 6: Kolon Uzayı ve Sıfır Uzayı

- Alt-uzayların kesişimi/birleşimi.
- $C(A)$ ve $Ax = \mathbf{b}$ çözülebilirliği.
- **Sıfır uzayı** $N(A)$: $Ax = \mathbf{0}$ 'ı çözen tüm \mathbf{x} 'ler.

⚠ Ders 6 öncesi

- Egzersizleri çöz, özellikle 3 (alt-uzay testi).
- `np.linalg.matrix_rank` ile birkaç matrisin kolon bağımsızlığını yokla.
- Ana cümleyi tekrar oku.

11.16 Anahtar Kavramlar (Cheat Sheet)

Kavram	Tanım	Strang'da
$PA = LU$	Satır takasıyla genel form	5m00
Partial pivoting	Numerik stabilite için en büyüğü seç	4m02
$n!$ permütasyon	$n \times n$ için tam sayı	7m46
$P^{-1} = P^T$	Permütasyonların ortogonalliği	8m39
Transpoz	$(A^T)_{ij} = A_{ji}$	12m39
Simetrik	$A^T = A$, kare matriste	13m16

Kavram	Tanım	Strang'da
$R^T R$ simetrik	Her R için; normal denklemler	15m44
Vektör uzayı	Kapalı küme	21m50
Kapalılık	Lineer kombinasyon uzayda kalır	31m14
$\mathbf{0}$ zorunlu	Her alt-uzay orijini içerir	26m29
Alt-uzay	Uzay içinde uzay	35m05
\mathbb{R}^2 alt-uzayları	Tüm \mathbb{R}^2 , doğrular, $\{\mathbf{0}\}$	38m01
$C(A)$	Kolonların tüm kombinasyonları	42m46

11.17 ML Bağlantıları Özeti

💡 7 köprü

1. \mathbb{R}^n = **embedding uzayı** → Boyut = temsil kapasitesi.
2. **Alt-uzay = düşük boyutlu yapı** → PCA, SVD, LoRA.
3. **Simetrik matrisler** → Covariance, Gram, kernel — spektral yöntemlerin temeli.
4. $R^T R$ / **normal denklemler** → Least squares, ridge, kernel methods.
5. **Permütasyon grubu** → **equivariance** → Transformer, GNN, group conv.
6. $C(W)$ → Lineer katmanın temsil sınırı; $A\mathbf{x} = \mathbf{b}$ çözülebilirliği.
7. **Alt-uzay** ↔ **afin** → Bias kayma; PCA merkezleme tam bu kuralı düzeltir.

❗ Tek bir şey alıp gideceksen

Lineer cebir artık “tek tek vektörlerden” çıkıp **vektör uzaylarına** geçti. Kapalı + $\mathbf{0}$ içeren her küme uzay; alt-uzay orijinden geçer. Bir matrisin kolonlarının tüm kombinasyonları $C(A)$ 'yı doldurur — $A\mathbf{x} = \mathbf{b}$ ancak $\mathbf{b} \in C(A)$ ise çözülebilir.

12 Kolon Uzayı ve Sıfır Uzayı

$C(A)$ ve $N(A)$ — bir matrisin iki yüzü

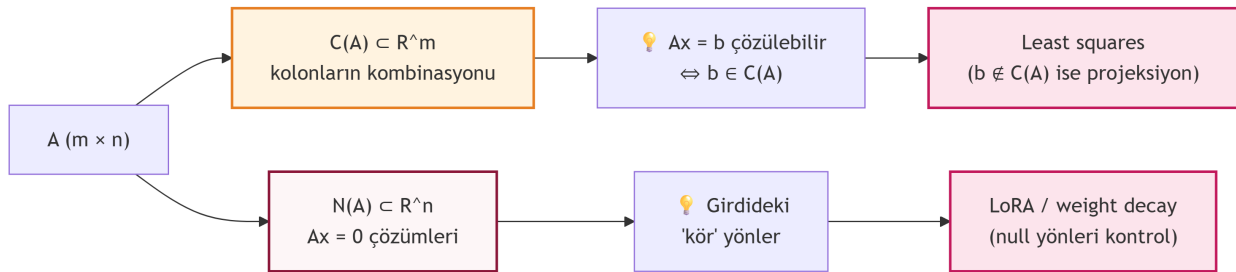
i Bölüm bilgisi

- **Strang'in videosu:** [YouTube — Lecture 6: Column Space and Nullspace](#) (≈50 dk)
- **OCW sayfası:** [MIT 18.06SC — Lecture 6](#)
- **Okuma süresi:** ≈40 dk

12.1 Bu Derste Ne Var?

Ders 5'te vektör uzayı, alt-uzay ve kolon uzayını tanıttık. Ders 6 bir matristen doğan **iki** temel alt-uzayı yan yana koyuyor.

1. **Kolon uzayı** $C(A)$: kolonların tüm kombinasyonları → $Ax = b$ hangi **b**'ler için **çözülebilir**?
2. **Sıfır uzayı** $N(A)$: $Ax = 0$ 'ı çözen tüm x 'ler → **girdideki fazlalık yönleri**.



Şekil 12.1: Bir matrisin iki yüzü: çıktı tarafı $C(A)$ (erişilebilirlik) ve girdi tarafı $N(A)$ (fazlalık).

“Two subspaces that we’re specially interested in: the column space and the null space.” — Strang, 0:47

💡 Builder Notu — $C(W)$ ve $N(W)$

- $C(W)$ = **erişilebilirlik / ifade gücü**. Lineer katman $y = Wx$ 'in üretebileceği tüm çıktılar $C(W)$ 'de. Hedef bunun dışındaysa tam çözüm yok → **least squares projeksiyonu**.
- $N(W)$ = **fazlalık / serbestlik**. $Wv = 0$ olan yön → girdideki o sinyal yok oluyor. Modern derin ağlar genelde aşırı parametrize: $N(W) \neq \{0\}$. **Weight decay / LoRA** bu null yönlerini kontrol

eder.

- “Kernel” kelimesi ML’de bu null uzayından gelir.

12.2 Alt-Uzaylar Üzerinde İşlemler

Birleşim ($P \cup L$) — alt-uzay DEĞİL:

Düzlem P ve doğru L ’nin vektörlerini bir araya toplar. Düzlemden bir vektör + doğrudan bir vektör = ikisinde de olmayan üçüncü bir vektör → kapalılık ihlali.

Kesişim ($S \cap T$) — her zaman alt-uzay:

$\mathbf{v}, \mathbf{w} \in S \cap T \rightarrow$ ikisi de hem S hem T ’de. $\mathbf{v} + \mathbf{w}$ hem S ’de hem T ’de (her ikisi alt-uzay) → kesişimde. ✓

Builder Notu: Kısıtların **kesişimi** lineer kalır; “ya/ya” tipi birleşim lineer-olmayan zorluk üretir.

12.3 Kolon Uzayı $C(A)$ — Tanım

Strang’ın örneği:

$$A = \begin{pmatrix} 1 & 1 & 2 \\ 2 & 1 & 3 \\ 3 & 1 & 4 \\ 4 & 1 & 5 \end{pmatrix}$$

Her kolon \mathbb{R}^4 ’te. $C(A)$, \mathbb{R}^4 ’ün alt-uzayı. İçinde: üç kolon ve tüm kombinasyonları.

Genel kural: $m \times n$ matris için $C(A) \subset \mathbb{R}^m$.

12.4 $A\mathbf{x} = \mathbf{b}$ Ne Zaman Çözülebilir?

$A\mathbf{x} = x_1\mathbf{c}_1 + x_2\mathbf{c}_2 + x_3\mathbf{c}_3$ — kolonların kombinasyonu (Ders 1!).

! Çözülebilirlik kuralı

$A\mathbf{x} = \mathbf{b}$ çözülebilir $\iff \mathbf{b} \in C(A)$.

“I can solve $A\mathbf{x} = \mathbf{b}$ exactly when \mathbf{b} is in the column space.” — Strang, 23:39

Hemen çözülebilen \mathbf{b} ’ler: $\mathbf{b} = \mathbf{0}$ ($\mathbf{x} = \mathbf{0}$), \mathbf{b} = bir kolonun kendisi.

Builder Notu — Least Squares: Gerçek hayatta \mathbf{b} genelde $C(A)$ ’da değildir (gürültülü, fazla denklem). Tam çözüm yoktur; \mathbf{b} ’nin $C(A)$ üzerine **izdüşümünü** alıp en yakın çözümü buluruz (Ders 15-16). Bir nöral ağın hedefi tam üretememesi de aynı durumdur.

12.5 Pivot Kolonlar — Bağımlı Kolonu At

Strang'ın örneğinde:

$$\mathbf{c}_1 + \mathbf{c}_2 = \mathbf{c}_3 \quad ((1, 2, 3, 4) + (1, 1, 1, 1) = (2, 3, 4, 5))$$

Üçüncü kolon bağımlı. **Atılabilir**, $C(A)$ **değişmez**. İlk iki = **pivot kolon**. Sonuç: $C(A)$, \mathbb{R}^4 içinde **2 boyutlu** alt-uzay (düzlem).

Bu “bağımsız kolon sayısı” = **rank** (Ders 7).

12.6 Sıfır Uzayı $N(A)$ — Tanım

Tamamen farklı bir alt-uzay. $C(A)$ sağ tarafla ilgiliydi; $N(A)$ çözümlerle ilgili.

$$N(A) = \{\mathbf{x} : A\mathbf{x} = \mathbf{0}\}$$

Strang'ın örneğinde (4×3): \mathbf{x} 'ler \mathbb{R}^3 'te $\rightarrow N(A) \subset \mathbb{R}^3$.

Genel kural: $m \times n$ matris için $N(A) \subset \mathbb{R}^n$. ($C(A)$ ile **farklı uzaylarda!**)

12.7 Null Uzayını Bulmak

$\mathbf{c}_3 = \mathbf{c}_1 + \mathbf{c}_2$ olduğundan:

$$1 \cdot \mathbf{c}_1 + 1 \cdot \mathbf{c}_2 - 1 \cdot \mathbf{c}_3 = \mathbf{0} \implies \mathbf{x} = (1, 1, -1)^T \in N(A)$$

Tüm katları da çözüm:

$$N(A) = \left\{ c \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix} : c \in \mathbb{R} \right\} = \mathbb{R}^3 \text{te orijinden geçen bir doğru}$$

$\text{rank}(A) = 2$

$N(A)$ baz vektörü =

[[0.57735027]

[0.57735027]

[-0.57735027]]

$A @ N =$

[[0.]

[0.]

[0.]

[0.]]

12.8 $N(A)$ Gerçekten Alt-Uzay mı? — İki Satırlık İspat

Toplama: $Av = \mathbf{0}, Aw = \mathbf{0} \rightarrow$

$$A(\mathbf{v} + \mathbf{w}) = Av + Aw = \mathbf{0} + \mathbf{0} = \mathbf{0} \checkmark$$

Skalerle çarpma: $A(c\mathbf{v}) = c(A\mathbf{v}) = c \cdot \mathbf{0} = \mathbf{0} \checkmark$

Dağılıma özelliği yetiyor. $N(A)$ bir alt-uzay.

💡 Builder Notu — Kernel = Null = Bilgi Kaybı

Bir nöral ağda, kayba (loss) hiç etki etmeyen parametre yönleri = N . Optimizasyon bu yönlerde **kör**. Weight decay bu null yönleri sıfıra çeker. LoRA düşük-rank zorlaması da N 'i kasıtlı genişletir (parametre sayısı azaltma).

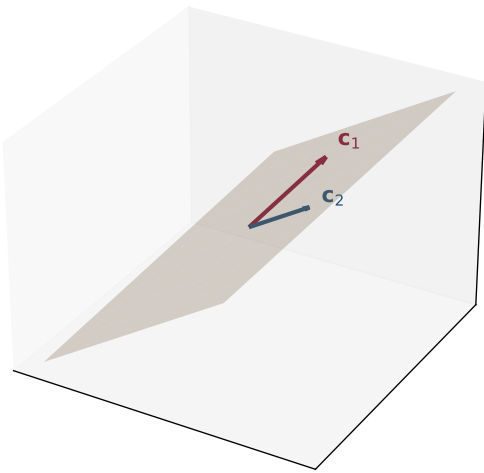
12.9 $Ax = \mathbf{b}$ 'nin Çözümleri Niye Alt-Uzay Değil?

$\mathbf{b} \neq \mathbf{0}$ için: $A \cdot \mathbf{0} = \mathbf{0} \neq \mathbf{b} \rightarrow$ **sıfır vektörü çözüm değil** \rightarrow alt-uzay olamaz.

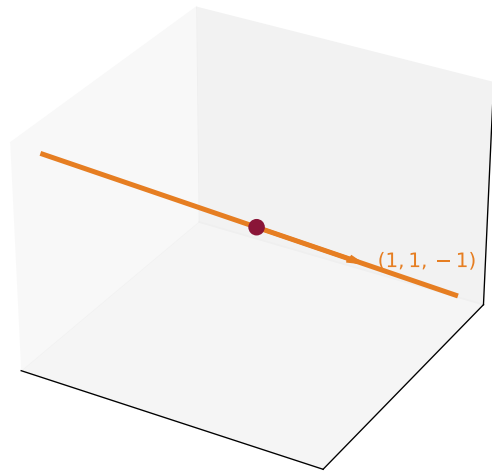
“The zero vector is not a solution, so I never even got started.” — Strang, 42:38

Yapı: Bir özel çözüm $\mathbf{x}_p + N(A) \rightarrow$ orijinden geçmeyen, $N(A)$ 'ya paralel kaymış doğru/düzlem (afin). Bu Ders 7'nin “tam çözüm = özel çözüm + null uzayı” temasının habercisi.

$C(A) \subset \mathbb{R}^m$ (kolonların düzlemi)



$N(A) \subset \mathbb{R}^n$ ($Ax = 0$ doğrusu)



Şekil 12.2: İki alt-uzay görsel: solda $C(A) \subset \mathbb{R}^m$ (kolonların doldurduğu yer), sağda $N(A) \subset \mathbb{R}^n$ ($Ax=0$ yönleri). Farklı uzaylarda yaşarlar.

12.10 Bir Alt-Uzayı Tanımlamanın İki Yolu

1. **Kolon uzayı tarzı** — vektör ver, kombinasyon al. Birkaç vektör, tüm kombinasyonları \rightarrow uzayı içeriden inşa et.
2. **Null uzayı tarzı** — denklem ver, sağlayanları bul. $Ax = \mathbf{0} \rightarrow$ uzayı kısıtlarla tarif et.

Lineer cebirin büyük kısmı bu iki bakış arasında gidip gelmektedir.

12.11 Bu Dersin Özeti

1. **Birleşim alt-uzay değil, kesişim her zaman alt-uzay.**
2. $C(A) \subset \mathbb{R}^m$ — kolonların kombinasyonları.
3. $Ax = \mathbf{b}$ **çözülebilir** $\iff \mathbf{b} \in C(A)$.
4. **Pivot kolonlar** = bağımsız kolonlar; bağımlılar uzaya katkı yok.
5. $N(A) \subset \mathbb{R}^n$ — $Ax = \mathbf{0}$ çözümleri.
6. $N(A)$ **alt-uzaydır** — dağılma ispatı.
7. $Ax = \mathbf{b}$ ($\mathbf{b} \neq \mathbf{0}$) **çözümleri alt-uzay değil** (orijini içermez).
8. İki tanımlama yolu: kombinasyon / denklem.

! Tek bir cümle

$C(A)$, $Ax = \mathbf{b}$ 'nin hangi \mathbf{b} 'ler için çözülebileceğini söyler ($\mathbf{b} \in C(A)$). $N(A)$, $Ax = \mathbf{0}$ 'ın tüm çözümlerini toplar. Biri çıktı tarafında erişilebilirliği, diğeri girdi tarafında serbestliği ölçer.

12.12 Kontrol Soruları

i Soru 1: $A = ((1,2),(2,4),(1,2))$ için $C(A)$ hangi uzayda, kaç boyutlu?

Kolonlar 3 bileşenli $\rightarrow C(A) \subset \mathbb{R}^3$. $c_2 = 2c_1 \rightarrow$ bağımlı. **1 boyutlu**: \mathbb{R}^3 'te orijinden geçen doğru.

i Soru 2: Aynı A için $\mathbf{b} = (3,6,3)^T$ ve $\mathbf{b} = (1,0,0)^T$ çözülebilir mi?

$C(A)$, $(1, 2, 1)$ yönündeki doğru.

- $\mathbf{b} = (3, 6, 3)^T = 3 \cdot (1, 2, 1) \rightarrow$ **çözülebilir**, $\mathbf{x} = (3, 0)$.
- $\mathbf{b} = (1, 0, 0)^T \rightarrow (1, 2, 1)$ 'in katı değil $\rightarrow C(A)$ dışı \rightarrow **çözüm yok**.

i Soru 3: $A = ((1,2,3),(2,4,6))$ için $N(A)$ — \mathbb{R} kaçta, hangi şekil?

\mathbf{x} 3 bileşenli $\rightarrow N(A) \subset \mathbb{R}^3$. İki satır birbirinin katı \rightarrow tek bağımsız denklem: $x_1 + 2x_2 + 3x_3 = 0$. \mathbb{R}^3 'te **orijinden geçen düzlem** (2 serbest yön):

$$N(A) = \left\{ c_1 \begin{pmatrix} -2 \\ 1 \\ 0 \end{pmatrix} + c_2 \begin{pmatrix} -3 \\ 0 \\ 1 \end{pmatrix} \right\}$$

i Soru 4: W 'nin null uzayı yalnız $\{0\}$ değilse, model hakkında ne söyler?

$N(W) \neq \{0\} \rightarrow \mathbf{v} \neq \mathbf{0}$ var ki $W\mathbf{v} = \mathbf{0}$. Yani $W(\mathbf{x} + c\mathbf{v}) = W\mathbf{x}$ — girdideki \mathbf{v} yönü **kör**.

- Boyut indirgemede istenir (sıkıştırma).
- Gereksizse aşırı parametrize; **weight decay / LoRA** budur.
- $\dim N(W)$ = “kaç serbestlik derecesi boşa harcanmış”.

12.13 Egzersizler

Egzersiz 1. $A = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & 1 \\ 1 & 1 & 2 \end{pmatrix}$. Pivot kolonlar? $C(A)$ kaç boyutlu? (İpucu: üçüncü = ilk + ikinci?)

Egzersiz 2. Aynı A için $\mathbf{b} = (6, 2, 4)^T$ kolon uzayında mı? Bir \mathbf{x} bul.

Egzersiz 3. $A = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \end{pmatrix}$ için $N(A)$ — geometrik şekil?

Egzersiz 4. (Python) `scipy.linalg.null_space` ile doğrula.

```
rank = 2
N(A) bazı =
[[ 0.57735027]
 [ 0.57735027]
 [-0.57735027]]
A @ N ~ 0?
[[0.]
 [0.]
 [0.]
 [0.]]
```

Egzersiz 5. Bu dersin matrisi: $C(A) = 2$ boyut, $N(A) = 1$ boyut. Kolon sayısı = 3 . $2 + 1 = 3$ tesadüf mü? (Bu **rank-nullity teoremi**'nin habercisi — Ders 7-10.)

12.14 Sonraki Ders İçin Hazırlık

Ders 7: $Ax = 0$ Çözme — Pivot Değişkenler ve Özel Çözümler

- Dikdörtgen matrislerde eliminasyon (echelon form U).

- **Rank** = pivot sayısı.
- **Pivot vs serbest değişkenler** ($n - r$ tane serbest).
- **Özel çözümler** (her serbest değişken için).
- İndirgenmiş satır echelon formu R .

⚠ Ders 7 öncesi

- Egzersiz 5 (rank-nullity sezgisi) kritik.
- `null_space` ile birkaç matris dene.
- Ana cümleyi içselleştir.

12.15 Anahtar Kavramlar (Cheat Sheet)

Kavram	Tanım	Strang'da
Birleşim	$P \cup L$ genelde alt-uzay DEĞİL	6m47
Kesişim	$S \cap T$ her zaman alt-uzay	11m05
$C(A)$	Kolonların kombinasyonları; \mathbb{R}^m	11m46
Çözülebilirlik	$A\mathbf{x} = \mathbf{b} \iff \mathbf{b} \in C(A)$	23m39
Pivot kolon	Bağımsız; bağımlı kolon uzaya katkı yok	27m02
$N(A)$	$A\mathbf{x} = \mathbf{0}$ çözümleri; \mathbb{R}^n	29m16
$N(A)$ ispat	Dağılmadan iki satır	38m47
$\mathbf{b} \neq \mathbf{0}$ çözümleri	Alt-uzay değil (orijini içermez)	42m38
İki tanım	Kombinasyon / denklem	44m53

12.16 ML Bağlantıları Özeti

💡 7 köprü

1. $C(W)$ = **erişilebilirlik** → Lineer katmanın üretebileceği çıktılar.
2. $N(W)$ = **kernel / fazlalık** → Bilgi kaybı yönleri; weight decay / LoRA bunları kontrol.
3. **Pivot sayısı = rank** → Modelin gerçek temsil boyutu.
4. $\mathbf{b} \in C(A)$ **testi** → **least squares** → Gürültülü/aşırı belirli sistemlerde projeksiyon.
5. **Birleşim vs kesişim** → Kısıtların kesişimi lineer; “ya/ya” lineer-olmayan zorluk.
6. **Rank + null = kolon sayısı** → Rank-nullity (Ders 7-10); parametre sayma.
7. $N(W) \neq \{\mathbf{0}\}$ = **aşırı parametrisasyon** → Modern derin ağların kaderi; genelleme gizemi.

! Tek bir şey alıp gideceksen

Her matris iki alt-uzay doğurur. $C(A)$ “hangi \mathbf{b} 'leri üretebilirim” sorusu; $N(A)$ “ $A\mathbf{x} = \mathbf{0}$ 'ı hangi \mathbf{x} 'ler çözer” sorusu. Biri çıktı erişilebilirliği, diğeri girdi serbestliği — ikisi de gerçek alt-uzaylar.

13 $Ax = 0$ Çözme — Pivot Değişkenler ve Özel Çözümler

Null uzayı için algoritma — rank, serbest değişkenler, rref

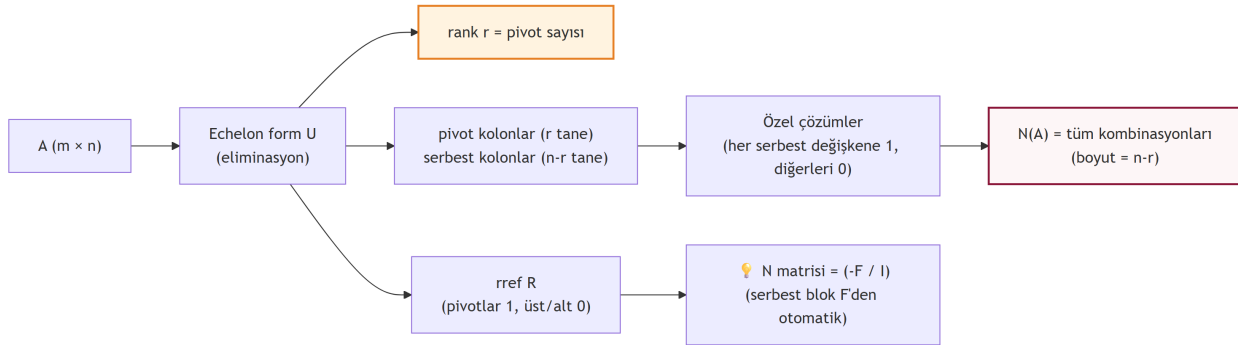
i Bölüm bilgisi

- **Strang'in videosu:** [YouTube — Lecture 7: Solving \$Ax = 0\$](#) (≈43 dk)
- **OCW sayfası:** [MIT 18.06SC — Lecture 7](#)
- **Okuma süresi:** ≈40 dk

13.1 Bu Derste Ne Var?

Ders 6'da null uzayını gözle bulduk. Ders 7 bunu **algoritmaya** çeviriyor.

1. **Dikdörtgen eliminasyon** → echelon form U .
2. **Rank** = pivot sayısı.
3. **Pivot vs serbest değişkenler** → özel çözümler.
4. **rref (R)** → null uzayı doğrudan oku.



Şekil 13.1: $Ax = 0$ algoritması: $A \rightarrow U \rightarrow R \rightarrow$ null uzayı bazı (özel çözümler).

💡 Builder Notu — Rank ve ML

- **Rank** = matrisin gerçek bilgi boyutu. `np.linalg.matrix_rank` bunu sayar. Düşük rank = sıkıştırılabilir → **LoRA**'nın temeli.
- **Serbest değişkenler** = serbestlik dereceleri. Az-belirtilmiş ($n > m$) sistemlerde çözüm tek değil, alt-uzay → modern aşırı-parametrize ağların sonsuz çözümünün kökü.

- **Özel çözümler** = null uzayı bazı. `scipy.linalg.null_space` bu vektörleri döndürür.
- **rank + null boyutu** = n — rank-nullity teoremi; modelin gerçek serbestlik derecelerini sayma.

13.2 Eliminasyon Null Uzayını Korur

“When I subtract a multiple of one equation from another, I’m not changing the null space.” — Strang, 2:42

Sağ taraf hep $\mathbf{0}$ — sıfırdan herhangi katını çıkarsan yine $\mathbf{0}$. Eliminasyon kolon uzayını değiştirir ama null uzayını korur. Bu yüzden $Ax = \mathbf{0}$ için güvenle U 'ya indirgeriz.

13.3 Dikdörtgen Eliminasyon → Echelon Form U

Strang'ın örneği:

$$A = \begin{pmatrix} 1 & 2 & 2 & 2 \\ 2 & 4 & 6 & 8 \\ 3 & 6 & 8 & 10 \end{pmatrix}$$

İlk pivot 1. $r_2 - 2r_1, r_3 - 3r_1$:

$$\rightarrow \begin{pmatrix} 1 & 2 & 2 & 2 \\ 0 & 0 & 2 & 4 \\ 0 & 0 & 2 & 4 \end{pmatrix}$$

(2, 2)'de **sıfır**, altında da sıfır → bu kolonda iş yok, sonrakine geç. (2, 3) pivot = 2. $r_3 - r_2$:

$$U = \begin{pmatrix} 1 & 2 & 2 & 2 \\ 0 & 0 & 2 & 4 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Echelon form — merdiven gibi iner. Sıfır satırı, $r_3 = r_1 + r_2$ bağımlılığını gösterir.

13.4 Rank, Pivot Kolonlar, Serbest Kolonlar

U 'da **iki pivot**: (1, 1) ve (2, 3).

$$\text{rank}(A) = \text{pivot sayısı} = 2$$

- **Pivot kolonlar (1, 3)**: karşılık gelen $x_1, x_3 \rightarrow$ **pivot değişkenler**.
- **Serbest kolonlar (2, 4)**: $x_2, x_4 \rightarrow$ **serbest değişkenler**.

13.5 Özel Çözümler — Serbest Değişkene 1 Ver

U 'dan iki denklem (sağ taraf 0):

$$x_1 + 2x_2 + 2x_3 + 2x_4 = 0, \quad 2x_3 + 4x_4 = 0$$

Strateji: serbest değişkenlere değer ata, pivot değişkenleri geri yerine koy.

Özel çözüm 1: $x_2 = 1, x_4 = 0 \rightarrow x_3 = 0 \rightarrow x_1 = -2$.

$$\mathbf{s}_1 = \begin{pmatrix} -2 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

(Anlamı: $-2\mathbf{c}_1 + \mathbf{c}_2 = \mathbf{0} \rightarrow \mathbf{c}_2 = 2\mathbf{c}_1$.)

Özel çözüm 2: $x_2 = 0, x_4 = 1 \rightarrow x_3 = -2 \rightarrow x_1 = 2$.

$$\mathbf{s}_2 = \begin{pmatrix} 2 \\ 0 \\ -2 \\ 1 \end{pmatrix}$$

Tam null uzayı:

$$N(A) = c_1\mathbf{s}_1 + c_2\mathbf{s}_2, \quad c_1, c_2 \in \mathbb{R}$$

\mathbb{R}^4 içinde **2 boyutlu** alt-uzay.

13.6 $n - r$ Formülü

serbest değişken sayısı = $n - r$

Örnekte $n = 4, r = 2 \rightarrow 2$ özel çözüm. **Rank-nullity teoremi:**

$$\dim N(A) + \text{rank}(A) = n$$

rank = 2

rref R =

Matrix([[1, 2.0000000000000000, 0, -2.0000000000000000], [0, 0, 1, 2.0000000000000000], [0, 0, 0, 0]])

pivot kolonlar: (0, 2)

özel çözümler:

Matrix([[-2.0000000000000000, 1, 0, 0]])

Matrix([[2.0000000000000000, 0, -2.0000000000000000, 1]])

13.7 Reduced Row Echelon Form (rref) R

U 'dan iki ek işlem:

1. Pivotların **üstünü** de temizle.
2. Her pivotu 1 yap.

$$r_2/2 \rightarrow (0, 0, 1, 2) \cdot r_1 - 2r_2 \rightarrow (1, 2, 0, -2):$$

$$R = \begin{pmatrix} 1 & 2 & 0 & -2 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Pivot kolonlarda (1, 3) birim matris. Serbest kolonlar (2, 4) “serbest” bilgiyi taşır.

13.8 Null Uzayı Matrisi — $N = \begin{pmatrix} -F \\ I \end{pmatrix}$ Kalıbı

R 'yi pivot ve serbest kolonlara ayır:

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad F = \begin{pmatrix} 2 & -2 \\ 0 & 2 \end{pmatrix}$$

$Rx = 0 \rightarrow$ (pivot değişkenler) + $F \cdot$ (serbest) = 0 \rightarrow pivot = $-F \cdot$ serbest. Serbest tarafa I koyarsan, pivot tarafa $-F$ gelir:

$$N \text{ matrisi} = \begin{pmatrix} -F \\ I \end{pmatrix} = \begin{pmatrix} -2 & 2 \\ 0 & -2 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}$$

(Satırlar pivot+serbest sırasında dizilir.) Daha önce elle bulduklarımızla aynı.

13.9 $\text{rank}(A) = \text{rank}(A^T)$

A^T (4×3) için aynı algoritmayı uygula \rightarrow 2 pivot çıkar. **Satır rank = kolon rank** — boyut teorisinin köşe taşı.

“The number of pivot columns for A and A transpose are the same.” — Strang, 37:11

13.10 Bu Dersin Özeti

1. Eliminasyon null uzayını korur.
2. **Echelon U , rank $r = \text{pivot sayısı}$.**
3. **Pivot / serbest kolonlar.**
4. **Özel çözümler:** serbest = 1, diğerleri = 0.
5. $n - r$ serbest değişken (rank-nullity).
6. **rref $R \rightarrow$ null uzayını doğrudan okutur.**
7. $N = \begin{pmatrix} -F \\ I \end{pmatrix}$ kalıbı.
8. $\text{rank}(A) = \text{rank}(A^T)$.

! Tek bir cümle

$Ax = \mathbf{0}$ 'ı çözmek artık algoritma: U 'ya in, rank r 'yi bul, $n - r$ serbest değişkenin her birine sırayla 1 ver, özel çözümleri üret. $N(A) =$ bu özel çözümlerin tüm kombinasyonları.

13.11 Kontrol Soruları

i Soru 1: $A = ((1,2,1),(2,4,3))$ — echelon form, rank, pivot/serbest kolonlar?

$$r_2 - 2r_1:$$

$$U = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

Rank = 2. Pivot kolonlar: 1, 3. Serbest: 2. Serbest değişken sayısı $n - r = 1$.

i Soru 2: Aynı A için $N(A)$ özel çözümler bul.

$$x_2 = 1 \rightarrow x_3 = 0 \rightarrow x_1 + 2 = 0 \rightarrow x_1 = -2.$$

$$\mathbf{s} = (-2, 1, 0)^T, \quad N(A) = c(-2, 1, 0)^T$$

\mathbb{R}^3 'te orijinden geçen doğru. ($\mathbf{c}_2 = 2\mathbf{c}_1$.)

i Soru 3: 3×3 matriste rank 3 ise $N(A)$? Ya rank 2 ise?

- **rank = 3:** $n - r = 0 \rightarrow N(A) = \{\mathbf{0}\}$. Tek çözüm $\mathbf{x} = \mathbf{0}$; matris tersinir.
- **rank = 2:** $n - r = 1 \rightarrow N(A)$ bir boyutlu (doğru). Matris singular.

Özet: tam rank $\iff N(A) = \{\mathbf{0}\} \iff$ tersinir.

i Soru 4: W 'nin rank'ı n 'den küçükse model hakkında ne söyler?

$r < n \rightarrow n - r > 0$ serbest yön. W n boyutlu girdiyi r boyuta indiriyor; $n - r$ **boyut kaybı**.

- Bu yönlere parametreler çıktıyı etkilemiyor \rightarrow fazlalık.
- $W \approx UV^T$ (r -rank) daha az parametreyle aynı \rightarrow **LoRA**.
- Eğitimde bazı yönlerde kayıp düz \rightarrow regularization belirler.

Kısaca: " $r < n =$ model görüldüğünden küçük dönüşüm yapıyor."

13.12 Egzersizler

Egzersiz 1. $A = \begin{pmatrix} 1 & 2 & 0 & 1 \\ 0 & 0 & 1 & 2 \\ 1 & 2 & 1 & 3 \end{pmatrix} \rightarrow$ echelon, rank, pivot/serbest kolonlar, iki özel çözüm.

Egzersiz 2. $A = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 8 \end{pmatrix} \rightarrow$ tam rref R ve null uzayı matrisi $\begin{pmatrix} -F \\ I \end{pmatrix}$.

Egzersiz 3. 3×5 matris, rank 3. Kaç özel çözüm? $N(A)$ hangi \mathbb{R}^k 'de, kaç boyutlu?

Egzersiz 4. (Python)

rank = 2

rref R =

```
Matrix([[1, 2.0000000000000000, 0, -2.0000000000000000], [0, 0, 1, 2.0000000000000000], [0, 0, 0, 0]])
```

pivot kolonlar: (0, 2)

özel çözümler:

```
Matrix([[-2.0000000000000000, 1, 0, 0]])
```

```
Matrix([[2.0000000000000000, 0, -2.0000000000000000, 1]])
```

Egzersiz 5. İspatla: A ($m \times n$), rank $r \rightarrow N(A)$ boyutu $n - r$. $n > m$ ise $N(A) \neq \{0\}$ (sıfırdan farklı çözüm hep var). Bu, Ders 8 ve ML'de aşırı-parametrizasyonun temeli.

13.13 Sonraki Ders İçin Hazırlık

Ders 8: $Ax = b$ Çözme — Tam Çözüm ve Rank

- Sağ taraf sıfırdan farklı: çözülebilirlik testi ($b \in C(A)$).
- **Tam çözüm = özel çözüm $x_p + N(A)$.**
- Augmented matris $[A \mid b]$ ile eliminasyon.
- Rank'in çözüm sayısına etkisi (0, 1, ∞).


 Ders 8 öncesi

- Egzersizleri çöz, özellikle 5 (rank-nullity ispat).
- `sympy.Matrix(A).rref()` ve `.nullspace()` ile birkaç matris dene.
- Ana cümleyi tekrar oku.


13.14 Anahtar Kavramlar (Cheat Sheet)

Kavram	Tanım	Strang'da
Eliminasyon null'ı korur	$Ax = 0$ çözümleri değişmez	2m42
Echelon U	Dikdörtgen eliminasyon merdiveni	6m12
Rank r	Pivot sayısı	6m44
Pivot/serbest kolon	Pivotlular pivot, kalanlar serbest	8m04
Serbest değişken	İstedüğün değeri verirsin ($n - r$ tane)	8m43
Özel çözüm	Serbest = 1, diğerleri = 0	15m05
$n - r$ formülü	Serbest = özel çözüm sayısı	17m28
rref R	Pivot 1, üst/alt 0	19m25
N matrisi	$\begin{pmatrix} -F \\ I \end{pmatrix}$ kalıbı	30m54
$\text{rank}(A) = \text{rank}(A^T)$	Satır rank = kolon rank	37m11

13.15 ML Bağlantıları Özeti

 7 köprü

1. **Rank = gerçek bilgi boyutu** \rightarrow `matrix_rank`; düşük rank \rightarrow LoRA.
2. **Serbest değişkenler = serbestlik dereceleri** \rightarrow Aşırı-parametrize ağların sonsuz çözümü.
3. **Özel çözümler = null bazı** \rightarrow `null_space`, dönüşümün kör yönleri.
4. **rref / Gauss** \rightarrow Her solver'in içi; "singular matrix" uyarılarını anlama.
5. $n > m \rightarrow N(A) \neq \{0\}$ \rightarrow Parametre > veri rejimi; derin öğrenme.
6. **Rank-nullity** \rightarrow Modelin gerçek serbestlik dereceleri.
7. **$\text{rank}(A) = \text{rank}(A^T)$** \rightarrow Boyut teorisinin köşe taşı.

 Tek bir şey alıp gideceksen

$Ax = 0$ artık tahmin değil, **algoritma**: U 'ya in, rank r bul, $n - r$ serbest değişkenin her birine sırayla 1 ver, özel çözümleri üret. $N(A) =$ bu özel çözümlerin tüm kombinasyonları; rank, matrisin tüm karakterini özetleyen tek sayı.

14 Ax = b Çözme — Tam Çözüm ve Rank

$x = x_p + x_n$; rank her şeyi söyler

i Bölüm bilgisi

- **Strang'ın videosu:** [YouTube — Lecture 8: Solving Ax = b: Row Reduced Form R](#) (≈47 dk)
- **OCW sayfası:** [MIT 18.06SC — Lecture 8](#)
- **Okuma süresi:** ≈40 dk

14.1 Bu Derste Ne Var?

Ders 7'de $Ax = 0$ 'ı çözdük. Ders 8 sağ tarafı sıfırdan farklı yapıp $Ax = \mathbf{b}$ 'nin **tam çözümünü** kuruyor.

1. **Çözülebilirlik:** $\mathbf{b} \in C(A)$.
2. **Tam çözüm = $x_p + x_n$:** bir particular + tüm null uzayı.
3. **Rank her şeyi söyler:** r ile m, n ilişkisi çözüm sayısını $(0, 1, \infty)$ belirler.

“The rank tells you everything about the number of solutions.” — Strang, 46:55

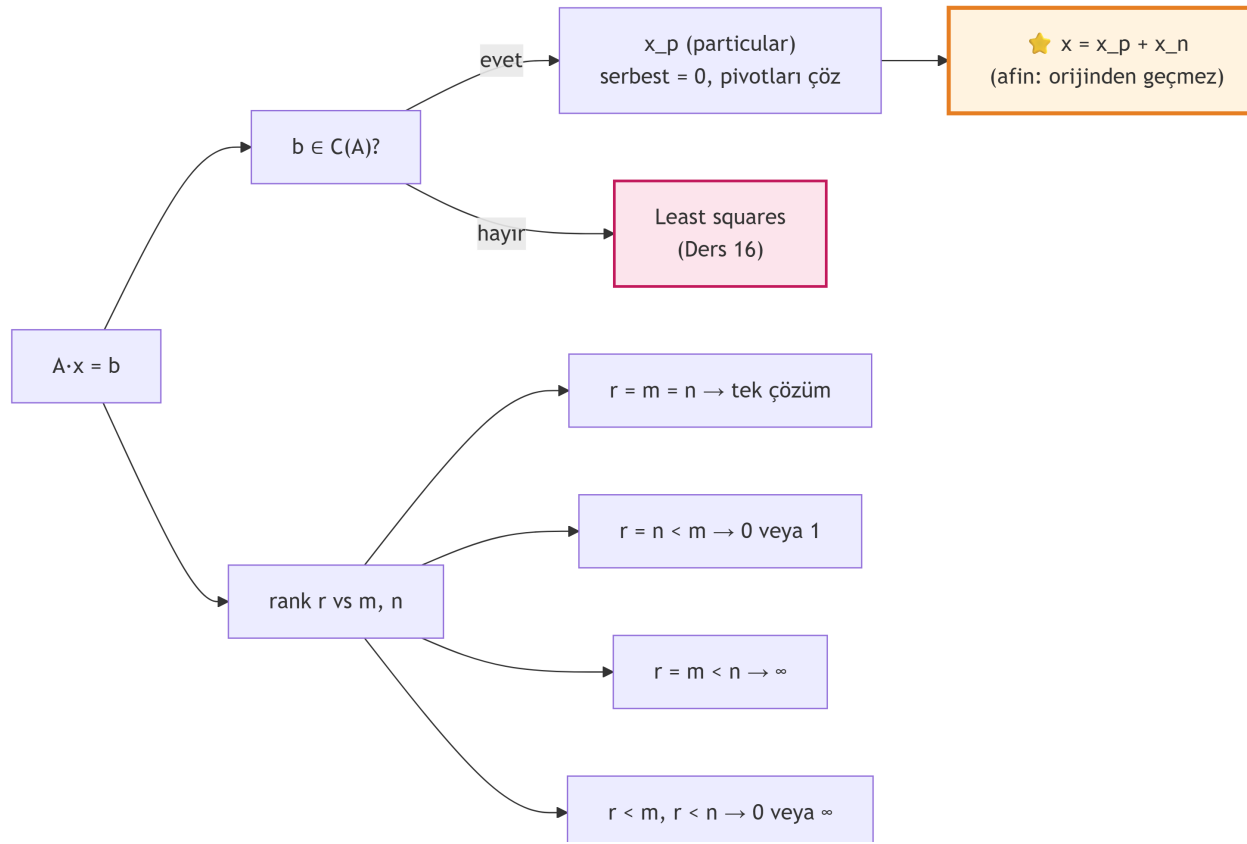
💡 Builder Notu — ML'de Tam Çözüm

- $x_p + x_n$ ML'in her yerinde: regularization ($\min \|x\|$) sonsuz çözüm arasından **en küçük normlu**'yu seçer.
- **Full column rank** ($r = n$) = injektif → girdi tek biçimde kodlanır; $A^T A$ tersinir.
- **Full row rank** ($r = m$) = sürjektif → her hedef üretilebilir, sonsuz çözüm; **aşırı-parametrize ağların** tipik hali.
- **Az/çok-belirtilmiş** ayrımı: $n > m$ (deep learning rejimi) vs $m > n$ (klasik regresyon).

14.2 Augmented Matrix [A | b]

\mathbf{b} 'yi A 'ya fazladan kolon olarak ekle. Strang'ın örneği, $\mathbf{b} = (1, 5, 6)^T$:

$$[A | \mathbf{b}] = \left[\begin{array}{cccc|c} 1 & 2 & 2 & 2 & 1 \\ 2 & 4 & 6 & 8 & 5 \\ 3 & 6 & 8 & 10 & 6 \end{array} \right]$$



Şekil 14.1: Tam çözüm = particular + null. Rank–m–n ilişkisi çözüm sayısını belirler.

Eliminasyon ($r_2 - 2r_1, r_3 - 3r_1, r_3 - r_2$):

$$\rightarrow \left[\begin{array}{cccc|c} 1 & 2 & 2 & 2 & 1 \\ 0 & 0 & 2 & 4 & 3 \\ 0 & 0 & 0 & 0 & 0 \end{array} \right]$$

Son satır: sol sıfır + sağ sıfır \rightarrow tutarlı.

14.3 Çözülebilirlik

Genel $\mathbf{b} = (b_1, b_2, b_3)$ ile son satır $0 = b_3 - b_2 - b_1$ derdi. $b_1 + b_2 = b_3$ ise çözülebilir.

İki dil:

- **Kolon dili:** $\mathbf{b} \in C(A)$.
- **Satır dili:** Satırların sıfır kombinasyonu ($r_3 - r_2 - r_1 = \mathbf{0}$) \mathbf{b} 'de de sıfır vermeli ($b_3 - b_2 - b_1 = 0$).

“Solvable exactly when \mathbf{b} is in the column space of A .” — Strang, 8:10

Builder Notu: $m > n$ 'de \mathbf{b} neredeyse hiç $C(A)$ 'da olmaz \rightarrow tam çözüm yok \rightarrow **least squares** ($A^T A \mathbf{x} = A^T \mathbf{b}$, Ders 16).

14.4 Particular Çözüm — Serbest Değişkenleri Sıfırla

$x_2 = x_4 = 0$ (serbest). Kalan: $x_1 + 2x_3 = 1, 2x_3 = 3$.

$x_3 = 3/2, x_1 = 1 - 3 = -2$.

$$\mathbf{x}_p = \begin{pmatrix} -2 \\ 0 \\ 3/2 \\ 0 \end{pmatrix}$$

14.5 Tam Çözüm = $\mathbf{x}_p + N(A)$

$A\mathbf{x}_p = \mathbf{b}, A\mathbf{x}_n = \mathbf{0} \rightarrow A(\mathbf{x}_p + \mathbf{x}_n) = \mathbf{b}$. Null uzayı serbestçe eklenir.

Ders 7'den iki özel çözüm $\mathbf{s}_1 = (-2, 1, 0, 0)^T, \mathbf{s}_2 = (2, 0, -2, 1)^T$:

$$\mathbf{x} = \mathbf{x}_p + c_1 \mathbf{s}_1 + c_2 \mathbf{s}_2$$

Dikkat: \mathbf{x}_p 'nin önünde katsayı yok — bir sabit nokta + bir alt-uzay.

Geometri: Çözüm kümesi $\mathbf{x}_p + N(A)$ — orijinden geçmez (afin), kaymış düzlem.

“It’s like a subspace, but it’s been shifted away from the origin.” — Strang, 23:56

rank = 2
 bir çözüm $x = [-0.1379 \ -0.2759 \ 0.1897 \ 0.6552]$
 $A @ x = [1. \ 5. \ 6.]$
 tutarsız b için residual = []

💡 Builder Notu — Regularization

“Bir özel çözüm + homojen çözümler” kalıbı lineer diferansiyel denklemlerde de aynen geçer. ML’de **az-belirtilmiş** sistemlerde sonsuz çözüm; **ridge / L2** ($\min \|x\|$) bu aileden tekini — genelde en küçük normlu olanı — seçer. Gradient descent çoğu zaman implicit olarak min-norm çözüme yakınsar.

14.6 Rank ve Boyut İlişkileri

Her zaman: $r \leq m$ ve $r \leq n$ (her satırda/kolonda en fazla 1 pivot).

Üç önemli durum:

Durum	R biçimi	Çözüm sayısı
$r = m = n$	$R = I$	Her zaman 1
$r = n < m$ (full column)	$\begin{pmatrix} I \\ 0 \end{pmatrix}$	0 veya 1
$r = m < n$ (full row)	$[I \ F]$ (sıfır satır yok)	∞
$r < m, r < n$	$[I, F] +$ sıfır satırlar	0 veya ∞

Özet mantık:

- **Sıfır satır var mı?** → çözülebilirlik koşulu (0 olabilir).
- **Serbest değişken var mı?** → benzersizlik (∞ olabilir).

14.7 Full Column Rank ($r = n$)

Her kolonda pivot, serbest değişken yok → $N(A) = \{0\}$ → çözüm varsa **tek**.

Tipik: “uzun-ince” matris ($m > n$). Örnek 4×2 :

$$A = \begin{pmatrix} 1 & 3 \\ 2 & 1 \\ 6 & 1 \\ 5 & 1 \end{pmatrix} \rightarrow R = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$$

Rastgele $b \rightarrow 0$ çözüm; $b \in C(A) \rightarrow 1$ çözüm.

Builder Notu: = **injektif** kodlama; $A^T A$ tersinir → least squares benzersiz (Ders 16).

14.8 Full Row Rank ($r = m$)

Her satırda pivot, sıfır satır yok \rightarrow her \mathbf{b} için çözüm; $n - m$ serbest değişken \rightarrow sonsuz çözüm.

Tipik: “kısa-geniş” ($m < n$). Önceki örneğin transpozu 2×4 :

$$A^T = \begin{pmatrix} 1 & 2 & 6 & 5 \\ 3 & 1 & 1 & 1 \end{pmatrix} \rightarrow R = [I \mid F]$$

Builder Notu: = sürjektif \rightarrow her hedef üretilir; aşırı-parametrize ağların hali. Sonsuz çözüm = eğitim verisini mükemmel ezberleme kapasitesi \rightarrow regularization / implicit bias şart.

14.9 $r = m = n$ — Tersinir

Kare + tam rank. $R = I$. $N(A) = \{\mathbf{0}\}$, her \mathbf{b} için tek çözüm $\mathbf{x} = A^{-1}\mathbf{b}$.

Chapter 2'nin tüm hikayesi — şimdi onu daha genel rank manzarasının bir köşesi olarak görüyoruz.

14.10 Bu Dersin Özeti

1. **Augmented matrix** $[A \mid \mathbf{b}]$.
2. **Çözülebilirlik:** $\mathbf{b} \in C(A)$.
3. **Particular:** serbest = 0, pivotları çöz.
4. **Tam çözüm:** $\mathbf{x}_p + \mathbf{x}_n$.
5. **Geometri:** afin küme (kaymış alt-uzay).
6. **Rank ilişkileri:** $r \leq m, r \leq n$.
7. **Full column** ($r = n$): 0 veya 1.
8. **Full row** ($r = m$): ∞ .
9. $r = m = n$: tersinir, tek.
10. **Rank her şeyi söyler.**

! Tek bir cümle

$A\mathbf{x} = \mathbf{b}$ tam çözümü = $\mathbf{x}_p + N(A)$; bir sistemin 0, 1 veya ∞ çözümü olduğu sadece **rank** r ile m, n ilişkisine bağlıdır.

14.11 Kontrol Soruları

i Soru 1: $A = ((1,1,1),(0,1,2))$, $b = (3,2)^T$ için x_p (serbest = 0).

Pivotlar kolon 1, 2; serbest x_3 . $x_3 = 0$: $x_1 + x_2 = 3$, $x_2 = 2 \rightarrow x_2 = 2$, $x_1 = 1$.

$$\mathbf{x}_p = (1, 2, 0)^T$$

i Soru 2: Aynı sistemin tam çözümü.

Null uzayı için $x_3 = 1$: $x_2 + 2 = 0 \rightarrow x_2 = -2$; $x_1 - 2 + 1 = 0 \rightarrow x_1 = 1$. $\mathbf{s} = (1, -2, 1)^T$:

$$\mathbf{x} = (1, 2, 0)^T + c(1, -2, 1)^T$$

Nokta + doğru. Sonsuz çözüm.

i Soru 3: 3×5 , rank 3 — $Ax = b$ kaç çözüm?

$r = 3 = m \rightarrow$ **full row rank**. Sıfır satır yok \rightarrow her \mathbf{b} çözülür. $n - r = 2$ serbest \rightarrow null 2 boyutlu.

Sonuç: her \mathbf{b} için **sonsuz** çözüm. 0 imkânsız (varlık garanti), 1 imkânsız (serbest var).

i Soru 4: Az-belirtilmiş sistemde ML neden min-norm seçer?

Çözüm $\mathbf{x}_p + N(A) =$ sonsuz nokta. Bir **kriter** lazım.

Min $\|\mathbf{x}\|$ orijine en yakın noktayı seçer — null uzayı bileşenini, \mathbf{x}_p 'nin null'a dik kısmıyla dengeleyip kalanı atar.

- **Ridge / L2** tam bunu yapar.
- **Gradient descent** (sıfırdan başlatılmışsa) çoğu lineer problemde implicit olarak min-norm'a yakınsar \rightarrow aşırı-parametrize ağların gizemli genellemesinin bir parçası.

14.12 Egzersizler

Egzersiz 1. $A = ((1, 2, 0, 1), (0, 0, 1, 2))$, $\mathbf{b} = (3, 1)^T$ — tam çözüm.

Egzersiz 2. $A = ((1, 1), (1, 2), (1, 3))$ için genel $\mathbf{b} = (b_1, b_2, b_3)$ cinsinden çözülebilirlik koşulu.

Egzersiz 3. Her matris için $(m, n, r) \rightarrow$ çözüm sınıfı:

- (a) 3×3 , rank 3
- (b) 2×4 , rank 2
- (c) 4×2 , rank 2
- (d) 3×4 , rank 2

Egzersiz 4. (Python) `np.linalg.lstsq` ile az/çok-belirtilmiş sistemleri dene; rank ve residual'a bak.

Egzersiz 5. İspatla: A full column rank ($r = n$) ise, $A\mathbf{x} = \mathbf{b}$ 'nin en fazla bir çözümü vardır. (İpucu: iki çözüm $\mathbf{x}_1, \mathbf{x}_2 \rightarrow \mathbf{x}_1 - \mathbf{x}_2 \in N(A) = \{\mathbf{0}\}$.) Ders 9'daki **lineer bağımsızlık** habercisi.

14.13 Sonraki Ders İçin Hazırlık

Ders 9: Lineer Bağımsızlık, Baz ve Boyut — kursun en kilit dersi.

- **Bağımsızlık:** vektörler ne zaman “yeni yön” katar?
- **Span, baz, boyut.**
- Rank ile bağlantı.


 Ders 9 öncesi

- Egzersizleri çöz, özellikle 5 (bağımsızlık habercisi).
- 1stsq ile rank çıktısını gözle.

14.14 Anahtar Kavramlar (Cheat Sheet)

Kavram	Tanım	Strang'da
Augmented $[A \mid \mathbf{b}]$	\mathbf{b} kolon olarak eklenir	3m17
Çözülebilirlik	$\mathbf{b} \in C(A)$	8m10
Particular	Serbest = 0, pivotları çöz	11m50
Tam çözüm	$\mathbf{x}_p + \mathbf{x}_n$	16m20
Kaymış alt-uzay	Afin, orijinden geçmez	23m56
Rank ilişkileri	$r \leq m, r \leq n$	25m09
Full column	$r = n; 0$ veya 1	26m49
Full row	$r = m; \infty$	36m23
$r = m = n$	Tersinir, $R = I$	40m53
Rank her şeyi söyler	Çözüm sayısı = $f(r, m, n)$	46m55

14.15 ML Bağlantıları Özeti

 7 köprü

1. $\mathbf{x}_p + \mathbf{x}_n \rightarrow$ **regularization** \rightarrow Min $\|\mathbf{x}\|$ aile içinden teki seçer.
2. **Full column = injektif** \rightarrow Girdi tek-biçimli kodlanır; $A^T A$ tersinir.
3. **Full row = sürjektif** \rightarrow Aşırı-parametrize ağ; sonsuz çözüm, ezberleme kapasitesi.
4. **Çözülebilirlik** \rightarrow **least squares** \rightarrow $m > n$ 'de en-yakın çözüm (Ders 16).
5. **Rank = etkin boyut** \rightarrow Veri/ağırlık matrisinin gerçek bilgi içeriği; LoRA, PCA.
6. **Afin çözüm kümesi** \rightarrow Optimizasyonda “düz vadi”; null yönlerde kayıp değişmez.
7. **Under/over-determined** \rightarrow $n > m$ (DL) vs $m > n$ (klasik); rank teşhis eder.

! Tek bir şey alıp gideceksen

$Ax = b$ tam çözümü = $x_p + x_n$; çözüm sayısı $(0, 1, \infty)$ yalnız **rank** r ile m, n ilişkisine bağlı — rank sistemin kaderini özetler.

15 Linear Bağımsızlık, Baz ve Boyut

Kursun kilit dersi — dört kavram bir arada

i Bölüm bilgisi

- **Strang'in videosu:** [YouTube](#) — [Lecture 9: Independence, Basis, and Dimension](#) (≈50 dk)
- **OCW sayfası:** [MIT 18.06SC](#) — [Lecture 9](#)
- **Okuma süresi:** ≈40 dk

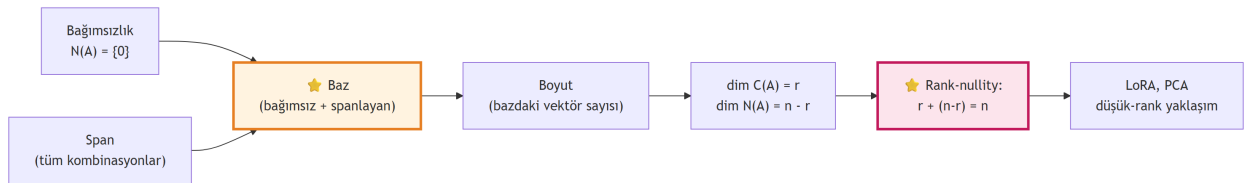
15.1 Bu Derste Ne Var?

“This is a key lecture, this is where we get these ideas of linear independence, the space they span, a basis, and the dimension.” — Strang, 0:21

Şimdiye dek sezgisel kullandığımız dört kelime burada kesin tanım kazanıyor:

1. **Linear bağımsızlık** — hiçbir kombinasyon **0** vermez (sıfır kombinasyonu hariç).
2. **Span** — vektör kümesinin tüm kombinasyonları.
3. **Baz** — bağımsız + spanlayan.
4. **Boyut** — bir bazdaki vektör sayısı.

Temel teorem: $\dim C(A) = r$, $\dim N(A) = n - r \rightarrow$ **rank-nullity**.



Şekil 15.1: Bağımsızlık + span = baz. Baz vektör sayısı = boyut. Boyutlar rank-nullity ile bağlanır.

💡 Builder Notu — ML'in Alfabeti

- **Bağımsızlık = multicollinearity yokluğu.** Bağımlı özellikler $A^T A$ 'yı tekilleştirir.
- **Baz = koordinat sistemi.** PCA/Fourier/wavelet — aynı bilgi, farklı eksenler.
- **Boyut = içsel boyut.** Manifold hipotezi: yüksek boyutlu veri düşük boyutta yaşar.
- **rank = dim C(A)** → veri matrisinin gerçek bilgi içeriği.

15.2 $n > m \implies$ Null Uzayı Boş Değil

Kolon sayısı satır sayısından fazlaysa, $A\mathbf{x} = \mathbf{0}$ 'ın sıfırdan farklı çözümü **her zaman** vardır. En fazla m pivot olur; $n > m \rightarrow$ en az bir serbest değişken $\rightarrow N(A) \neq \{\mathbf{0}\}$.

“More unknowns than equations, then there are some non-zero x 's such that Ax is zero.” — Strang, 2:15

Bu, “ m boyutlu uzayda m 'den fazla vektör daima bağımlıdır” sonucunu getirecek.

15.3 Lineer Bağımsızlık

$\mathbf{x}_1, \dots, \mathbf{x}_n$ bağımsızdır \iff

$$c_1\mathbf{x}_1 + \dots + c_n\mathbf{x}_n = \mathbf{0} \implies c_1 = \dots = c_n = 0$$

Sıfırı elde etmenin **tek yolu** her şeyi 0 ile çarpmak.

Örnekler:

- \mathbf{v} ve $2\mathbf{v}$: bağımlı ($2\mathbf{v} - 1 \cdot (2\mathbf{v}) = \mathbf{0}$).
- İçinde $\mathbf{0}$ olan her küme bağımlı ($0 \cdot \mathbf{v} + 5 \cdot \mathbf{0} = \mathbf{0}$).
- \mathbb{R}^2 'de 3 vektör: her zaman bağımlı ($n > m$).

15.4 Matris Dili — Bağımsızlık $\iff N(A) = \{\mathbf{0}\}$

Vektörleri A 'nın kolonları yap. Kombinasyon = $A\mathbf{c}$. Dolayısıyla:

$$\text{kolonlar bağımsız} \iff N(A) = \{\mathbf{0}\} \iff r = n$$

“The columns are independent if the null space of A is the zero vector.” — Strang, 14:30

Kare matris için: bağımsız \iff tersinir.

Builder Notu: `np.linalg.matrix_rank(A) == n` \rightarrow kolonlar bağımsız. Bağımlı kolonlar $A^T A$ 'yı tekilleştirir \rightarrow regresyon kütüphaneleri ya kolonu atar ya da **ridge** (λI ekleyerek) by-pass eder.

15.5 Span

$$\text{span}(\mathbf{v}_1, \dots, \mathbf{v}_l) = \{c_1\mathbf{v}_1 + \dots + c_l\mathbf{v}_l : c_i \in \mathbb{R}\}$$

Vektörleri içeren **en küçük** alt-uzay. Bir matrisin kolonları $C(A)$ 'yı spanlar.

Span ve bağımsızlık birbirinden bağımsız özellikler: spanlamak için yeterli, bağımlı olabilir; ya da bağımsız olabilir, spanlamaz. **Baz** ikisini birden ister.

15.6 Baz (Basis)

Bir uzayın **bazı**: aynı anda

1. **Bağımsız** (fazla vektör yok).
2. **Spanlayan** (eksik vektör yok).

“A basis is a sequence of vectors with two properties: they are independent, and they span the space.” — Strang, 22:14

\mathbb{R}^3 standart bazı: $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3 \rightarrow$ birim matris.

Test: \mathbb{R}^n 'de n vektör baz \iff kolon yapan $n \times n$ matris tersinir.

 **Builder Notu** — Baz = Koordinat Sistemi

Baz seçimi = koordinat sistemi seçimi. Aynı embedding'i PCA bazında, Fourier modlarında, wavelet'lerde göstermek aynı bilgiyi farklı eksenlerde sunar. **PCA** verinin en çok değiştiği yönleri baz yapar; sıkıştırmanın özü budur (Ders 31 baz değişimi).

15.7 Baz Tek Değil — Ama Boyut Sabit

\mathbb{R}^3 için sonsuz baz var (her tersinir 3×3). Ama her bazda tam **3** vektör.

“There are many, many bases, but they all have the same number of vectors.” — Strang, 34:18

Sezgi: Az vektörle spanlayamazsın (boşluk kalır); çok vektörle bağımsız olamazsın (tekrar olur). “Tam doğru” sayı = **boyut**.

15.8 Boyut

$\dim(\text{uzay}) =$ bir bazdaki vektör sayısı

- $\dim \mathbb{R}^n = n$.
- \mathbb{R}^3 'te düzlem $\rightarrow 2$, doğru $\rightarrow 1$, $\{\mathbf{0}\} \rightarrow 0$.

Boyut uzayın özelliğidir, matrisin değil. “Alt-uzayın boyutu” \rightarrow evet. “Matrisin boyutu” \rightarrow yok.

15.9 Temel Teorem — rank-nullity

$$A = \begin{pmatrix} 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 1 \\ 1 & 2 & 3 & 1 \end{pmatrix}$$

Pivot kolonlar 1 ve 2; kolon 3 = 1 + 2, kolon 4 = kolon 1.

(1) $\dim C(A) = r$. Pivot kolonlar $C(A)$ için baz oluşturur.

(2) $\dim N(A) = n - r$. Özel çözümler $N(A)$ için baz.


$$\dim C(A) + \dim N(A) = r + (n - r) = n$$

Rank-nullity teoremi — Ders 7'deki sezginin tam ispatı.

$$\text{rank} = \dim C(A) = 2$$

$$\dim N(A) = n - r = 2$$


$$\text{rank-nullity: } r + (n-r) = 4 == n = 4$$

 **Builder Notu — Boyut Muhasebesi**

$\dim C(A) = r =$ etkin boyut (`matrix_rank`); $\dim N(A) = n - r =$ fazlalık. Bir lineer katmanın “kaç boyut taşıdığı + kaç boyut yuttuğu = girdi boyutu” muhasebesidir; **LoRA, PCA** düşük-rank yaklaşımlar bunu sömürür.

15.10 Bu Dersin Özeti

1. $n > m \implies N(A) \neq \{\mathbf{0}\}$.
2. **Bağımsızlık** — hiçbir sıfır-olmayan kombinasyon $\mathbf{0}$ vermez.
3. **Matris dili** — bağımsız $\iff N(A) = \{\mathbf{0}\} \iff r = n$.
4. **Span, baz** (bağımsız + spanlayan), **boyut** (baz vektör sayısı).
5. \mathbf{R}^n 'de **n vektör baz** \iff kare matris tersinir.
6. **dim C(A) = r, dim N(A) = n - r**.
7. **Rank-nullity: r + (n - r) = n**.

 **Tek bir cümle**

Baz = bağımsızlık + spanlama (ne eksik, ne fazla); **boyut** = baz vektör sayısı. Bir matris için $\dim C(A) = r$, $\dim N(A) = n - r$ — lineer cebirin boyut muhasebesinin kalbi.

15.11 Kontrol Soruları

i Soru 1: $(1,1,0)$, $(0,1,1)$, $(1,0,-1)$ bağımsız mı?

$1 \cdot (1, 1, 0) - 1 \cdot (0, 1, 1) = (1, 0, -1) = 3.$ vektör. Yani:

$$\mathbf{v}_1 - \mathbf{v}_2 - \mathbf{v}_3 = \mathbf{0}$$

Bağımlı. Matris tekil, rank 2, \mathbb{R}^3 spanlamaz.

i Soru 2: $A = ((1,2,3),(2,4,1),(3,6,4))$ için $C(A)$ bazı ve boyutu.

$\mathbf{c}_2 = 2\mathbf{c}_1 \rightarrow$ bağımlı. \mathbf{c}_3 bağımsız. Pivot kolonlar 1 ve 3:

$$\text{baz} = \{(1, 2, 3)^T, (3, 1, 4)^T\}, \quad \dim C(A) = r = 2$$

\mathbb{R}^3 içinde bir düzlem.

i Soru 3: Boyut 3 olan uzayda 3 bağımsız vektör — otomatik baz mı?

Evet. Spanlamasalar, span dışından bir vektör eklenip 4 bağımsız vektör elde edilirdi; ama 3 boyutta 4 bağımsız vektör olamaz \rightarrow çelişki \rightarrow spanlar.

Kural: Doğru sayıda (= boyut) vektör varsa, “bağımsızlık” ile “spanlama” birbirini getirir — sadece birini kontrol et.

i Soru 4: Bağımlı özellikler regresyonda ne sorun çıkarır?

$C(A)$ boyutu $< n$, $N(A) \neq \{\mathbf{0}\}$:

- $A^T A$ **tekil** \rightarrow normal denklemler $(A^T A)^{-1} A^T \mathbf{b} \rightarrow$ tek çözüm yok; katsayılar belirsiz.
- Bir özelliğin ağırlığını artırıp diğerininkini azaltarak aynı tahmin \rightarrow null yönleri.
- **Çözümler:** bağımlı kolonu at, ya da **ridge** (λI ekle) ile $A^T A + \lambda I$ 'yi tersinir yap, ya da **PCA** ile bağımsız baza geç.

İçsel boyut (rank) kaç bağımsız özelliğin olduğunu söyler.

15.12 Egzersizler

Egzersiz 1. $(1, 2, 3)$, $(1, 0, 1)$, $(1, 4, 5)$ bağımsız mı? Matrise koy, rank bul.

Egzersiz 2. $A = \begin{pmatrix} 1 & 2 & 1 & 1 \\ 2 & 4 & 3 & 0 \\ 1 & 2 & 2 & -1 \end{pmatrix} \rightarrow C(A)$ ve $N(A)$ için baz ve boyut. Rank-nullity'yi doğrula.

Egzersiz 3. \mathbb{R}^4 'te 5 vektör bağımsız olabilir mi? 3 vektör \mathbb{R}^4 'ü spanlayabilir mi? (Boyut argümanı.)

Egzersiz 4. (Python) Rank-nullity'yi farklı matrislerde doğrula.

Egzersiz 5. İspatla: $n \times n$ matris tersinir \iff kolonları \mathbb{R}^n için baz. (İpucu: tersinir $\iff N(A) = \{\mathbf{0}\}$ \iff bağımsız; n bağımsız vektör \mathbb{R}^n 'i spanlar.)

15.13 Sonraki Ders İçin Hazırlık

Ders 10: Dört Temel Alt-Uzay

Bir matrisin **dört** alt-uzayı:

- $C(A) \subset \mathbb{R}^m$, boyut r
- $N(A) \subset \mathbb{R}^n$, boyut $n - r$
- $C(A^T) \subset \mathbb{R}^n$, boyut r (sıra uzayı)
- $N(A^T) \subset \mathbb{R}^m$, boyut $m - r$ (sol null)

⚠ Ders 10 öncesi

- Egzersizleri çöz, özellikle 5 (tersinir \iff baz).
- `matrix_rank` ile rank-nullity'yi birkaç matriste doğrula.

15.14 Anahtar Kavramlar (Cheat Sheet)

Kavram	Tanım	Strang'da
$n > m$ gerçeği	Kolon $>$ satır \rightarrow null boş değil	2m15
Bağımsızlık	Hiçbir non-zero komb. = $\mathbf{0}$	6m06
Matris dili	Bağımsız $\iff N(A) = \{\mathbf{0}\}$ $\iff r = n$	14m30
Span	Tüm kombinasyonlar	19m24
Baz	Bağımsız + spanlayan	22m14
Baz \iff tersinir	\mathbb{R}^n 'de n vektör baz \iff kare tersinir	30m06
Baz sayısı sabit	Tüm bazlar aynı boyutta	34m18
Boyut	Bazdaki vektör sayısı	36m33
$\dim C(A) = r$	Pivot kolonlar = $C(A)$ bazı	41m55
$\dim N(A) = n - r$	Özel çözümler = $N(A)$ bazı	49m09

15.15 ML Bağlantıları Özeti

💡 7 köprü

1. **Bağımsızlık = multicollinearity yokluğu** → $A^T A$ tersinir, katsayılar belirli.
2. **Baz = koordinat sistemi** → PCA, Fourier, wavelet, öğrenilmiş embedding.
3. **Boyut = içsel boyut** → Manifold hipotezi.
4. **rank = $\dim C(A)$ = etkin boyut** → `matrix_rank`; veri/ağırlığın gerçek bilgisi.
5. **Span = erişilebilir uzay** → $C(W)$.
6. **Rank-nullity** → Parametre muhasebesi: taşınan + yutulan = girdi.
7. **Düşük rank = sıkıştırma** → $\dim C(A) \ll n$ → LoRA, PCA.

❗ Tek bir şey alıp gideceksen

Baz, bir uzayı tam doğru sayıda bağımsız vektörle tanımlar; **boyut** bu sayıdır, seçilen baza bağlı değildir. Bir matris için $\dim C(A) = r$, $\dim N(A) = n - r$ — boyut muhasebesinin kalbi.

16 Dört Temel Alt-Uzay

Bir matrisin tam geometrisi — $C(A)$, $N(A)$, $C(A^T)$, $N(A^T)$

Bölüm bilgisi

- **Strang'in videosu:** [YouTube — Lecture 10: The Four Fundamental Subspaces](#) (≈50 dk)
- **OCW sayfası:** [MIT 18.06SC — Lecture 10](#)
- **Okuma süresi:** ≈40 dk

16.1 Bu Derste Ne Var?

“This is the heart of this approach to linear algebra, to see these four subspaces.” — Strang, 4:44

Her $m \times n$ matris **dört temel alt-uzay** doğurur:

1. $C(A)$ — kolon uzayı, \mathbb{R}^m , boyut r .
2. $N(A)$ — null uzayı, \mathbb{R}^n , boyut $n - r$.
3. $C(A^T)$ — satır uzayı, \mathbb{R}^n , boyut r .
4. $N(A^T)$ — sol null uzayı, \mathbb{R}^m , boyut $m - r$.

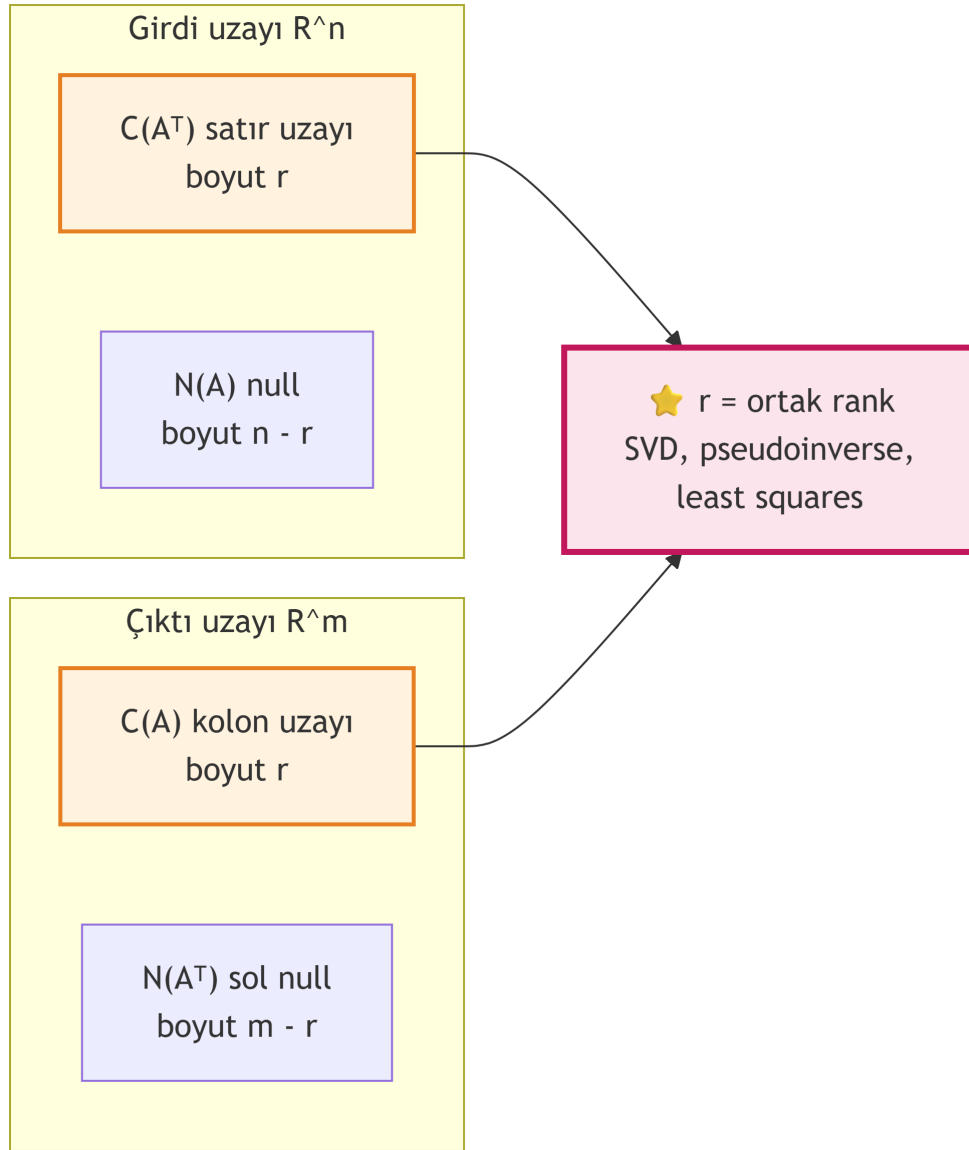
Harika gerçek: satır rank = kolon rank = r .

Builder Notu — Dört Alt-Uzay = Dönüşüm Geometrisi

- **Satır uzayı = etkin girdi yönleri;** null = yutulan. $\mathbf{x} = \mathbf{x}_{\text{row}} + \mathbf{x}_{\text{null}}$, $A\mathbf{x}$ sadece satır bileşenine bağlı.
- **Kolon uzayı = erişilen çıktı yönleri;** sol null = least squares **artıkları**.
- **Satır rank = kolon rank** → SVD'de tek bir r .
- **Matris uzayları** → ağırlık tensörleri için soyut temel.

16.2 Dört Alt-Uzay — Yaşadıkları Uzaylar

Hangi büyük uzayda yaşarlar?



Şekil 16.1: Dört alt-uzay: girdi R^n (satır + null), çıktı R^m (kolon + sol null). Boyut toplamları n ve m .

Alt-uzay	Uzay	Boyut
$C(A)$	\mathbb{R}^m	r
$C(A^T)$	\mathbb{R}^n	r
$N(A)$	\mathbb{R}^n	$n - r$
$N(A^T)$	\mathbb{R}^m	$m - r$

İki toplam:

$$\dim C(A^T) + \dim N(A) = r + (n - r) = n$$

$$\dim C(A) + \dim N(A^T) = r + (m - r) = m$$

\mathbb{R}^n satır + null'a, \mathbb{R}^m kolon + sol null'a bölünür.

16.3 Satır Uzayı $C(A^T)$

Satırlar yatay, kolonlar dikey. Çözüm: **transpoze et**. A 'nın satırları A^T 'nin kolonları olur:

$$\text{satır uzayı} = C(A^T)$$

"We just got some vectors that were lying down to stand up." — Strang, 6:55

Builder Notu: Girdi tarafındaki "etkin" alt-uzay. \mathbf{x} = satır uzayı bileşeni + null uzayı bileşeni; $A\mathbf{x}$ sadece satır bileşenine bağlı. **SVD ve pseudoinverse**'in geometrik temeli.

16.4 Sol Null Uzayı $N(A^T)$

$$N(A^T) = \{\mathbf{y} : A^T \mathbf{y} = \mathbf{0}\}$$

Neden "sol"? $A^T \mathbf{y} = \mathbf{0} \rightarrow$ transpoz al $\rightarrow \mathbf{y}^T A = \mathbf{0}^T$. \mathbf{y} satır vektörü olarak A 'yı **soldan** çarpıyor.

Sezgi: A 'nın satırlarının hangi kombinasyonu sıfır satır verir.

16.5 Satır Rank = Kolon Rank — Harika Gerçek

$$\dim C(A) = \dim C(A^T) = r$$

“The row space and the column space have the same dimension. That’s a wonderful fact.” — Strang, 15:06

Bariz değil! 1000 kolon, 3 satır → bağımsız kolon en fazla 3.

Pratik kestirme: Strang’ın Ders 9 örneğinde $(1, 1, 2)$, $(2, 2, 5)$, $(3, 3, 8)$ — iki satırı eşit ($r_1 = r_3$). Satır rank $\leq 2 \rightarrow$ kolon rank = 2 → üç kolon bağımlı.

16.6 Bazlar — Satır, Sol Null

Satır uzayı bazı: rref R ’nin sıfır-olmayan satırları (ilk r).

Örnek:

$$A = \begin{pmatrix} 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 1 \\ 1 & 2 & 3 & 1 \end{pmatrix} \rightarrow R = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Satır uzayı bazı: $(1, 0, 1, 1)$, $(0, 1, 1, 0)$.

Neden işe yarar? Satır işlemleri **satır uzayını korur** (kolon uzayını **değiştirir!**). A ’nın satırları R ’nin satırlarının kombinasyonu, ve tersi → iki satır uzayı özdeş.

Sol null uzayı bazı: Gauss-Jordan ile $[A \mid I] \rightarrow [R \mid E]$, yani $EA = R$.

$$\dim C(A) = 2$$

$$\dim C(A^T) = 2 \quad (\text{satır} = \text{kolon rank})$$

$$\dim N(A) = 2$$

$$\dim N(A^T) = 1$$

sol null uzayı bazı:

$$[-7.07106781e-01]$$

$$[-2.22044605e-16]$$

$$[7.07106781e-01]$$

R ’nin son satırı sıfırdı (bağımlı satır 3). $EA = R$ ’nin son $m - r$ satırı, A ’nın satırlarının sıfır veren kombinasyonunu söyler.

16.7 İki Çift — \mathbb{R}^n ve \mathbb{R}^m Bölünmesi


Girdi tarafı:

$$\mathbb{R}^n = C(A^T) \oplus N(A), \quad \text{boyutlar } r + (n - r) = n$$

Çıktı tarafı:

$$\mathbb{R}^m = C(A) \oplus N(A^T), \quad \text{boyutlar } r + (m - r) = m$$

Bu çiftler **birbirine dik** olacak (Ders 14).

 **Builder Notu — Least Squares ve SVD**

Least squares (Ders 16): $\mathbf{b} \in \mathbb{R}^m$ = kolon bileşeni (erişilebilir) + sol null bileşeni (artık). Least squares sol null'u atıp $C(A)$ üzerine izdüşümünü çözer.

SVD (Ders 29): Dört alt-uzaya **ortonormal bazlar** üretir; r tekil değer = ortak rank. Dört alt-uzay = dönüşümün tüm geometrisi; SVD = onun sayısal/ortonormal hâli.

16.8 Matris Uzayları

Şimdiye dek vektörler \mathbb{R}^n 'deydi. Ama **matrisler de vektör**: toplayabilir, sayıyla çarpabilir, kombinasyon alabilirsin.

3×3 matrisler uzayı M : 9 bağımsız giriş $\rightarrow \dim M = 9$.

Alt-uzaylar:

- Üst üçgensel: $3 + 3 = 6$ (köşegen + üst).
- Simetrik: $3 + 3 = 6$ (köşegen + üst, alt yansır).
- Diagonal: sadece köşegen = **3**.

Kesişim: simetrik \cap üst üçgensel = diagonal (Ders 6 kesişim kuralı).

Builder Notu: Bir ağın tüm ağırlıkları dev bir vektör uzayında nokta; yapısal kısıtlar (simetrik, düşük-rank, seyrek, blok) alt-uzaylar. **Tensörler** çok-boyutlu uzantı.

16.9 Bu Dersin Özeti

1. **Dört temel alt-uzay.**
2. **Satır** = $C(A^T)$, **sol null** = $N(A^T)$.
3. **Boyutlar:** $r, r, n - r, m - r$.
4. **Satır rank** = **kolon rank** = r .
5. **Satır bazı** = rref'in sıfır-olmayan satırları.
6. **Sol null bazı** = E 'nin son $m - r$ satırı ($[A \mid I] \rightarrow [R \mid E]$).

7. **İki çift:** $\mathbb{R}^n = C(A^T) + N(A)$, $\mathbb{R}^m = C(A) + N(A^T)$.
 8. **Matris uzayları** ve alt-uzayları.

! Tek bir cümle

Her matris dört temel alt-uzay doğurur; girdi \mathbb{R}^n ve çıktı \mathbb{R}^m ikiye parçaya bölünür. **Satır rank = kolon rank = r** — bir lineer dönüşümün tüm geometrisinin (SVD'den least squares'e) kodu.

16.10 Kontrol Soruları

i Soru 1: 4×6 matris, rank 3. Dört alt-uzayın boyutu ve uzayı?

$$m = 4, n = 6, r = 3.$$

- $C(A)$: \mathbb{R}^4 , boyut 3.
- $C(A^T)$: \mathbb{R}^6 , boyut 3.
- $N(A)$: \mathbb{R}^6 , boyut $6 - 3 = 3$.
- $N(A^T)$: \mathbb{R}^4 , boyut $4 - 3 = 1$.

$$\text{Kontrol: } \mathbb{R}^6 = 3 + 3\checkmark; \mathbb{R}^4 = 3 + 1\checkmark.$$

i Soru 2: $A = ((1,2,1),(2,4,3),(1,2,2))$ için satır uzayı bazı.

rref: $r_2 - 2r_1 = (0, 0, 1)$, $r_3 - r_1 = (0, 0, 1)$, $r_3 - r_2 = 0$. Pivot üstünü temizle:

$$R = \begin{pmatrix} 1 & 2 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

Satır uzayı bazı: $(1, 2, 0)$, $(0, 0, 1)$. $\dim = r = 2$.

i Soru 3: Satır operasyonları satır mı kolon uzayını mı korur?

Satır uzayını korur, kolon uzayını DEĞİL.

- Satır işlemi satırların kombinasyonu \rightarrow satır uzayında.
- Kolon uzayı değişir. Örnek: A 'da $(1, 1, 1)$ kolon uzayında olabilir, R 'de değil.

Rank ikisinde de aynı kalır (r), ama kolon uzayının **kendisi** değişir. Kolon uzayı bazı için A 'nın pivot kolonlarını alırız, R 'ninkileri değil.

i Soru 4: Dört alt-uzay \rightarrow least squares \rightarrow SVD bağı.

Least squares: $\mathbf{b} \in \mathbb{R}^m = C(A)$ bileşeni + $N(A^T)$ bileşeni (artık). LS, sol null'u atıp $C(A)$ üzerine izdüşümünü çözer.

SVD: Dört alt-uzaya ortonormal bazlar:

- Sağ tekil vektörler $V \rightarrow$ satır uzayı + null.
- Sol tekil vektörler $U \rightarrow$ kolon uzayı + sol null.
- r tekil değer = ortak rank.

Dört alt-uzay = dönüşüm geometrisi; SVD = onun sayısal hali. Düşük-rank yaklaşım, PCA, pseudoinverse hepsi buradan.

16.11 Egzersizler

Egzersiz 1. $A = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \end{pmatrix}$ — dört alt-uzayın boyutları.

Egzersiz 2. Aynı A için satır uzayı, null, sol null için baz.

Egzersiz 3. Kare ($n \times n$) tersinir matris — dört alt-uzay nedir? ($r = n$.)

Egzersiz 4. (Python) `null_space(A.T)` ile sol null'u çıkar.

Egzersiz 5. İspatla: $C(A^T) \cap N(A) = \{\mathbf{0}\}$. (İpucu: \mathbf{x} ikisinde de ise $\mathbf{x} = A^T \mathbf{z}$ ve $A\mathbf{x} = \mathbf{0} \rightarrow \mathbf{x}^T \mathbf{x} = ?$.)
Bu, Ders 14 **ortogonallik** habercisi.

16.12 Sonraki Ders İçin Hazırlık

Ders 11: Matris Uzayları, Rank 1 ve Küçük Dünya Grafikleri

- Matris alt-uzayları için tam sayım.
- **Rank 1 matrisler** (uv^T) — her matrisin yapı taşı; SVD ve LoRA çekirdeği.
- Küçük dünya grafikleri (Ders 12 geçişi).

⚠ Ders 11 öncesi

- Egzersiz 5 (ortogonallik habercisi) kritik.
- `null_space(A.T)` ile sol null'u birkaç matriste çıkar.

16.13 Anahtar Kavramlar (Cheat Sheet)

Kavram	Tanım	Strang'da
Dört alt-uzay	C, N, C^T, N^T	4m27
Satır uzayı	$C(A^T), \mathbb{R}^n$, boyut r	6m01
Sol null	$N(A^T), \mathbb{R}^m$, boyut $m - r$	7m15
Satır = kolon rank	$\dim C(A) =$ $\dim C(A^T) = r$	15m06

Kavram	Tanım	Strang'da
Satır bazı	rref'in sıfır-olmayan satırları	26m15
Sol null bazı	E 'nin son $m - r$ satırı	40m52
İki çift	$\mathbb{R}^n = C(A^T) + N(A)$; $\mathbb{R}^m = C(A) + N(A^T)$	42m15
Matris uzayı	$3 \times 3 \rightarrow \dim 9$; alt-uzaylar	43m31

16.14 ML Bağlantıları Özeti

💡 7 köprü

1. **Dört alt-uzay = SVD geometrisi.**
2. **Satır = kolon rank** \rightarrow SVD'de tek r ; ağırlık matrisinin etkin kapasitesi.
3. **Sol null = LS artıkları** \rightarrow Erişilemeyen çıktı yönleri.
4. **Satır + null = girdi ayrışımı** \rightarrow Etkin + yutulan; Ax sadece etkin parçaya.
5. **Rank-nullity iki uzayda** \rightarrow Parametre/boyut muhasebesi.
6. **Matris uzayları = parametre uzayı** \rightarrow Ağ ağırlıkları nokta; yapısal kısıtlar alt-uzaylar.
7. **Dört alt-uzay \rightarrow pseudoinverse A^+** \rightarrow Satır'ı kolon'a birebir eşler, null'ları yok sayar (Ders 33).

! Tek bir şey alıp gideceksen

Her matris dört temel alt-uzay üretir; girdi \mathbb{R}^n ve çıktı \mathbb{R}^m ikiye parçaya bölünür. **Satır = kolon rank = r** — bir lineer dönüşümün tüm geometrisini kodlar (SVD, LS, pseudoinverse).

17 Matris Uzayları, Rank 1 ve Küçük Dünya Grafikleri

Matrisler/fonksiyonlar/grafklar vektör; uv^T tüm matrislerin atomu

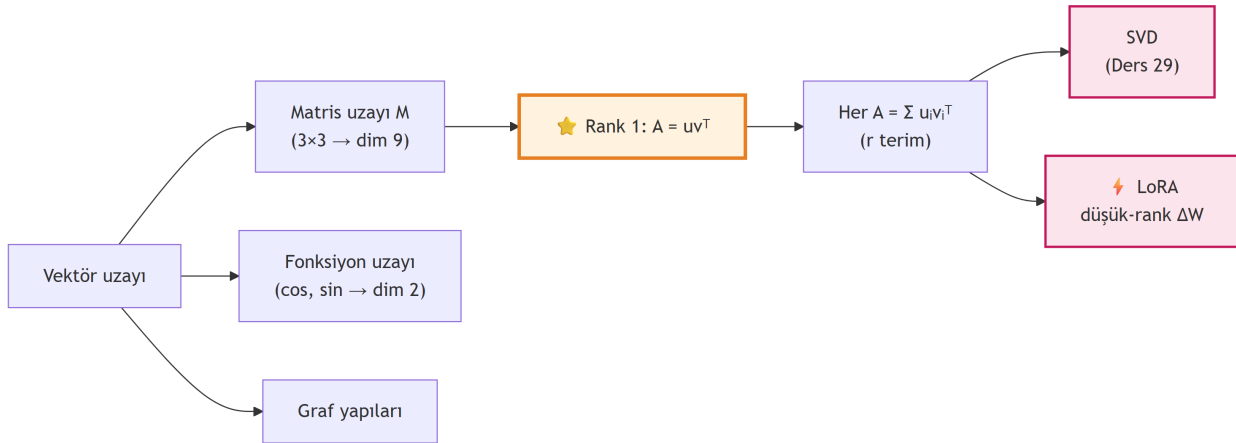
i Bölüm bilgisi

- **Strang'in videosu:** [YouTube — Lecture 11: Matrix Spaces, Rank 1, Small World Graphs](#) (≈ 50 dk)
- **OCW sayfası:** [MIT 18.06SC — Lecture 11](#)
- **Okuma süresi:** ≈ 40 dk

17.1 Bu Derste Ne Var?

“Rank one matrices are the building blocks for all matrices.” — Strang, 24:19

1. **Matris uzayları** — simetrik, üst üçgensel, diagonal; boyut formülü.
2. **Fonksiyon uzayları** — ODE çözümleri = vektör uzayı.
3. **Rank 1 matrisler** — $A = uv^T$, tüm matrislerin yapı taşı.
4. **Küçük dünya grafikleri** — altı derece ayırım, Ders 12'ye köprü.



Şekil 17.1: Vektör kavramı matrislerden fonksiyonlara/grafklara uzanır; rank-1 uv^T tüm matrislerin atomu — SVD/LoRA çekirdeği.

💡 Builder Notu — uv^T Her Yerde

- **Attention** QK^T dış-çarpım yapısı; düşük-rank yaklaşım hızlandırma (LInformer, Performer).
- **Embedding outer product’lar** — matris faktörizasyonu tabanlı öneri sistemleri.
- **LoRA**: $\Delta W = BA$ (birkaç rank-1) $\rightarrow n^2$ yerine $2nr$ parametre.
- **Fonksiyon uzayları** \rightarrow Fourier, wavelet, RKHS, Gaussian process.
- **Grafikler** \rightarrow GNN (adjacency), PageRank (özvektör), küçük dünya topoloji.

17.2 Matris Uzayı M

3×3 matrisler vektör uzayı (matris çarpımı umursanmaz, sadece toplama/skalerle çarpma):

$$\dim M = 9$$

Standart baz: 9 matris, her birinde tek bir 1.

“Our space is practically the same as nine-dimensional space.” — Strang, 4:40

17.3 Alt-Uzaylar — Simetrik, Üst Üçgensel, Diagonal

Alt-uzay	Serbest parametre	Boyut
Simetrik S	köşegen (3) + üst (3)	6
Üst üçgensel U	köşegen (3) + üst (3)	6
Diagonal D	sadece köşegen	3

17.4 Boyut Formülü — Kesişim ve Toplam


$S \cap U$: hem simetrik hem üst üçgensel \rightarrow alt = üst = 0 \rightarrow diagonal. $S \cap U = D$, dim 3.

$S + U$: tüm $s + u$ kombinasyonları \rightarrow tüm M , dim 9.

$$\dim S + \dim U = \dim(S \cap U) + \dim(S + U)$$

$$6 + 6 = 3 + 9 \checkmark$$

İçerme-dışlama’nın vektör uzayı versiyonu.

 **Builder Notu** — Yapısal Kısıtlar = Alt-Uzaylar

- **Simetrik:** kovaryans/kernel/attention.
- **Diagonal:** ölçekleme katmanları (LayerNorm gain, diagonal Fisher).
- **Üst üçgensel:** nedensel maskeler, Cholesky çarpanları.

Simetrik kısıt: n^2 yerine $n(n + 1)/2$ parametre.

17.5 Fonksiyon Uzayları — ODE Çözümleri

$$\frac{d^2y}{dx^2} + y = 0 \implies y = c_1 \cos x + c_2 \sin x$$

Çözümler vektör uzayı: $\text{span}\{\cos x, \sin x\}$, **dim 2** (ikinci mertebe denklem \rightarrow 2 boyut).

“Cosine and sine — they’re like the special solutions ... the dimension of the solution space will be two, because we have a second-order equation.” — Strang, 17:44

cos ve sin fonksiyonlar, ama toplanıp ölçeklendikleri için lineer cebirin tüm dili (baz, boyut, bağımsızlık) onlara da uygulanır.

Builder Notu: Fourier modları (sin/cos), wavelet’ler, **kernel methods**’un RKHS’i, Gaussian process’ler — hep sonsuz boyutlu fonksiyon vektör uzayları.

17.6 Rank 1 Matrisler — $A = \mathbf{u}\mathbf{v}^T$

$$A = \begin{pmatrix} 1 & 4 & 5 \\ 2 & 8 & 10 \end{pmatrix}$$

$r_2 = 2r_1 \rightarrow$ rank 1. Tüm satırlar $(1, 4, 5)$ ’in katı, tüm kolonlar $(1, 2)$ ’nin katı.

$$A = \begin{pmatrix} 1 \\ 2 \end{pmatrix} (1 \ 4 \ 5) = \mathbf{u}\mathbf{v}^T$$

Dış çarpım (outer product). Her rank-1 matris bu biçimde.

```
rank = 1
u v^T =
[[ 1  4  5]
 [ 2  8 10]]
tekil değerler = [1.44913767e+01 3.55271368e-15]
rank-1 yeniden inşa:
[[ 1.  4.  5.]
 [ 2.  8. 10.]]
```

17.7 Rank 1 = Yapı Taşı

Her rank r matris, r tane rank-1 matrisin toplamı:

$$A = \mathbf{u}_1 \mathbf{v}_1^T + \mathbf{u}_2 \mathbf{v}_2^T + \dots + \mathbf{u}_r \mathbf{v}_r^T$$

“Rank one matrices are the building blocks for all matrices.” — Strang, 24:19

Bu ayrışım = **SVD** (Ders 29):

$$A = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^T + \sigma_2 \mathbf{u}_2 \mathbf{v}_2^T + \dots$$

— terimleri tekil değer büyüklüğüne göre sıralar (Eckart–Young: en büyük k terim = en iyi rank- k yaklaşım).

💡 Builder Notu — LoRA'nın Matematiği

LoRA: Bir $n \times n$ ağırlık güncellemesi ΔW 'yi tam matris (n^2 parametre) yerine düşük-rank ($r \ll n$):

$$\Delta W = BA, \quad B \in \mathbb{R}^{n \times r}, A \in \mathbb{R}^{r \times n}$$

= r rank-1 dış-çarpımın toplamı, $2nr$ parametre. Fine-tuning değişimi pratikte düşük etkin rank'e sahip → çok az kayıpla. **Rank-1 = yapı taşı** sezgisi LoRA'nın doğrudan matematiği.

17.8 Sabit Rank Alt-Uzay Değil

“Tüm rank- k matrisler” alt-uzay mı? **Hayır.**

$$\text{rank}(A + B) \leq \text{rank}(A) + \text{rank}(B)$$

İki rank-1 toplandığında genelde **rank 2** olur. Toplamada kapalı değil → alt-uzay değil.

Builder Notu: Düşük-rank kısıtı **non-convex** → matris tamamlama, LoRA eğitimi zor problemler; **nükleer norm gevşetmesi** numarası gerekir.

17.9 Toplamı Sıfır Alt-Uzayı

$$S = \{\mathbf{v} \in \mathbb{R}^4 : v_1 + v_2 + v_3 + v_4 = 0\}$$

Bu $N(A)$, $A = (1, 1, 1, 1)$ için. A rank 1, $\dim N(A) = n - r = 4 - 1 = 3$.

Baz (özel çözümler):

$$(-1, 1, 0, 0)^T, (-1, 0, 1, 0)^T, (-1, 0, 0, 1)^T$$

Builder Notu: “Merkezleme” alt-uzayı. Bir veriyi ortalamadan çıkarmak onu bu alt-uzaya izdüşürür. Softmax gradyanı, simpleks kısıtları, contrast kodlamaları burada yaşar. Tek satırlık matris = bir kısıt bir boyut “yer”.

17.10 Dört Alt-Uzay — $A = (1, 1, 1, 1)$ Örneği

A (1×4 , rank 1):

- $C(A^T)$: $(1, 1, 1, 1)$ 'in katları, \mathbb{R}^4 'te, **boyut 1**.
- $N(A)$: toplamı sıfır olanlar, **boyut 3**. ($1 + 3 = 4 \checkmark$)
- $C(A)$: \mathbb{R}^1 'in tamamı, **boyut 1**.
- $N(A^T)$: sadece sıfır, **boyut 0**. ($1 + 0 = 1 \checkmark$)

“The basis of the smallest subspace is the empty set. The number of members in the empty set is zero — that’s the dimension.” — Strang, 38:49

17.11 Küçük Dünya Grafikleri

Graf = **düğüm**ler + **kenarlar** (kalkülüsteki “grafik” değil, ağ yapısı). Klasik örnek: insanlar düğüm, arkadaşlık kenar.

Altı derece ayırım: Rastgele iki insan ortalama ~6 arkadaşlık adımıyla bağlı.

“Six degrees of separation ... with a few shortcuts, the distances come down dramatically.” — Strang, 43:46

Bir graf bir **incidence matrisi** (m kenar \times n düğüm) ile temsil edilir. Lineer cebir bu matris üzerinde işler (Ders 12).

Builder Notu: GNN komşuluk matrisi üzerinde mesaj geçirir; **PageRank** web grafının özvektör problemi (Ders 21+); **küçük dünya / ölçeksiz** topoloji sosyal/nöral ağların temeli.

17.12 Bu Dersin Özeti

1. M (3×3) \rightarrow dim 9.
2. **Alt-uzaylar:** simetrik (6), üst üçgensel (6), diagonal (3).
3. **Boyut formülü:** $\dim S + \dim U = \dim(S \cap U) + \dim(S + U)$.
4. **Fonksiyon uzayları:** ODE çözümleri, $\text{span}\{\cos, \sin\} = 2$.
5. **Rank 1:** $A = uv^T$.
6. **Her matris** = r rank-1 toplamı (SVD özü).
7. **Sabit rank \neq alt-uzay** ($\text{rank}(A + B) \leq r_A + r_B$).
8. **Toplamı sıfır:** $N((1, 1, 1, 1))$, dim $n - 1$.
9. **Boş baz:** sıfır boyutlu uzayın bazı boş küme.
10. **Grafikler:** altı derece, incidence matrisi (Ders 12).

! Tek bir cümle

Vektör uzayı matrislerden fonksiyonlara/grafılara uzanır; **rank-1** (uv^T) tüm matrislerin atomudur ve her matris r rank-1 teriminin toplamıdır — **SVD, PCA ve LoRA**'nın ortak çekirdeği.

17.13 Kontrol Soruları

Soru 2: $A = ((2,6),(3,9),(1,3))$, rank 1 mi? uv^T ?

Soru 3: Tüm 3×3 rank-2 matrisler alt-uzay mı?

Soru 4: LoRA neden rank-1 (düşük-rank) yazar?

Tam SVD \rightarrow 2 parametre. Ama her $\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$ $\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$ rank 1 olabilir.

17.14 Egzersizler

Egzersiz 1. 3×3 antisimetrik ($A^T = -A$) matrislerin boyutu, baz. (İpucu: köşegen?)

Egzersiz 2. $A = \begin{pmatrix} 3 & 6 & 9 \\ 1 & 2 & 3 \\ 2 & 4 & 6 \end{pmatrix}$ rank? rank 1 ise uv^T .

Egzersiz 3. $S, U \subset \mathbb{R}^8$; $\dim S = 5$, $\dim U = 6$. $\dim(S \cap U)$ en az kaç? (Boyut formülü + $\dim(S + U) \leq 8$.)


Egzersiz 4. (Python) Rank-1 SVD ayrışımı.

Egzersiz 5. İspatla: $A = uv^T$ ($u, v \neq 0$) \rightarrow rank 1. (İpucu: her kolon u 'nun katı.) SVD'nin tek terimli hâli.

17.15 Sonraki Ders İçin Hazırlık

Ders 12: Graflar, Ağlar ve Incidence Matrisleri

- **Incidence matrisi** (m kenar $\times n$ düğüm).
- Dört alt-uzay grafta: null (potansiyeller), sol null (çevrimler).
- **Kirchhoff yasaları**, Euler formülü.

 Ders 12 öncesi

- Egzersiz 5 (rank-1 \rightarrow SVD habercisi) kritik.
- `np.outer` ve `np.linalg.svd` ile rank-1 dene.

17.16 Anahtar Kavramlar (Cheat Sheet)

Kavram	Tanım	Strang'da
Matris uzayı M	$3 \times 3 \rightarrow \dim 9$	3m59
Simetrik / üst üçgensel	Her biri $\dim 6$	6m13
Diagonal	$\dim 3 (S \cap U = D)$	8m36
Boyut formülü	$\dim S + \dim U =$ $\dim(S \cap U) + \dim(S + U)$	13m48
Fonksiyon uzayı	ODE çözümleri, $\dim 2$	14m43
Rank 1 = uv^T	Dış çarpım	23m06
Yapı taşı	Her rank- r matris = r rank-1 toplamı	24m19
rank toplamı	$\text{rank}(A + B) \leq r_A + r_B$	27m35
Toplamı sıfır	$N((1, 1, 1, 1))$, $\dim n - 1$	29m21
Altı derece	Küçük dünya, graf	43m46

17.17 ML Bağlantıları Özeti

💡 7 köprü

1. **Rank 1 = dış çarpım** → Attention QK^T , embedding outer products, matris faktörizasyonu.
2. **Rank-1 toplamı = SVD** → Görüntü sıkıştırma, gürültü giderme.
3. **LoRA = düşük-rank güncelleme** → Fine-tuning standart yöntemi.
4. **Fonksiyon uzayları** → Fourier, wavelet, RKHS, GP.
5. **Boyut formülü** → İçerme-dışlama, özellik alt-uzayları muhasebesi.
6. **Sabit rank non-convex** → Düşük-rank optim zorluğu, nükleer norm.
7. **Grafikler + matrisler** → GNN, PageRank, küçük dünya topoloji.

! Tek bir şey alıp gideceksen

Vektör uzayı matrislerden fonksiyonlara/grafllara genelleşir; **rank-1** (uv^T) tüm matrislerin atomu — her matris r rank-1 toplamı (SVD, PCA, LoRA çekirdeği).

18 Graflar, Ağlar ve Incidence Matrisleri

Dört alt-uzay fizikselleşir — potansiyel, akım, ağaç, çevrim

Bölüm bilgisi

- **Strang'ın videosu:** [YouTube — Lecture 12: Graphs, Networks, Incidence Matrices](#) (≈48 dk)
- **OCW sayfası:** [MIT 18.06SC — Lecture 12](#)
- **Okuma süresi:** ≈40 dk

18.1 Bu Derste Ne Var?

Bu ders **uygulama** dersi: matris gerçek bir yapıdan (graf) doğuyor, dört temel alt-uzay fiziksel anlam kazanıyor.

1. **Incidence matrisi:** graf \rightarrow matris.
2. **Dört alt-uzayın fiziksel anlamı:** null = potansiyel, sol null = Kirchhoff Akım Yasası, satır = ağaç.
3. **Euler formülü:** düğüm – kenar + çevrim = 1.
4. **$A^T CA$ çerçevesi:** uygulamalı matematiğin temel denklemi.

“A transpose y equals zero is probably the most fundamental equation of applied mathematics.”
— Strang, 20:07

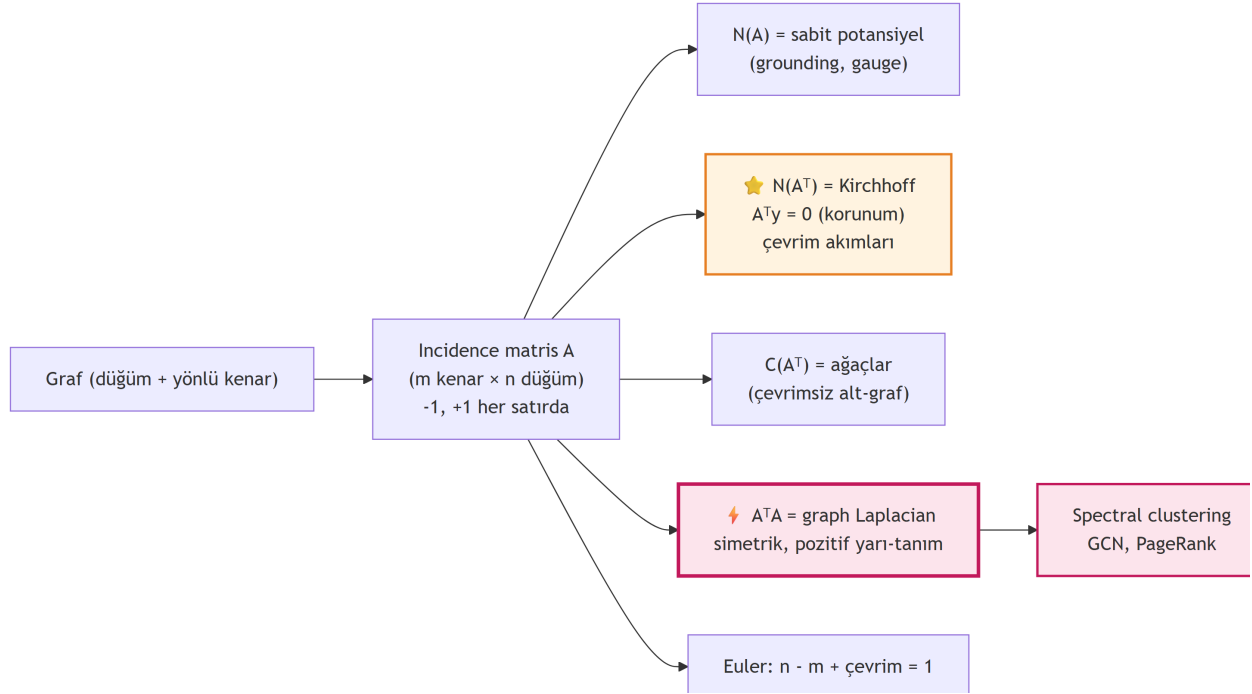
Builder Notu — Graf Matrisleri ML’de

- **Incidence / adjacency** \rightarrow GNN’lerin girdi yapısı.
- **Graph Laplacian** $L = A^T A \rightarrow$ spectral clustering (Fiedler), GCN konvolüsyon, label propagation.
- **Null = gauge özgürlüğü** $\rightarrow L$ ’nin sıfır özdeğeri = bağlı bileşen sayısı.
- **Sol null = çevrimler** \rightarrow grafın homolojisi; topolojik veri analizi (TDA).

18.2 Graf ve Incidence Matrisi

Strang’ın örneği: 4 düğüm, 5 kenar.

- Kenar 1: $1 \rightarrow 2$
- Kenar 2: $2 \rightarrow 3$



Şekil 18.1: Graf \rightarrow incidence $A \rightarrow$ dört alt-uzay fizikselleşir; $A^T A =$ graph Laplacian.

- Kenar 3: $1 \rightarrow 3$
- Kenar 4: $1 \rightarrow 4$
- Kenar 5: $3 \rightarrow 4$

Her kenar bir satır. Kenar $i \rightarrow j$ için satırda i kolonunda -1 , j kolonunda $+1$:

$$A = \begin{pmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ -1 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 1 \end{pmatrix}$$

Seyrek (her satırda 2 sıfır-olmayan). İlk 3 kenar bir çevrim \rightarrow satır 1 + satır 2 = satır 3 \rightarrow **bağımlı**.

“Loops correspond to linearly dependent rows.” — Strang, 8:54

18.3 Null Uzayı — Potansiyeller ve Grounding

\mathbf{x} = düğüm potansiyelleri. $A\mathbf{x}$ = kenar potansiyel farkları:

$$A\mathbf{x} = (x_2 - x_1, x_3 - x_2, x_3 - x_1, x_4 - x_1, x_4 - x_3)^T$$

A = fark operatörü (graf gradyanı).

$A\mathbf{x} = \mathbf{0} \iff$ tüm farklar sıfır \iff tüm potansiyeller eşit:

$$N(A) = \{c(1, 1, 1, 1)^T\}, \quad \dim N(A) = 1$$

$$r = n - 1 = 3$$

Grounding: Bir düğümü 0 yaparak referansı sabitle. Bu **gauge özgürlüğü** (L 'nin sıfır özdeğeri = bağlı bileşen sayısı).

18.4 Sol Null Uzayı — Kirchhoff Akım Yasası

\mathbf{y} = kenar akımları. $A^T \mathbf{y} = \mathbf{0}$ = **Kirchhoff Akım Yasası (KCL)**.

A^T 'nin her satırı bir düğüm \rightarrow “giren akım = çıkan akım”.

Düğüm 1: $-y_1 - y_3 - y_4 = 0$ (üç çıkış toplamı).

“*Kirchhoff's Current Law is a balance equation, a conservation law. In equals out.*” — Strang, 28:43

Boyut: $\dim N(A^T) = m - r = 5 - 3 = 2$.

18.5 Loop Akımları — Bazlar

Bir çevrim etrafında dolaşan akım her zaman KCL'i sağlar. İki bağımsız çevrim:

$$\mathbf{y}_1 = (1, 1, -1, 0, 0)^T, \quad \mathbf{y}_2 = (0, 0, 1, -1, 1)^T$$

Büyük çevrim = $\mathbf{y}_1 + \mathbf{y}_2$ (kenar 3'te akımlar birbirini götürür) \rightarrow bağımsız değil.

$\dim N(A^T)$ = **bağımsız çevrim sayısı**.

18.6 Ağaç (Tree)

Çevrimsiz graf. Bağımsız kenar = ağacın kenarı. Bir grafın bağımsız kenar sayısı = $n - 1$ (ağacın kenar sayısı).

18.7 Euler Formülü

Boyut sayımını birleştir:

$$\text{çevrim} = m - r = m - (n - 1) = m - n + 1$$

$$n - m + \text{çevrim} = 1$$

Örnekte: $4 - 5 + 2 = 1$ ✓. **Lineer cebir bir topoloji teoremini kanıtladı.**

“The number of nodes minus the number of edges plus the number of loops is one. It’s known as Euler’s Formula.” — Strang, 41:52

Builder Notu: Euler formülü **topolojik veri analizi (TDA)** ve **homolojinin** kalbidir. Persistent homology bir veri bulutunun çoklu ölçekte “deliklerini” sayar — graf düzeyinde çevrim sayısı = ağın redundancy/dayanıklılık ölçüsü.

18.8 $A^T C A$ Çerçevesi — Uygulamalı Matematiğin Temeli

Üç fiziksel adım:

1. **Potansiyel farkları:** $\mathbf{e} = A\mathbf{x}$ (gradyan).
2. **Ohm Yasası:** $\mathbf{y} = C\mathbf{e}$ (iletkenlik).
3. **KCL:** $A^T \mathbf{y} = \mathbf{f}$ (korunum, dış kaynak \mathbf{f}).

Birleştir:

$$A^T C A \mathbf{x} = \mathbf{f}$$

“There’s the basic equation of applied math.” — Strang, 46:22

Her yerde: elektrik devreleri, yapısal mekanik, ısı akışı, akışkanlar. Denge (equilibrium) problemlerinin evrensel iskeleti.

18.9 Graph Laplacian = $A^T A$

$C = I$ özel hâli: $A^T A$. **Her zaman simetrik, pozitif yarı-tanımlı.**

L (graph Laplacian) =

```
[[ 3. -1. -1. -1.]
 [-1.  2. -1.  0.]
 [-1. -1.  3. -1.]
 [-1.  0. -1.  2.]]
```

Simetrik? True

Özdeğerler = [0. 2. 4. 4.]

rank(A) = 3

L 'nin köşegen = düğüm derecesi; köşegen-dışı = bağlı düğümlere -1 . **Sabit vektör L 'nin null uzayında** \rightarrow sıfır özdeğer.

💡 Builder Notu — Graph Laplacian ML'de

- **Spectral clustering:** En küçük (Fiedler) özvektör grafi zayıf-bağlı yerden böler.
- **GCN:** Normalize L konvolüsyon çekirdeği.
- **Diffusion / heat kernel:** PageRank, label propagation, GraphSAGE.
- **Sıfır özdeğer sayısı = bağlı bileşen sayısı.**
- L 'nin spektrumu grafin geometrisini kodlar.

18.10 Bu Dersin Özeti

1. **Graf** \rightarrow düğüm + yönlü kenar.
2. **Incidence matrisi:** kenar \times düğüm.
3. **Null = sabit potansiyel** (gauge); rank $n - 1$.
4. A = **fark operatörü**; grounding.
5. **Ağaç** = çevrimsiz; bağımsız kenarlar.
6. **Sol null = Kirchhoff.**
7. **Loop akımları** = sol null bazıı.
8. **Euler:** $n - m + \text{çevrim} = 1$.
9. $A^T C A$ denge denklemi.
10. $A^T A = \text{graph Laplacian}$, spectral yöntemler.

! Tek bir cümle

Bir graf bir incidence matrisine dönüşür; dört alt-uzay fizikselleşir — null = potansiyel, sol null = Kirchhoff korunumu (çevrimler), satır = ağaçlar. $A^T C A$ denge denklemi, $A^T A$ graph Laplacian olarak graf öğrenmesinin merkezi.

18.11 Kontrol Soruları

i Soru 1: 3 düğüm üçgen graf ($1 \rightarrow 2, 2 \rightarrow 3, 1 \rightarrow 3$) — incidence matris, rank.

$$A = \begin{pmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \\ -1 & 0 & 1 \end{pmatrix}$$

$r_1 + r_2 = r_3 \rightarrow$ bağımlı. **rank** = $n - 1 = 2$.

i Soru 2: Üçgen grafın null uzayı ve anlamı.

$$N(A) = \{c(1, 1, 1)^T\}, \dim 1.$$

Fiziksel: potansiyeller sabite kadar belirli; bir düğümü topraklayarak (potansiyel = 0) belirsizlik giderilir.

i Soru 3: 6 düğüm, 9 kenar bağlı graf — kaç bağımsız çevrim?

Euler: $6 - 9 + \text{çevrim} = 1 \rightarrow$ **çevrim = 4**.

Alternatif: $r = n - 1 = 5$; $\text{çevrim} = m - r = 4$. ✓

i Soru 4: Graph Laplacian GNN ve spectral clustering’de nasıl?

Sıfır özdeğer: $L(1, \dots, 1) = \mathbf{0} \rightarrow L$ ’nin sıfır özdeğeri. **Sayısı = bağlı bileşen sayısı**.

Spectral clustering: En küçük sıfırdan sonraki özvektör (Fiedler) grafı zayıf bağlı yerden böler.

GCN: Normalize L konvolüsyon çekirdeği; her katman komşu özelliklerini Laplacian ağırlıklarıyla toplar.

L ’nin spektrumu grafın tüm spektral/difüzyon davranışını kodlar.

18.12 Egzersizler

Egzersiz 1. 4 düğüm yol grafı ($1 \rightarrow 2, 2 \rightarrow 3, 3 \rightarrow 4$) için incidence yaz. Ağaç mı? Rank, çevrim sayısı.

Egzersiz 2. $A = \begin{pmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}$ — null ve sol null boyutu. (3. ve 4. satır aynı \rightarrow ne anlama gelir?)

Egzersiz 3. Bağlı graf, 10 düğüm, 15 kenar — rank, null boyut, çevrim, Euler doğrulaması.

Egzersiz 4. (Python) Graph Laplacian özdeğerleri.

Egzersiz 5. İspatla: Bağlı grafta $\dim N(A) = 1$, sabit vektör tarafından spanlanır. (İpucu: $Ax = \mathbf{0}$ komşu düğümlerin eşit potansiyelini gerektirir; bağlılık tüm grafa yayar.) Bu, Laplacian’ın tam bir sıfır özdeğerine sahip olmasını açıklar.

18.13 Sonraki Ders İçin Hazırlık

Ders 13: Quiz 1 İncelemesi

Chapter 1-3'ün toplu tekrarı: eliminasyon, LU, vektör uzayları, dört alt-uzay, rank, $Ax = b$, bağımsızlık/baz/boyut.

⚠ Ders 13 öncesi

- Ders 1-12 egzersizlerini gözden geçirin.
- A.T @ A ile Laplacian'ı inceleyin.

18.14 Anahtar Kavramlar (Cheat Sheet)

Kavram	Tanım	Strang'da
Graf	Düğüm + yönlü kenar	2m38
Incidence matrix	Kenar \times düğüm, $-1, +1$	5m49
Loop = bağımlı satır	Çevrimler bağımlılık	8m01
A = fark operatörü	$Ax =$ potansiyel farkları	13m19
Null = sabit potansiyel	dim 1, rank $n - 1$	15m22
Grounding	Referansı sabitle	18m06
KCL: $A^T y = 0$	Korunum	25m50
Loop akımları	Sol null bazı	33m19
Ağaç	Çevrimsiz	39m06
Euler	$n - m + \text{çevrim} = 1$	41m52
$A^T C A$	Uygulamalı mat. denge	46m22

18.15 ML Bağlantıları Özeti

💡 7 köprü

1. **Incidence / adjacency** \rightarrow GNN girdi yapısı.
2. **Graph Laplacian ($A^T A$)** \rightarrow Spectral clustering, GCN, label propagation.
3. **Null = gauge** \rightarrow Laplacian sıfır özdeğer = bağlı bileşen sayısı.
4. **Sol null = çevrimler** \rightarrow Homoloji, TDA, persistent homology.
5. **$A^T C A$ denge denklemi** \rightarrow Physics-informed models, FEM.
6. **Korunum** \rightarrow Akış, optimal transport, GNN aggregation.
7. **Euler / topoloji** \rightarrow Veri bulutunun deliklerini sayma; TDA.

! Tek bir şey alıp gideceksen

Graf \rightarrow incidence matris \rightarrow dört alt-uzay fizikselleşir (potansiyel, akım, ağaç, çevrim). $A^T C A$ denge denklemleri her yerde; $A^T A = \mathbf{graph Laplacian}$ graf öğrenmesinin merkezi.

19 Quiz 1 İncelemesi

Ders 1-12 toplu tekrar — rank, $x_p + x_n$, alt-uzay testleri

i Bölüm bilgisi

- **Strang'ın videosu:** [YouTube — Quiz 1 Review](#) (≈ 47 dk)
- **OCW sayfası:** [MIT 18.06SC — Exam 1 Review](#)
- **Okuma süresi:** ≈ 35 dk

19.1 Bu Derste Ne Var?

Tekrar oturumu. Yeni kavram yok; eski quiz sorularıyla Chapter 1-3 birleştirilir.

1. **Boyut sayma:** span, rank, dört alt-uzay.
2. **Ters mühendislik:** tam çözüm \rightarrow matrisi geri çıkar.
3. **Doğru/yanlış mantığı:** alt-uzay testleri, $B^2 = 0$ tuzakları.
4. **Önizleme:** satır uzayı \perp null uzayı (Ders 14).

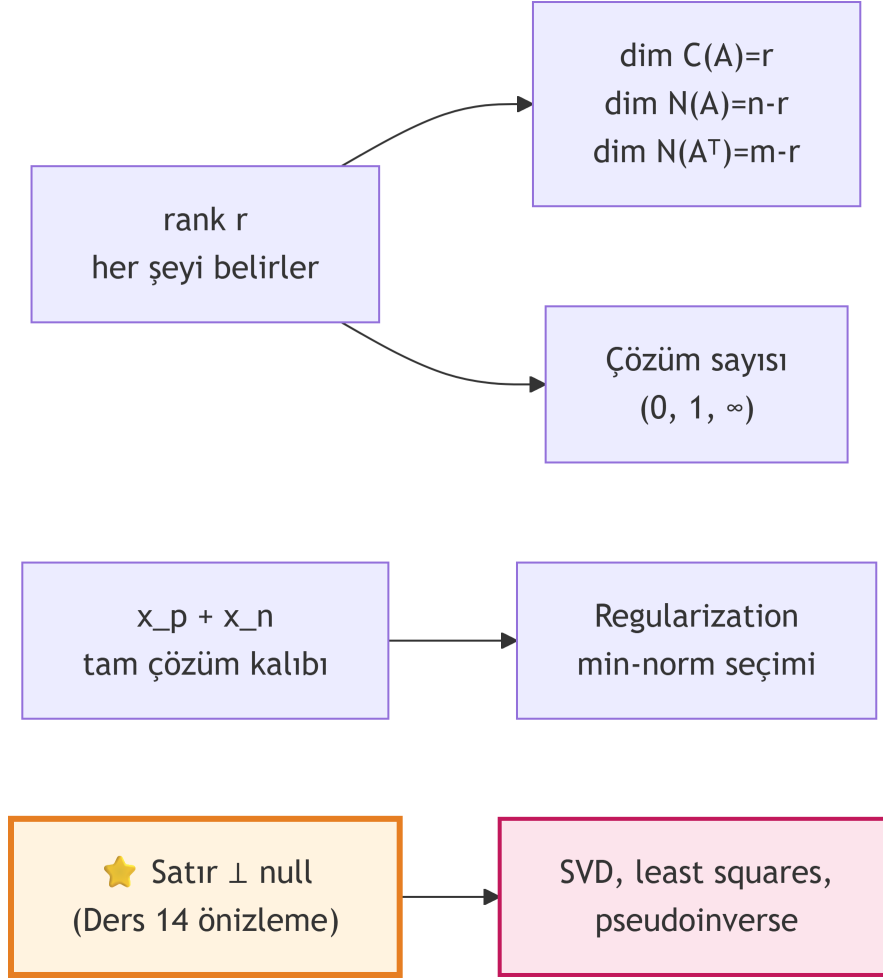
“This particular plus null space pattern goes throughout mathematics of linear systems... it spreads everywhere.” — Strang, 36:31

19.2 Span ve Boyut

Soru: u, v, w sıfır-olmayan \mathbb{R}^7 'de. Span boyutu?

Cevap: 1, 2 veya 3. En fazla 3 (vektör sayısı), en az 1 (hepsi sıfır-olmayan).

Genel: k vektörün span'ı en fazla k boyut = bağımsız sayısı (rank).



Şekil 19.1: Quiz 1 ana temaları: rank her şeyi belirler; $x_p + x_n$ her yerde.

19.3 Blok Matrisler ve Rank

U (5×3 , rank 3 = full column).

(a) $N(U) = \{\mathbf{0}\}$ (full column rank).

(b) $\begin{pmatrix} U \\ 2U \end{pmatrix}$ (10×3): $2U$ birinin katı \rightarrow eliminasyonda sıfırlanır \rightarrow rank **3**.

(c) $\begin{pmatrix} U & U \\ U & 0 \end{pmatrix}$ (10×6): blok eliminasyonu \rightarrow 2 bağımsız $U \rightarrow$ rank **6**.

“That has rank six, I can tell.” — Strang, 8:56

Builder Notu: Derin öğrenmede blok-yapılı ağırlık matrisleri (multi-head attention, blok-diagonal katmanlar). “Tekrarlı blok rank katmaz” sezgisi weight tying’in etkin rank’i nasıl sınırladığını verir.

19.4 Dört Alt-Uzayın Boyutları

C 10×6 , rank 6 $\rightarrow \dim N(C^T) = m - r = 10 - 6 = 4$.

Formüller ($m \times n$, rank r):

- $\dim C(A) = r$
- $\dim C(A^T) = r$
- $\dim N(A) = n - r$
- $\dim N(A^T) = m - r$

19.5 Ters Mühendislik — Çözümünden Matrise

$A\mathbf{x} = (2, 4, 2)^T$ tam çözüm: $\mathbf{x}_p = (2, 0, 0) + c(1, 1, 0) + d(0, 0, 1)$. A nedir?

- Null uzayında 2 vektör $\rightarrow \dim N = 2$, $n = 3 \rightarrow$ **rank 1**.
- $(0, 0, 1) \in N(A) \rightarrow$ 3. kolon = $\mathbf{0}$.
- $(1, 1, 0) \in N(A) \rightarrow$ kolon₁ + kolon₂ = $\mathbf{0} \rightarrow$ kolon₂ = $-\text{kolon}_1$.
- $\mathbf{x}_p = (2, 0, 0) \rightarrow 2 \cdot \text{kolon}_1 = (2, 4, 2) \rightarrow \text{kolon}_1 = (1, 2, 1)$.

$$A = \begin{pmatrix} 1 & -1 & 0 \\ 2 & -2 & 0 \\ 1 & -1 & 0 \end{pmatrix}$$

Builder Notu: Bu “çözümünden modeli geri çıkarma” = **kimliklenebilirlik (identifiability)** sorusu. Null uzayı büyükse model tam belirlenemez (birden çok parametre aynı çıktıyı).

19.6 Çözülebilirlik

$Ax = \mathbf{b}$ ancak $\mathbf{b} \in C(A) = \{c(1, 2, 1)^T\}$.

19.7 Doğru/Yanlış

(a) **Kare** A , $N(A) = \{\mathbf{0}\} \rightarrow N(A^T) = ? \rightarrow$ Kare matriste $N(A) = \{\mathbf{0}\} \iff$ tersinir $\iff N(A^T) = \{\mathbf{0}\}$.

(b) **5×5 tersinir matrisler alt-uzay mı?** \rightarrow **HAYIR**. Sıfır matris tersinir değil.

(c) **5×5 singular matrisler alt-uzay mı?** \rightarrow **HAYIR**. İki singular toplam tersinir olabilir:

$$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} = I$$

Builder Notu: Tersinir matrisler = manifold (GL grubu); alt-uzay değil. Normalizing flows, RNN stabilizasyonu **manifold optimizasyonu** gerektirir.

19.8 $B^2 = 0$ ve Çarpım Null Uzayı

(a) $B^2 = 0 \rightarrow B = 0$? **YANLIŞ**. Nilpotent örnek:

$$B = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \quad B^2 = 0$$

(b) C tersinir $\rightarrow N(CD) = N(D)$. $CD\mathbf{x} = \mathbf{0} \iff D\mathbf{x} = \mathbf{0}$.

Builder Notu: Soldan tersinir çarpım = **preconditioning**, whitening, baz değişimi — çözümü değiştirmez ama optimizasyonu iyileştirir.

19.9 Satır vs Kolon Uzayı

(a) **Kare için satır = kolon uzayı?** **YANLIŞ**. $A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$ — satır $(0, 1)$, kolon $(1, 0)$ katları. Boyutları aynı (r), uzaylar farklı. **Simetrik** ise eşit.

(b) A ve $-A$ aynı dört alt-uzay? **EVET**.

(c) **Aynı dört alt-uzay** $\rightarrow A = cB$? **YANLIŞ**. İki tersinir 6×6 aynı alt-uzaylara sahip ($C = \mathbb{R}^6$, $N = \{\mathbf{0}\}$) ama farklı matris.

19.10 Satır Operasyonları

Satır takası neyi korur?

- **Satır uzayı** ✓ (kümesi aynı)
- **Null uzayı** ✓ ($Ax = \mathbf{0}$ aynı)
- **Kolon uzayı** × (değişir)

“The row space does stay the same. And the null space stays the same. Column space would be a wrong answer.” — Strang, 43:53

Bu yüzden kolon uzayı bazı için **orijinal** A 'nın pivot kolonlarını alırız.

19.11 Satır Uzayı \perp Null Uzayı — Ders 14 Önizleme

Soru: $(1, 2, 3)$ neden hem satır hem null uzayında olamaz?

$$(1, 2, 3) \cdot (1, 2, 3) = 14 \neq 0$$

Ama null uzayında olsaydı bu çarpım sıfır olurdu. **Çelişki** → bir vektör hem satır hem null'da olamaz (sıfır hariç).

“The null space is perpendicular to the row space.” — Strang, 47:14

Chapter 4 **ortogonallik** burada başlıyor.

19.12 Bu Tekrarın Özeti

1. **Span boyutu** — bağımsız sayısı.
2. **Blok rank** — tekrar katmaz.
3. **Dört boyut formülü.**
4. **Ters mühendislik** — çözüm → kolonlar.
5. **Çözülebilirlik** = $\mathbf{b} \in C(A)$.
6. **Alt-uzay testi** — sıfır + kapalılık.
7. **Tuzaklar** — $B^2 = 0 \not\Rightarrow B = 0$; $N(CD) = N(D)$ tersinir C .
8. **Satır \neq kolon uzayı** (genel).
9. **Satır takası** satır + null korur.
10. **Satır \perp null** (Ders 14).

! Tek bir cümle

Rank her şeyi belirler (boyut, çözüm sayısı); **tam çözüm** = $\mathbf{x}_p + \mathbf{x}_n$; ve satır uzayı \perp null uzayı — Chapter 4'ün kapısı.

19 Quiz 1 İncelemesi

i Soru 1: 7×4 , rank 3 — dört boyut, $Ax = b$ durumu.

19.13 Kontrol Soruları

$$m = 7, n = 4, r = 3.$$

- $\dim C(A) = 3$ (\mathbb{R}^7 'de)
- $\dim C(A^T) = 3$ (\mathbb{R}^4 'te)
- $\dim N(A) = 1$
- $\dim N(A^T) = 4$

$r < m \rightarrow$ bazı \mathbf{b} için 0 çözüm; $r < n \rightarrow$ çözüm varsa ∞ . **0 veya ∞ .**

i Soru 2: $Ax = (1,2)$ tam çözüm $(3,0,0) + c(1,-1,0) + d(0,0,1) \rightarrow A$?

- $(0, 0, 1) \in N \rightarrow 3. \text{ kolon} = \mathbf{0}$.
- $(1, -1, 0) \in N \rightarrow \text{kolon}_1 = \text{kolon}_2$.
- $\mathbf{x}_p = (3, 0, 0) \rightarrow 3 \cdot \text{kolon}_1 = (1, 2) \rightarrow \text{kolon}_1 = (1/3, 2/3)$.

$$A = \begin{pmatrix} 1/3 & 1/3 & 0 \\ 2/3 & 2/3 & 0 \end{pmatrix}$$

rank 1, $\dim N = 2 \checkmark$.

i Soru 3: 3×3 iz=0 matrisler alt-uzay mı?

Doğru. İz lineer: $\text{tr}(A + B) = \text{tr}(A) + \text{tr}(B)$, $\text{tr}(cA) = c \cdot \text{tr}(A)$.

- Sıfır matris izi 0 \rightarrow içerir.
- Toplamada/çarpmada kapalı.

Boyut: $9 - 1 = 8$ (1 kısıt = 1 boyut yer). Bu, $\mathfrak{sl}(n)$ Lie cebiri uzayı.

i Soru 4: P tersinir $\rightarrow PA$ hangi alt-uzayları korur? ML karşılığı?

- **Null:** $PA\mathbf{x} = \mathbf{0} \iff A\mathbf{x} = \mathbf{0} \rightarrow$ korunur.
- **Satır uzayı:** P 'nin satırları A 'nın satırlarının komb. \rightarrow korunur.
- **Kolon uzayı:** değişir.

ML:

- **Preconditioning:** $PA\mathbf{x} = P\mathbf{b} \rightarrow$ çözüm aynı, yakınsama hızlı.
- **Whitening:** veriyi tersinirle ölçüle \rightarrow lineer yapı korunur, optimizasyon iyi.
- **Baz değişimi (Ders 31):** problemin özü değişmez.

19.14 Egzersizler

Egzersiz 1. 4×7 , rank 4 — dört boyut, $A\mathbf{x} = \mathbf{b}$ çözüm durumu.

Egzersiz 2. $A\mathbf{x} = (4, 8)^T$ tam çözüm $(2, 0, 0) + c(1, -2, 0) + d(0, 1, 1) \rightarrow A (2 \times 3)$.

Egzersiz 3. Doğru/Yanlış:

- (a) $N(A) = \{\mathbf{0}\} \rightarrow$ kolonlar bağımsız.
- (b) 3×3 simetrik matrisler alt-uzay mı?
- (c) $A^2 = I \rightarrow A = I$ veya $A = -I$.
- (d) Satır takası kolon uzayını korur.

Egzersiz 4. (Python) Boyut sayımı ve çözülebilirlik testleri.

Egzersiz 5. İspatla: Bir vektör hem satır hem null uzayındaysa sıfırdır. (İpucu: $\mathbf{v} = A^T \mathbf{z}$ ve $A\mathbf{v} = \mathbf{0} \rightarrow \mathbf{v}^T \mathbf{v} = 0$.) **Ders 14'ün tam ifadesi.**

19.15 Sonraki Ders İçin Hazırlık

Ders 14: Ortogonal Vektörler ve Alt-Uzaylar

- $\mathbf{x}^T \mathbf{y} = 0$, Pisagor.
- Ortogonal alt-uzaylar.
- Satır \perp null, kolon \perp sol null.
- $A^T A$ matrisi.

Ders 14 öncesi

- Ders 1-12 egzersizlerini gözden geçir.
- `matrix_rank` ile boyut sayımı ve çözülebilirlik pratik.
- Ana cümleyi tekrar oku.

19.16 Sınav Formülleri (Cheat Sheet)

Soru tipi	Anahtar	Strang'da
Span boyutu	En fazla k ; bağımsız sayısı	1m47
Full column null	$r = n \rightarrow N(A) = \{\mathbf{0}\}$	3m03
Blok rank	Tekrar katmaz	4m31
Dört boyut	$r, r, n - r, m - r$	9m08
Çözümünden matrise	$\mathbf{x}_p \rightarrow$ kolon; null \rightarrow ilişki	11m05
Çözülebilirlik	$\mathbf{b} \in C(A)$	16m44
Kare $N=\{\mathbf{0}\}$	\Leftrightarrow tersinir	18m46
Alt-uzay testi	Sıfır + kapalılık	19m20
$N(CD) = N(D)$	C tersinir	30m02

Soru tipi	Anahtar	Strang'da
Satır takası	Satır + null korur	43m29
Satır \perp null	Sadece $\mathbf{0}$ 'da kesişir	47m14

19.17 ML Bağlantıları Özeti

💡 7 köprü

1. **Boyut sayma refleksi** → serbestlik dereceleri teşhisi.
2. $\mathbf{x}_p + \mathbf{x}_n$ → regularization, min-norm.
3. **Tersinir sol çarpım null korur** → preconditioning, whitening.
4. **Satır \perp null** → SVD, LS, pseudoinverse zemini.
5. **Ters mühendislik = identifiability** → null büyükse model belirsiz.
6. **Tersinir = manifold** → Riemannian/manifold optimizasyon.
7. **Blok rank** → weight tying / parametre paylaşımı.

❗ Tek bir şey alıp gideceksen

Rank tek başına sistemin kaderini belirler; tam çözüm $\mathbf{x}_p + \mathbf{x}_n$; satır uzayı \perp null uzayı — Chapter 4 (ortogonallik) ve SVD'nin başlangıcı.

20 Ortogonal Vektörler ve Alt-Uzaylar

90 derece bölümü — temel teorem 2. kısım

Bölüm bilgisi

- **Strang'in videosu:** [YouTube — Lecture 14: Orthogonal Vectors and Subspaces](#) (≈48 dk)
- **OCW sayfası:** [MIT 18.06SC — Lecture 14](#)
- **Okuma süresi:** ≈40 dk

20.1 Bu Derste Ne Var?

Yeni bölüm: **ortogonallik**. Ders 13'ün sonundaki “sıtr uzayı \perp null uzayı” sezgisi burada tam teori.

1. $\mathbf{x}^T \mathbf{y} = 0$ — Pisagor.
2. Ortogonal alt-uzaylar.
3. **Temel teorem 2. kısım:** sıtr \perp null, kolon \perp sol null — **ortogonal tümleyen**.
4. $A^T A$ — çözümsüz $A\mathbf{x} = \mathbf{b}$ için least squares hazırlık.

“Row space is orthogonal to the null space.” — Strang, 20:49

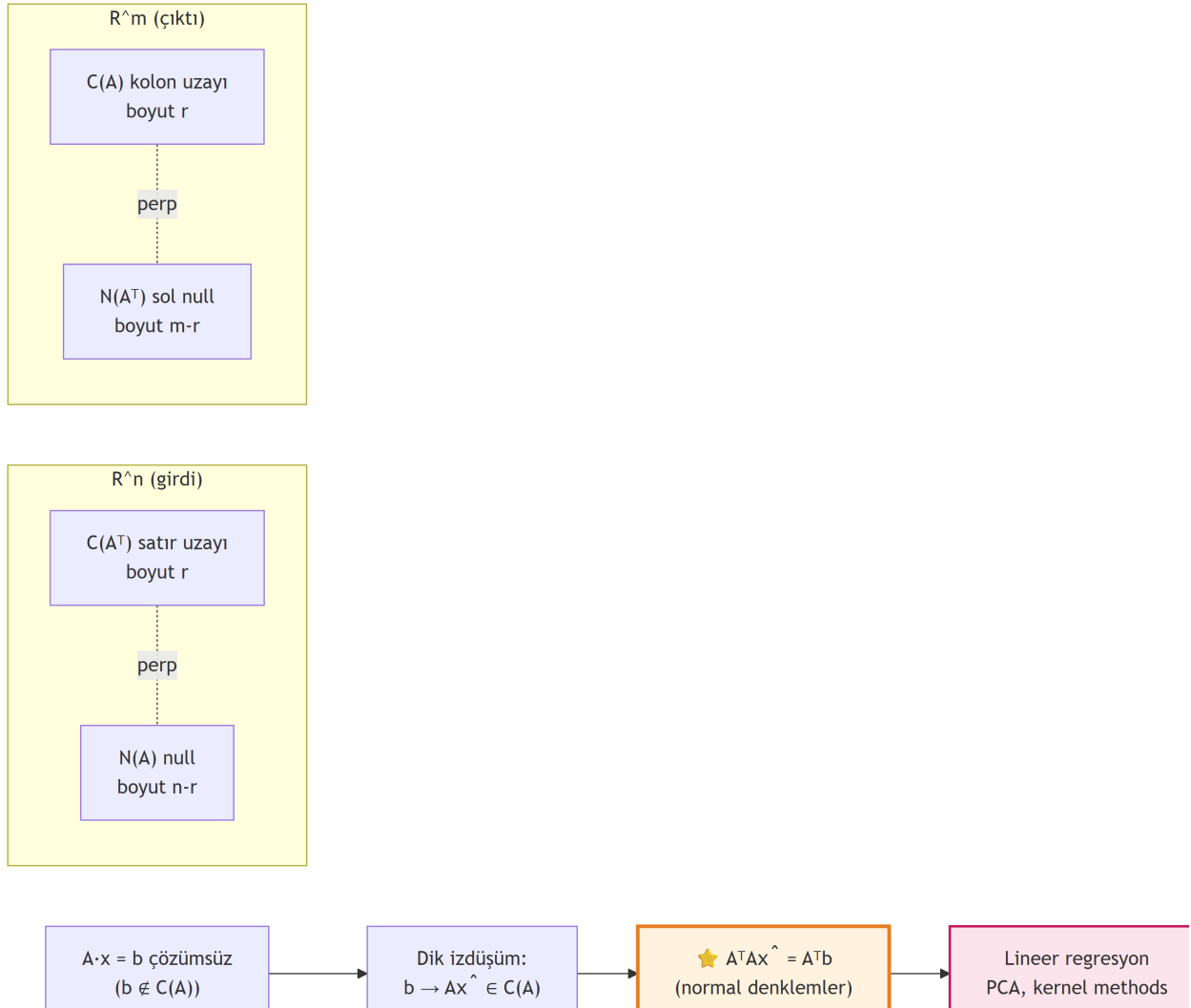
Builder Notu — Diklik ML'in Geometrik Dili

- $\mathbf{x}^T \mathbf{y} = 0 = \text{cosine similarity} = \mathbf{0}$; embedding'lerde “ilgisiz” yönler.
- **Ortogonal ayrışım** = SVD, PCA temeli; veriyi dik bileşenlere ayır.
- **Sıtr \perp null** = projeksiyon, pseudoinverse, “sinyal + gürültü” ayrımı.
- $A^T A$ = normal denklemler / Gram matrisi / kovaryans; least squares + lineer regresyon + kernel methods + PCA çekirdeği.

20.2 Ortogonal Vektörler — $\mathbf{x}^T \mathbf{y} = 0$

$$\mathbf{x}^T \mathbf{y} = x_1 y_1 + x_2 y_2 + \dots + x_n y_n = 0 \iff \mathbf{x} \perp \mathbf{y}$$

Örnek: $\mathbf{x} = (1, 2, 3)$, $\mathbf{y} = (2, -1, 0)$. $\mathbf{x}^T \mathbf{y} = 2 - 2 + 0 = 0 \checkmark$.



Şekil 20.1: Dört alt-uzay = iki dik çift. \mathbb{R}^n ve \mathbb{R}^m ortogonal tümleyenlere ayrılır; $A^T A$ least squares anahtarı.

20.3 Pisagor ve Uzunluk

$$\|\mathbf{x}\|^2 = \mathbf{x}^T \mathbf{x} = x_1^2 + \dots + x_n^2$$

$$\|\mathbf{x} + \mathbf{y}\|^2 = \mathbf{x}^T \mathbf{x} + \mathbf{y}^T \mathbf{y} + 2\mathbf{x}^T \mathbf{y}$$

Pisagor ($\|\mathbf{x}\|^2 + \|\mathbf{y}\|^2 = \|\mathbf{x} + \mathbf{y}\|^2$) $\Leftrightarrow \mathbf{x}^T \mathbf{y} = 0$.

Builder Notu: Normalize edilmiş vektörlerde $\mathbf{x}^T \mathbf{y}$ = **cosine similarity**. Semantik arama, öneri, kontrastif öğrenme.

20.4 Ortogonal Alt-Uzaylar

$S \perp T$: S 'deki her vektör T 'deki her vektöre dik. **Yalnızca 0'da kesişebilirler** (sıfır-olmayan ortak vektör kendine dik olamaz, $\mathbf{v}^T \mathbf{v} > 0$).

Tahta-zemin örneği: \mathbb{R}^3 'te iki düzlem — fiziksel olarak 90° 'de buluşsalar bile **alt-uzay olarak ortogonal değil**, çünkü ortak bir doğruları var.

“Those two planes aren't orthogonal.” — Strang, 18:25

Builder Notu: Bağımsızlık \neq diklik. Gerçek diklik (PCA bileşenleri, ortogonal başlatma, Stiefel manifoldu) çok daha güçlü ve sayısal kararlılık + yorumlanabilirlik sağlar.

20.5 Satır Uzayı \perp Null Uzayı

$\mathbf{x} \in N(A) \rightarrow A\mathbf{x} = \mathbf{0}$:

$$A\mathbf{x} = \begin{pmatrix} \text{satır}_1 \cdot \mathbf{x} \\ \vdots \\ \text{satır}_m \cdot \mathbf{x} \end{pmatrix} = \mathbf{0}$$

Her satır $\cdot \mathbf{x} = 0 \rightarrow \mathbf{x}$ her satıra dik \rightarrow tüm satır uzayına dik.

Aynı argüman A^T 'ye: **kolon uzayı \perp sol null uzayı.**

20.6 Ortogonal Tümleyen

Diklik + boyut: satır uzayı ve null uzayı **ortogonal tümleyen** — null uzayı, satır uzayına dik olan **TÜM** vektörleri içerir.

$$\dim(\text{satır}) + \dim(\text{null}) = r + (n - r) = n$$

“The null space contains all, not just some, vectors that are perpendicular to the row space.” — Strang, 32:42

Örnek: $A = (1, 2, 5)$ (1×3). Satır uzayı = $(1, 2, 5)$ yönündeki doğru. Null uzayı = $(1, 2, 5)$ 'e dik düzlem. Doğru + dik düzlem = \mathbb{R}^3 .

Builder Notu: Bir vektör satır bileşeni + null bileşeni olarak **tek biçimde** ayrışır \rightarrow projeksiyonun, PCA'nın, pseudoinverse'in geometrik motoru. “Sinyal + gürültü” çoğu zaman ortogonal ayrışım.

20.7 Temel Teorem 2. Kısım

Girdi uzayı \mathbb{R}^n :

$$C(A^T) \perp N(A), \quad r + (n - r) = n$$

Çıktı uzayı \mathbb{R}^m :

$$C(A) \perp N(A^T), \quad r + (m - r) = m$$

“We've carved up n -dimensional space into two subspaces, and they're orthogonal complements.” — Strang, 33:23

20.8 En İyi Çözüm Problemi — Least Squares Motivasyonu

Tipik: 1000 ölçüm (denklem), 6 parametre. $m \gg n \rightarrow$ çoğu \mathbf{b} için $\mathbf{b} \notin C(A) \rightarrow$ tam çözüm yok + gürültü.

Kötü: denklemleri at, kare sistem bırak.

İyi: tüm ölçümleri kullan, gürültü/bilgi ayır, en iyi tahmin bul.

“I want to separate the noise from the information.” — Strang, 37:05

Çözüm: \mathbf{b} 'yi $C(A)$ 'ya **izdüşür** (Ders 15-16):

$$\boxed{A^T A \hat{\mathbf{x}} = A^T \mathbf{b}} \quad (\text{normal denklemler})$$

Builder Notu: Bu **linear regresyon**. m örnek, n özellik, $m \gg n$, gürültülü \mathbf{b} . Denetimli öğrenmenin prototipi.

20.9 $A^T A$ Özellikleri

- **Kare** ($n \times n$).
- **Simetrik** ($(A^T A)^T = A^T A$).
- **rank**($A^T A$) = **rank**(A), $N(A^T A) = N(A)$.

“A transpose A is invertible exactly if A has independent columns.” — Strang, 48:51

Niye $N(A^T A) = N(A)$? $A^T A \mathbf{x} = \mathbf{0} \rightarrow \mathbf{x}^T A^T A \mathbf{x} = \|A \mathbf{x}\|^2 = 0 \rightarrow A \mathbf{x} = \mathbf{0}$.

```
A^T A =
[[ 3.  8.]
 [ 8. 30.]]
simetrik mi? True
det = 25.999999999999999
rank(A) = 2, rank(A^T A) = 2
```

💡 Builder Notu — $A^T A$ Her Yerde

- **Lineer regresyon kapalı-form:** $\hat{\mathbf{x}} = (A^T A)^{-1} A^T \mathbf{b}$.
- **Gram matrisi:** girişler kolon çiftlerinin dot product'ları; kernel methods çekirdeği.
- **Kovaryans:** merkezlenmiş veri için $A^T A / (m - 1)$; **PCA** bunun özvektörleri.
- **Multicollinearity** $\rightarrow A^T A$ tekil \rightarrow **ridge** ($A^T A + \lambda I$) düzeltir.

20.10 Bu Dersin Özeti

1. $\mathbf{x}^T \mathbf{y} = 0$.
2. **Pisagor.**
3. **Ortogonal alt-uzaylar.**
4. **Tahta-zemin:** ayrıklık \neq diklik.
5. **Satır \perp null** ($A \mathbf{x} = 0$ ispatı).
6. **Ortogonal tümleyen:** dik olan TÜM vektörler.
7. **Temel teorem 2. kısım.**
8. **En iyi çözüm:** dik izdüşüm.
9. **Normal denklemler** $A^T A \hat{\mathbf{x}} = A^T \mathbf{b}$.
10. $A^T A$ tersinir \iff kolonlar bağımsız.

! Tek bir cümle

Dört alt-uzay boyutlarıyla değil, **diklikleriyle de** tanımlanır: satır \perp null, kolon \perp sol null (ortogonal tümleyen). $A^T A$ çözümsüz sistemleri en iyi anlamda çözmenin (least squares) anahtarı.

20.11 Kontrol Soruları

i Soru 1: $\mathbf{x} = (1,1,1,1)$, $\mathbf{y} = (1,-1,2,-2)$ dik mi? Pisagor.

$$\mathbf{x}^T \mathbf{y} = 1 - 1 + 2 - 2 = 0 \checkmark.$$

$$\text{Pisagor: } \|\mathbf{x}\|^2 = 4, \|\mathbf{y}\|^2 = 10. \mathbf{x} + \mathbf{y} = (2, 0, 3, -1), \|\cdot\|^2 = 4 + 0 + 9 + 1 = 14 = 4 + 10 \checkmark.$$

i Soru 2: $A = (2, 1, 2)$. Satır ve null uzayı ortogonal tümleyen mi?

- Satır uzayı: $(2, 1, 2)$ doğrultusu, dim 1.
- Null uzayı: $2x_1 + x_2 + 2x_3 = 0 \rightarrow (2, 1, 2)$ 'ye dik düzlem, dim $n - r = 2$.

Tümleyen: $1 + 2 = 3 = n \checkmark$. Null uzayı $(2, 1, 2)$ 'ye dik **her** vektörü içerir.

i Soru 3: $A = ((1,2),(2,4),(3,6))$ için $A^T A$ tersinir mi?

$$\mathbf{c}_2 = 2\mathbf{c}_1 \rightarrow \text{bağımlı} \rightarrow \text{rank } 1.$$

$A^T A$ tersinir \iff kolonlar bağımsız. **Hayır, tersinir değil** (rank 1, tekil).

$$A^T A = \begin{pmatrix} 14 & 28 \\ 28 & 56 \end{pmatrix}, \det = 784 - 784 = 0 \checkmark.$$

Ridge: $A^T A + \lambda I$ tersinir.

i Soru 4: Lineer regresyon neden $A^T A \hat{\mathbf{x}} = A^T \mathbf{b}$? Ortogonallikle bağı?

$m \gg n$, gürültülü $\mathbf{b} \rightarrow \mathbf{b} \notin C(A)$.

En iyi tahmin: $A \hat{\mathbf{x}} = \mathbf{b}$ 'nin $C(A)$ 'ya **dik izdüşümü**. Hata $\mathbf{e} = \mathbf{b} - A \hat{\mathbf{x}}$ kolon uzayına dik \rightarrow her kolona dik:

$$A^T (\mathbf{b} - A \hat{\mathbf{x}}) = 0 \implies A^T A \hat{\mathbf{x}} = A^T \mathbf{b}$$

Kolonlar bağımsız $\rightarrow \hat{\mathbf{x}} = (A^T A)^{-1} A^T \mathbf{b}$ benzersiz least squares.

Denetimli öğrenmenin lineer prototipi; ortogonallik “en iyi”yi (dik projeksiyon) tanımlar.

20.12 Egzersizler

Egzersiz 1. $(1, 2, 2, 1)$ 'e dik vektörler \mathbb{R}^4 'te hangi alt-uzay? Boyut, baz.

Egzersiz 2. $A = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$ — satır ve null ortogonal tümleyen mi (boyut + diklik kontrol)?

Egzersiz 3. Hangi A için $A^T A$ tersinir değil?

- (a) $\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$
- (b) $\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$

- (c) $(1, 2, 3)^T$


Egzersiz 4. (Python) $A^T A$ ve diklik kontrolü.

Egzersiz 5. İspatla: $N(A^T A) = N(A)$. (İpucu: $A^T A \mathbf{x} = \mathbf{0} \rightarrow \mathbf{x}^T A^T A \mathbf{x} = \|A \mathbf{x}\|^2 = 0$.)

20.13 Sonraki Ders İçin Hazırlık

Ders 15: Alt-Uzaylara Projeksiyon

- Doğruya projeksiyon: $\mathbf{p} = \frac{\mathbf{a}^T \mathbf{b}}{\mathbf{a}^T \mathbf{a}} \mathbf{a}$.
- Projeksiyon matrisi P ($P^2 = P, P^T = P$).
- Alt-uzaya projeksiyon \rightarrow normal denklemler.


 Ders 15 öncesi

- Egzersiz 5 ($N(A^T A) = N(A)$) kritik.
- A.T @ A ile diklik ilişkilerini doğrula.

20.14 Anahtar Kavramlar (Cheat Sheet)

Kavram	Tanım	Strang'da
Ortogonal vektör	$\mathbf{x}^T \mathbf{y} = 0$	4m20
Pisagor	$\ \mathbf{x}\ ^2 + \ \mathbf{y}\ ^2 = \ \mathbf{x} + \mathbf{y}\ ^2 \iff \mathbf{x}^T \mathbf{y} = 0$	5m41
Uzunluk	$\ \mathbf{x}\ ^2 = \mathbf{x}^T \mathbf{x}$	7m25
Ortogonal alt-uzay	Sadece $\mathbf{0}$ 'da kesişir	16m17
Satır \perp null	$A \mathbf{x} = \mathbf{0}$ her satıra diklik	20m49
Ortogonal tümleyen	Dik olan TÜM vektörler	32m17
Normal denklemler	$A^T A \hat{\mathbf{x}} = A^T \mathbf{b}$	34m21
$A^T A$ tersinir	\iff kolonlar bağımsız	48m51

20.15 ML Bağlantıları Özeti

 7 köprü

1. $\mathbf{x}^T \mathbf{y} = 0 = \cos \sin \mathbf{0} \rightarrow$ Embedding, kontrastif öğrenme.
2. **Ortogonal ayrışım** \rightarrow SVD, PCA.
3. **Satır \perp null** \rightarrow Projeksiyon, pseudoinverse, sinyal/gürültü.
4. $A^T A = \mathbf{normal} / \mathbf{Gram} / \mathbf{kov} \rightarrow$ Lineer regresyon, kernel, PCA.
5. **Ortogonal tümleyen** \rightarrow Bir vektörün dik ayrışımı; boyut indirgemede atılan.
6. $A^T A$ **tersinir = bağımsız özellik** \rightarrow Ridge multicollinearity'yi düzeltir.

7. **Ortogonal kısıt** → RNN/normalizing flow, Stiefel manifold optim.

! Tek bir şey alıp gideceksen

$\mathbf{x}^T \mathbf{y} = 0$ → dik. Dört alt-uzay dik çiftler (ortogonal tümleyen). $A^T A \hat{\mathbf{x}} = A^T \mathbf{b}$ — çözümsüz $A \mathbf{x} = \mathbf{b}$ 'yi en iyi anlamda çözmenin temeli.

21 Alt-Uzaylara Projeksiyon

Hatayı dik yap — normal denklemler ve P matrisi

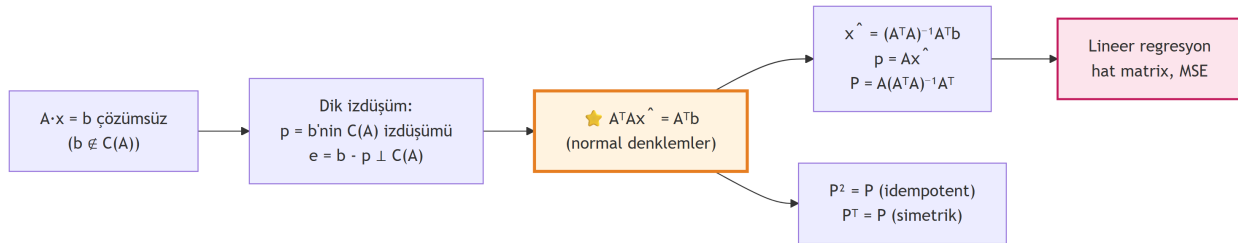
i Bölüm bilgisi

- **Strang'in videosu:** [YouTube — Lecture 15: Projections onto Subspaces](#) (≈48 dk)
- **OCW sayfası:** [MIT 18.06SC — Lecture 15](#)
- **Okuma süresi:** ≈40 dk

21.1 Bu Derste Ne Var?

1. **Doğruya projeksiyon:** $\mathbf{p} = \frac{\mathbf{a}^T \mathbf{b}}{\mathbf{a}^T \mathbf{a}} \mathbf{a}$.
2. **Projeksiyon matrisi** $P = \mathbf{a} \mathbf{a}^T / \mathbf{a}^T \mathbf{a}$ — $P^T = P, P^2 = P$.
3. **Alt-uzaya projeksiyon** → **normal denklemler** $A^T A \hat{\mathbf{x}} = A^T \mathbf{b}$.
4. **Uygulama:** noktalara doğru fitleme.

“A transpose A x hat equals A transpose b. That’s the central equation of the subject.” — Strang, 27:52



Şekil 21.1: Çözümsüz $Ax = b \rightarrow b$ 'yi $C(A)$ 'ya dik izdüşür → normal denklemler → P matrisi.

💡 Builder Notu — Projeksiyon = ML'in En Aza İndirme Geometrisi

- **Normal denklemler = lineer regresyonun kapalı-form.**
- $P = \text{hat matrix}$ istatistikte: $\hat{\mathbf{y}} = P\mathbf{b}$; köşegen = leverage, $\text{tr}(P) = \text{rank}$.
- $\mathbf{e} \perp C(A)$ = artıkların özelliği; **MSE** kaybının kapalı-form ve dik geometrisinin sebebi.
- **Rank-1 projeksiyon** = bir embedding'i tek yöne projelendirme (activation steering, concept erasure).

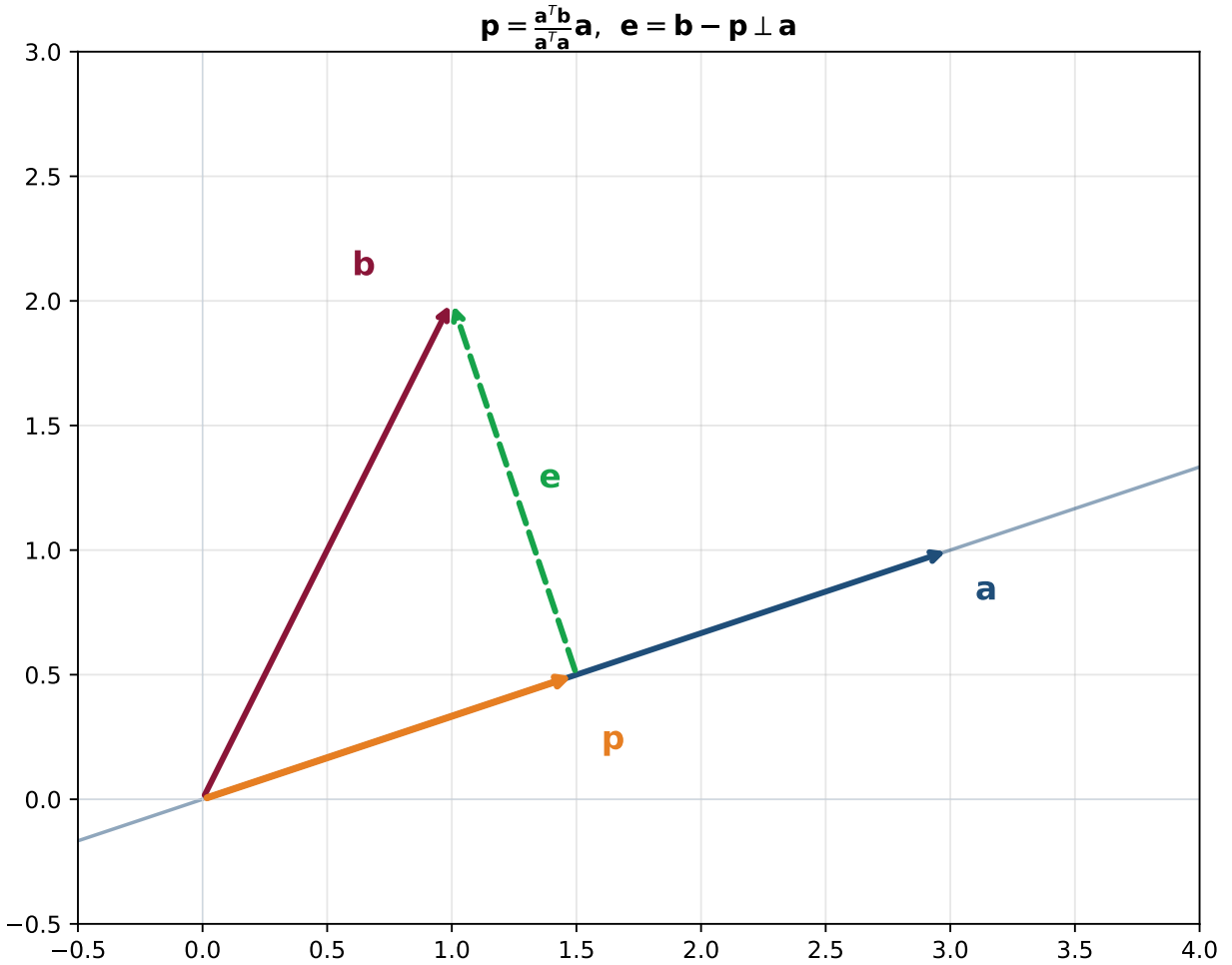
21.2 Doğruya Projeksiyon

\mathbf{b} vektörünü \mathbf{a} doğrultusuna izdüşür: $\mathbf{p} = x\mathbf{a}$. Hata $\mathbf{e} = \mathbf{b} - x\mathbf{a}$, \mathbf{a} 'ya **dik** olmalı:

$$\mathbf{a}^T(\mathbf{b} - x\mathbf{a}) = 0 \implies x = \frac{\mathbf{a}^T\mathbf{b}}{\mathbf{a}^T\mathbf{a}}, \quad \mathbf{p} = \mathbf{a} \frac{\mathbf{a}^T\mathbf{b}}{\mathbf{a}^T\mathbf{a}}$$

“The key to everything is that perpendicular.” — Strang, 3:17

Sezgi: $\mathbf{b} \rightarrow 2\mathbf{b} \rightarrow \mathbf{p} \rightarrow 2\mathbf{p}$. $\mathbf{a} \rightarrow 2\mathbf{a} \rightarrow \mathbf{p}$ değişmez (doğru aynı).



Şekil 21.2: \mathbf{b} vektörünün \mathbf{a} doğrusuna projeksiyonu \mathbf{p} ; hata $\mathbf{e} = \mathbf{b} - \mathbf{p}$, \mathbf{a} 'ya dik.

$$x = 0.5000, \quad \mathbf{a}^T \cdot \mathbf{e} = 0.000000$$

21.3 Projeksiyon Matrisi P (1-D)

$$\mathbf{p} = \frac{\mathbf{a}\mathbf{a}^T}{\underbrace{\mathbf{a}^T\mathbf{a}}_P} \mathbf{b}$$

$\mathbf{a}\mathbf{a}^T$ = matris ($n \times n$); $\mathbf{a}^T\mathbf{a}$ = sayı. Sadeleşmez!

Özellikleri:

- $C(P) = \mathbf{a}$ doğrusu.
- **Rank 1** (kolon \times satır).

21.4 İki İmza Özelliği — $P^T = P, P^2 = P$

Simetrik: $P^T = (\mathbf{a}\mathbf{a}^T / \mathbf{a}^T\mathbf{a})^T = P$.

İdempotent: $P^2 = P$. **Sezgi:** ikinci kez izdüürürsen zaten oradasın.

“*P transpose equals P ... P squared equals P.*” — Strang, 13:18

Bu iki özellik bir matrisin projeksiyon matrisi olduğunu **karakterize eder**.

21.5 Çözümsüz $\mathbf{Ax} = \mathbf{b} \rightarrow$ Projeksiyon

\mathbf{Ax} hep $C(A)$ 'da; \mathbf{b} değilse $\rightarrow \mathbf{b}$ 'nin $C(A)$ 'ya **en yakın** noktasını $\mathbf{p} = \mathbf{Ax}$:

$$A\hat{\mathbf{x}} = \mathbf{p}$$

$\hat{\mathbf{x}}$ = en iyi yaklaşım, gerçek çözüm değil.

Builder Notu: Bu **denetimli öğrenmenin özü:** hedef \mathbf{b} modelin uzayında yok \rightarrow en iyi yaklaşım. Kayıp minimize = dik projeksiyon.

21.6 Alt-Uzaya Projeksiyon — Normal Denklemler

$\mathbf{p} = A\hat{\mathbf{x}}$. Hata $\mathbf{e} = \mathbf{b} - A\hat{\mathbf{x}}$ kolon uzayına dik \rightarrow her kolona dik:

$$A^T(\mathbf{b} - A\hat{\mathbf{x}}) = \mathbf{0} \implies \boxed{A^T A\hat{\mathbf{x}} = A^T \mathbf{b}}$$

Normal denklemler — bölümün merkez denklemi.

Bağlantı: $\mathbf{e} \in N(A^T)$ (Ders 14 sol null) $\iff \mathbf{e} \perp C(A)$. Teori kendini doğruluyor.

21.7 Üç Formül

$$\hat{\mathbf{x}} = (A^T A)^{-1} A^T \mathbf{b}, \quad \mathbf{p} = A\hat{\mathbf{x}} = A(A^T A)^{-1} A^T \mathbf{b}, \quad P = A(A^T A)^{-1} A^T$$

Tuzak: $P = AA^{-1}(A^T)^{-1}A^T = I$ diye **sadeleştiremezsin!** Çünkü A kare değil.

21.8 A Kare ise P = I

A kare ve tersinir $\rightarrow C(A) = \mathbb{R}^n \rightarrow$ her \mathbf{b} zaten kolon uzayında $\rightarrow P = I$.

$$P = A(A^T A)^{-1} A^T = AA^{-1}(A^T)^{-1} A^T = I$$

(Burada ayrıştırma geçerli; A tersinir.)

Builder Notu: P istatistikte “**hat matrix**” $H: \hat{\mathbf{y}} = H\mathbf{y}$. Köşegen = **leverage**; $\text{tr}(P) = \text{rank} =$ parametre sayısı = modelin serbestlik derecesi.

21.9 Doğru Fitleme — Sayısal Örnek

Noktalar $(t, b) = (1, 1), (2, 2), (3, 2)$. Doğru: $b = C + Dt$.

$$A = \begin{pmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} C \\ D \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix}$$

3 denklem, 2 bilinmeyen \rightarrow çözümsüz. Normal denklemler:

$$A^T A = \begin{pmatrix} 3 & 6 \\ 6 & 14 \end{pmatrix}, \quad A^T \mathbf{b} = \begin{pmatrix} 5 \\ 11 \end{pmatrix}$$

Sayısal çözüm $C = 2/3, D = 1/2$ (Ders 16’da tamamlanır).

$C = 0.6667, D = 0.5000$
 $P^2 = P?$ True
 $P^T = P?$ True
 $\text{tr}(P) = 2.00 = \text{rank}(A) = 2$
 $A^T \mathbf{e} \approx \mathbf{0}?$ True

21.10 Least Squares Geometrisi

Hata $e_i = b_i - (C + Dt_i)$. Minimize:

$$\|\mathbf{e}\|^2 = \sum e_i^2 = \|\mathbf{b} - A\mathbf{x}\|^2$$

İki bakış:

- **Cebirsel:** $\|\mathbf{b} - A\mathbf{x}\|^2$ türev = 0 → normal denklemler.
- **Geometrik:** \mathbf{b} 'yi $C(A)$ 'ya dik izdüşür; en kısa hata diktir.

İkisi de aynı denklemlere çıkar — **least squares = dik projeksiyon.**

Builder Notu: Kare hata (MSE) dik projeksiyon → temiz kapalı-form. **L1, Huber** farklı geometriler verir, iteratif çözüm gerekir.

21.11 Bu Dersin Özeti

1. **Doğruya projeksiyon** ($\mathbf{e} \perp \mathbf{a}$).
2. $P_{1D} = \mathbf{a}\mathbf{a}^T / \mathbf{a}^T \mathbf{a}$, rank 1.
3. $P^T = P, P^2 = P$.
4. **Çözümsüz** $A\mathbf{x} = \mathbf{b}$ → en yakın \mathbf{p} .
5. **Alt-uzaya projeksiyon:** $\mathbf{e} \perp C(A)$.
6. **Normal denklemler** $A^T A \hat{\mathbf{x}} = A^T \mathbf{b}$.
7. **Üç formül.**
8. **A kare** → $P = I$.
9. **Doğru fitleme.**
10. **Least squares = dik projeksiyon.**

! Tek bir cümle

Projeksiyon = hatayı alt-uzaya dik yapmak; normal denklemler $A^T A \hat{\mathbf{x}} = A^T \mathbf{b}$; $P = A(A^T A)^{-1} A^T$ (idempotent + simetrik) — çözümsüz $A\mathbf{x} = \mathbf{b}$ 'yi least squares ile çözmenin tam reçetesi.

21.12 Kontrol Soruları

i Soru 1: $\mathbf{b} = (1,1,1)$, $\mathbf{a} = (1,2,2)$ — projeksiyon.

$$x = \frac{1+2+2}{1+4+4} = \frac{5}{9}, \mathbf{p} = \frac{5}{9}(1, 2, 2) = \left(\frac{5}{9}, \frac{10}{9}, \frac{10}{9}\right).$$

$$\mathbf{e} = (4/9, -1/9, -1/9). \mathbf{a}^T \mathbf{e} = 4/9 - 2/9 - 2/9 = 0 \checkmark.$$

i Soru 2: $\mathbf{a} = (1, 1)$ için P , $P^2 = P$.

$$P = \frac{1}{2} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

$P^2 = P$ ✓, $P^T = P$ ✓, rank 1. Her (x, y) 'yi eşit-bileşen doğrusuna → **ortalama** alır.

i Soru 3: A kare tersinir → P ?

$$P = A(A^T A)^{-1} A^T = A A^{-1} (A^T)^{-1} A^T = I.$$

Her \mathbf{b} zaten $C(A) = \mathbb{R}^n$ 'de; projeksiyon değiştirmez.

i Soru 4: Lineer regresyon → projeksiyon + normal denklemler bağı.

Veri A , hedef \mathbf{b} . $m \gg n \rightarrow$ çözümsüz.

Least squares = projeksiyon: $\|\mathbf{b} - A\mathbf{x}\|^2$ minimize = \mathbf{b} 'nin $C(A)$ 'ya dik izdüşümü → $A^T A \hat{\mathbf{x}} = A^T \mathbf{b}$.

Tahminler $\hat{\mathbf{y}} = P\mathbf{b}$ (hat matrix).

Pratik:

- $A^T A$ tekil → **ridge** ($A^T A + \lambda I$).
- Sayısal: **QR** (Ders 17) veya **SVD**.
- Polinom = A 'ya kolon ekleme.

21.13 Egzersizler

Egzersiz 1. $\mathbf{b} = (2, 3, 4)^T$, $\mathbf{a} = (1, 1, 1)^T$ — projeksiyon, hata, $\mathbf{a}^T \mathbf{e} = 0$ doğrula.

Egzersiz 2. $\mathbf{a} = (1, 2, 2)^T$ için P (3×3). $P^2 = P$, $\text{tr}(P) = 1$.

Egzersiz 3. $(0, 1), (1, 3), (2, 4) \rightarrow$ en iyi doğru. A , \mathbf{b} , $A^T A$, $A^T \mathbf{b}$.


Egzersiz 4. (Python) Projeksiyon ve regresyon.

Egzersiz 5. İspatla: $P = A(A^T A)^{-1} A^T$ için $P^2 = P$. Ayrıca $\text{tr}(P) = \text{rank}(A)$ (P özdeğerleri 0 ve 1).

21.14 Sonraki Ders İçin Hazırlık

Ders 16: Projeksiyon Matrisleri ve Least Squares

- Doğru fitleme sayısal çözüm ($C = 2/3, D = 1/2$).
- $\mathbf{p} + \mathbf{e}$ dik bileşenler.
- $A^T A$ tersinirliğinin tam ispatı.

 Ders 16 öncesi

- Egzersiz 5 ($P^2 = P$).
- `np.linalg.lstsq` ile regresyon.


21.15 Anahtar Kavramlar (Cheat Sheet)

Kavram	Tanım	Strang'da
Doğruya projeksiyon	$\mathbf{p} = (\mathbf{a}^T \mathbf{b} / \mathbf{a}^T \mathbf{a}) \mathbf{a}$	0m36
P_{1D}	$\mathbf{a} \mathbf{a}^T / \mathbf{a}^T \mathbf{a}$, rank 1	8m51
İki özellik	$P^T = P, P^2 = P$	13m18
Alt-uzaya projeksiyon	$\mathbf{p} = A \hat{\mathbf{x}}, \mathbf{e} \perp C(A)$	18m50
Normal denklemler	$A^T A \hat{\mathbf{x}} = A^T \mathbf{b}$	27m52
$\hat{\mathbf{x}}$	$(A^T A)^{-1} A^T \mathbf{b}$	32m55
P_{nD}	$A(A^T A)^{-1} A^T$	35m24
A kare $\rightarrow P = I$	Tüm uzaya projeksiyon	39m18
Least squares	$\ \mathbf{b} - A\mathbf{x}\ ^2$ minimize	42m42

21.16 ML Bağlantıları Özeti

 7 köprü

1. **Normal denklemler** \rightarrow Lineer regresyon kapalı-form.
2. $P = \text{hat matrix}$ \rightarrow Leverage, serbestlik dereceleri.
3. $\mathbf{e} \perp C(A)$ \rightarrow Artıklar dikliği; MSE neden temiz.
4. **Least squares = MSE** \rightarrow Dik projeksiyon; L1/Huber farklı geometri.
5. **Rank-1 projeksiyon** \rightarrow Activation steering, concept erasure, PCA tek bileşen.
6. $A^T A$ tekil \rightarrow ridge.
7. **Polinom/özellik mühendisliği** \rightarrow A 'ya kolon ekleme.

 Tek bir şey alıp gideceksen

Projeksiyon = hatayı dik yapmak; $A^T A \hat{\mathbf{x}} = A^T \mathbf{b}$; $P = A(A^T A)^{-1} A^T$ — çözümsüz $A\mathbf{x} = \mathbf{b}$ 'yi least squares ile çözmek = lineer regresyonun ta kendisi.

22 Projeksiyon Matrisleri ve Least Squares

Lineer regresyonu uçtan uca — $\mathbf{b} = \mathbf{p} + \mathbf{e}$, $A^T A \hat{\mathbf{x}} = A^T \mathbf{b}$

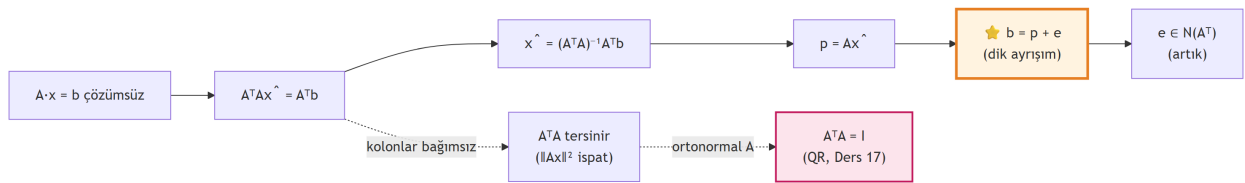
i Bölüm bilgisi

- **Strang'ın videosu:** [YouTube — Lecture 16: Projection Matrices and Least Squares](#) (≈48 dk)
- **OCW sayfası:** [MIT 18.06SC — Lecture 16](#)
- **Okuma süresi:** ≈40 dk

22.1 Bu Derste Ne Var?

1. **Projeksiyon recap:** $\mathbf{b} \in C(A) \rightarrow P\mathbf{b} = \mathbf{b}$; $\mathbf{b} \perp C(A) \rightarrow P\mathbf{b} = \mathbf{0}$.
2. **Doğru fitleme sayısal:** $(1, 1), (2, 2), (3, 2) \rightarrow y = 2/3 + (1/2)t$.
3. **İki resim:** noktalar+doğru vs vektörler ($\mathbf{b} = \mathbf{p} + \mathbf{e}$).
4. $A^T A$ tersinir \iff **bağımsız kolonlar** — tam ispat.
5. **Ortonormal vektörler** ($A^T A = I$) — Ders 17 köprüsü.

“A transpose A x hat equals A transpose b — the most important equation in statistics and estimation.” — Strang, 21:23



Şekil 22.1: Least squares uçtan uca: $A \rightarrow A^T A \rightarrow \hat{\mathbf{x}} \rightarrow \mathbf{p} + \mathbf{e} \rightarrow$ ortonormal son.

💡 Builder Notu — Lineer Regresyon Uçtan Uca

- **Least squares = MSE regresyon** kapalı-form.
- $\mathbf{b} = \mathbf{p} + \mathbf{e}$ = fit + residual; artık analizi model teşhisi.
- **Outlier** → **robust loss** (L1, Huber); kareler outlier'a duyarlı.
- $A^T A$ tersinir = **bağımsız özellik** → multicollinearity'de ridge.
- **Ortonormal** → **QR / kararlılık** → Ortonormal başlatma gradyanları korur.

22.2 Projeksiyon Recap — İki Uç

$\mathbf{b} \in C(A)$: $P\mathbf{b} = A(A^T A)^{-1}(A^T A)\mathbf{x} = A\mathbf{x} = \mathbf{b} \checkmark$.

$\mathbf{b} \perp C(A)$: $A^T \mathbf{b} = \mathbf{0} \rightarrow P\mathbf{b} = A(A^T A)^{-1} \cdot \mathbf{0} = \mathbf{0} \checkmark$.

22.3 $\mathbf{b} = \mathbf{p} + \mathbf{e}$ ve $I - P$

$$\mathbf{b} = \mathbf{p} + \mathbf{e}, \quad \mathbf{p} = P\mathbf{b}, \quad \mathbf{e} = (I - P)\mathbf{b}$$

$I - P$ dik tümleyene projeksiyon: simetrik, idempotent, $C(A)$ 'ya değil $N(A^T)$ 'ye projeler.

Builder Notu: “Fit + residual” ayrışımı. Artıkların yapısı (rastgele mi, desenli mi) model eksikliğini gösterir. PCA'da da veri = bileşenler (P) + kalan ($I - P$).

22.4 Doğru Fitleme — Sayısal Çözüm

$$A^T A = \begin{pmatrix} 3 & 6 \\ 6 & 14 \end{pmatrix}, \quad A^T \mathbf{b} = \begin{pmatrix} 5 \\ 11 \end{pmatrix}.$$

$$3C + 6D = 5, \quad 6C + 14D = 11 \rightarrow 2D = 1 \rightarrow D = 1/2, \quad C = 2/3.$$

$$y = \frac{2}{3} + \frac{1}{2}t$$

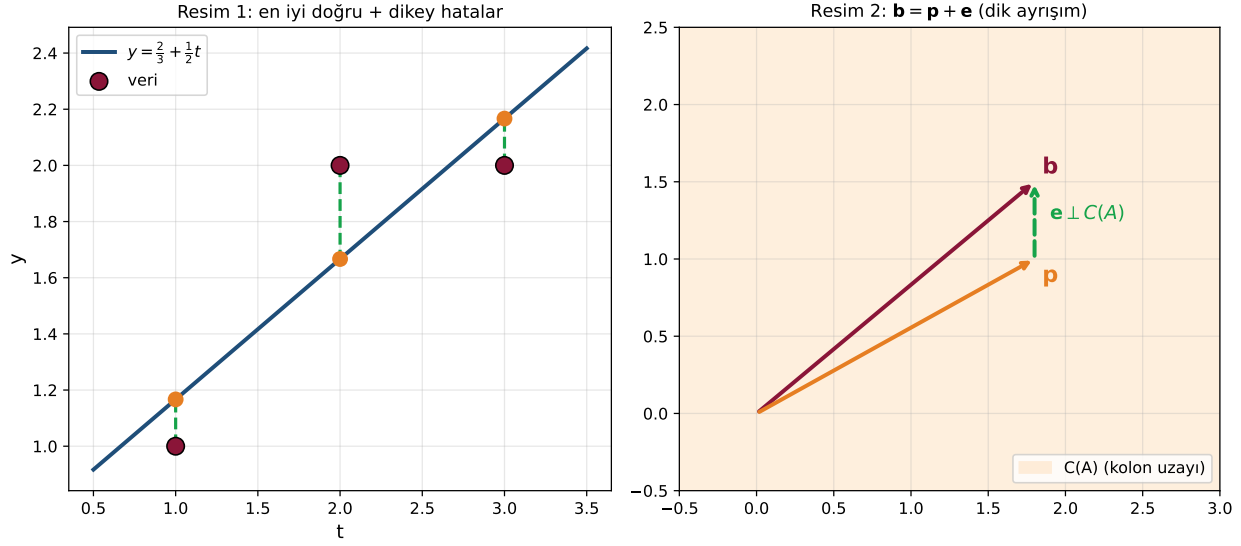
```
C = 0.6667, D = 0.5000
p = [1.16666667 1.66666667 2.16666667]
e = [-0.16666667 0.33333333 -0.16666667]
e |_| kolonlar? True
Pisagor: ||b||2 = 9.0000, ||p||2 + ||e||2 = 9.0000
```

22.5 İki Resim — Noktalar vs Vektörler

Resim 1 (noktalar): $p_1 = 7/6, p_2 = 5/3, p_3 = 13/6$. Hatalar $e_i = b_i - p_i = (-1/6, 2/6, -1/6)$.

Resim 2 (\mathbb{R}^3 vektörler): $\mathbf{b} = (1, 2, 2) \rightarrow C(A)$ 'ya \mathbf{p} , dik kalan \mathbf{e} .

Aynı sayılar, iki bakış: parametre uzayı (C, D) vs veri uzayı (3 nokta).



Şekil 22.2: Resim 1: noktalar + en iyi doğru, dikey hatalar. Resim 2: 3D vektörler (gösterim için 2D'ye düşürülmüş kavramsal).

22.6 Least Squares = Kalkülüs

$$\|\mathbf{e}\|^2 = \sum (C + Dt_i - b_i)^2$$

$\partial/\partial C = 0, \partial/\partial D = 0$ — kare olduğu için türevler **lineer** → tam olarak normal denklemler.

İki yol aynı yere: **lineer cebir (projeksiyon) = kalkülüs (minimizasyon)**.

22.7 Outlier Uyarısı

Kareler aykırı değerlere aşırı duyarlı — bir outlier doğruyu çeker.

“Statisticians would not be happy to see the whole problem turned topsy-turvy by this one outlier.”
— Strang, 16:21

💡 Builder Notu — Robust Loss

- **L2 (kare)**: kapalı-form, ama outlier'a duyarlı.
- **L1 (mutlak)**: outlier'a dayanıklı, iteratif; medyan-benzeri.
- **Huber**: küçükte L2, büyükte L1; robust regresyon standardı.

22.8 $\mathbf{e} \perp$ Kolon Uzayı — Doğrulama

$\mathbf{e} = (-1/6, 2/6, -1/6)$. Kolonlar $(1, 1, 1)$ ve $(1, 2, 3)$:

$$\mathbf{e} \cdot (1, 1, 1) = -\frac{1}{6} + \frac{2}{6} - \frac{1}{6} = 0 \checkmark$$

$$\mathbf{e} \cdot (1, 2, 3) = -\frac{1}{6} + \frac{4}{6} - \frac{3}{6} = 0 \checkmark$$

22.9 $\mathbf{A}^T \mathbf{A}$ Tersinir \iff Bağımsız Kolonlar

İddia: A 'nın kolonları bağımsız $\rightarrow A^T A$ tersinir.

İspat: $A^T A \mathbf{x} = \mathbf{0}$ varsay. Soldan \mathbf{x}^T :

$$\mathbf{x}^T A^T A \mathbf{x} = (\mathbf{A}\mathbf{x})^T (\mathbf{A}\mathbf{x}) = \|\mathbf{A}\mathbf{x}\|^2 = 0 \implies \mathbf{A}\mathbf{x} = \mathbf{0}$$

Bağımsız kolonlar $\rightarrow \mathbf{A}\mathbf{x} = \mathbf{0} \rightarrow \mathbf{x} = \mathbf{0}$. Demek ki $N(A^T A) = \{\mathbf{0}\} \rightarrow$ tersinir. ■

Builder Notu: Regresyonun benzersiz çözümü için bağımsız özellikler şart. Multicollinearity $\rightarrow A^T A$ tekil \rightarrow ridge ($A^T A + \lambda I$) düzeltir. $A^T A$ ayrıca **pozitif yarı-tanımlı** (Ders 27).

22.10 Ortonormal Vektörler — $\mathbf{A}^T \mathbf{A} = \mathbf{I}$

Ortonormal = dik + birim. $A^T A$ 'nın (i, j) girişi = kolon _{i} · kolon _{j} :

$$A^T A = I$$

Sonuç: $\hat{\mathbf{x}} = A^T \mathbf{b}$, $P = A A^T$ — **tersine alma yok**.

“Those are the best columns you could ask for.” — Strang, 46:23

💡 Builder Notu — Ortonormal = Sayısal Kararlılık

- **QR ayrışımı:** Regresyonu $A^T A$ tersine almadan ortonormal Q ile çözer.
- **Ortogonal ağırlık başlatma:** Derin ağlarda gradyan patlamasını önler.
- **Ortonormal bazlar:** PCA, SVD (U, V), Fourier, wavelet.
- **Ortogonal/üniter katmanlar:** RNN stabilizasyonu, normalizing flows.

22.11 Bu Dersin Özeti

1. Projeksiyon uç durumları.
2. $\mathbf{b} = \mathbf{p} + \mathbf{e}$; $I - P$ dik tümleyene.
3. Doğru fitleme: $y = 2/3 + (1/2)t$.
4. İki resim.
5. Least squares = kalkülüs.
6. Outlier uyarısı.
7. $\mathbf{e} \perp C(A)$ doğrulandı.
8. $A^T A$ tersinir \iff bağımsız kolonlar ($\|Ax\|^2$ ispat).
9. Ortonormal ($A^T A = I$).
10. Ortonormal = en iyi kolonlar (Ders 17).

! Tek bir cümle

Least squares = çözümsüz $Ax = \mathbf{b}$ 'yi dik projeksiyonla çözer ($A^T A \hat{\mathbf{x}} = A^T \mathbf{b}$); $\mathbf{e} \perp C(A)$; yalnız bağımsız kolonlarda benzersiz, ortonormal kolonlarda en sade ($A^T A = I$).

22.12 Kontrol Soruları

i Soru 1: (0,0), (1,1), (2,1) \rightarrow en iyi doğru.

$$A^T A = \begin{pmatrix} 3 & 3 \\ 3 & 5 \end{pmatrix}, A^T \mathbf{b} = (2, 3)^T.$$

$$3C + 3D = 2, 3C + 5D = 3 \rightarrow 2D = 1, D = 1/2; C = 1/6.$$

$$y = 1/6 + (1/2)t.$$

i Soru 2: Pisagor doğrulaması.

$$\|\mathbf{b}\|^2 = 9. \|\mathbf{e}\|^2 = 6/36 = 1/6. \|\mathbf{p}\|^2 = 53/6.$$

$$\|\mathbf{p}\|^2 + \|\mathbf{e}\|^2 = 53/6 + 1/6 = 9 = \|\mathbf{b}\|^2 \checkmark \text{ (dik ayrışım sonucu).}$$

i Soru 3: $A = ((1,2),(2,4),(0,0))$ — $A^T A$ tersinir mi?

$\mathbf{c}_2 = 2\mathbf{c}_1 \rightarrow$ bağımlı $\rightarrow A^T A$ tersinir DEĞİL.
Regresyon benzersiz çözümler; bağımlı kolonu at ya da ridge.

i Soru 4: Outlier \rightarrow robust loss. Ortonormal kolonların avantajı?

Outlier: L2 kareler outlier'ı abartır \rightarrow L1 (medyan), **Huber** (küçükte L2, büyükte L1) ya da RANSAC.
Ortonormal ($A^T A = I$):

- Normal denklemler $\hat{\mathbf{x}} = A^T \mathbf{b}$ (tersine alma yok).
- **Sayısal kararlılık** (koşul sayısı 1) \rightarrow QR ayrışımı pratikte tercih edilir.

- Katsayılar bağımsız.
- ML'de: ortogonal başlatma, PCA/SVD, Fourier, normalizing flows.

22.13 Egzersizler

Egzersiz 1. $(-1, 1), (0, 0), (1, 1) \rightarrow$ en iyi doğru. (Simetri: $D = ?$.)

Egzersiz 2. Aynı noktalara **parabol** $y = C + Dt + Et^2$ — tam çözülür mü?

Egzersiz 3. $(I - P)^2 = I - P$ göster, hangi alt-uzaya projeler?

Egzersiz 4. (Python) `np.linalg.lstsq` ile regresyon + Pisagor doğrulama.

Egzersiz 5. İspatla: $A^T A$ pozitif yarı-tanımlı ($\mathbf{x}^T A^T A \mathbf{x} \geq 0$); bağımsız kolonlarda pozitif tanımlı (> 0 for $\mathbf{x} \neq \mathbf{0}$). Ders 27 ve Cholesky temeli.

22.14 Sonraki Ders İçin Hazırlık

Ders 17: Ortogonal Matrisler ve Gram-Schmidt

- Q matrisi ($Q^T Q = I$), $P = Q Q^T$.
- **Gram-Schmidt** — bazı ortonormale çevir.
- $A = QR$ ayrışımı.

⚠ Ders 17 öncesi

- Egzersiz 5 (pozitif tanım).
- `np.linalg.qr` ile regresyon dene.

22.15 Anahtar Kavramlar (Cheat Sheet)

Kavram	Tanım	Strang'da
Uç durumlar	$\mathbf{b} \in C \rightarrow P\mathbf{b} = \mathbf{b}$; dik $\rightarrow \mathbf{0}$	1m01
$\mathbf{b} = \mathbf{p} + \mathbf{e}$	Dik ayrışım	6m24
$I - P$	Dik tümleyene projeksiyon	7m30
Doğru fitleme	$y = 2/3 + (1/2)t$	8m46
Normal denklemler	$A^T A \hat{\mathbf{x}} = A^T \mathbf{b}$	21m23
LS = kalkülüs	$\ \mathbf{e}\ ^2$ türev = 0	24m16
Outlier	Kareler abartır \rightarrow robust loss	16m14
$\mathbf{e} \perp C(A)$	Her kolona dik	33m22
$A^T A$ tersinir	\Leftrightarrow bağımsız ($\ A\mathbf{x}\ ^2 = 0$)	37m07

Kavram	Tanım	Strang'da
Ortonormal	$A^T A = I$	46m23

22.16 ML Bağlantıları Özeti

💡 7 köprü

1. **Normal denklemler = lineer regresyon** kapalı-form.
2. **fit + residual** → Artık analizi model teşhisi.
3. **Outlier** → **robust loss** → L1, Huber.
4. $A^T A$ **tersinir = bağımsız özellik** → Ridge multicollinearity.
5. P = **hat matrix** → Leverage, serbestlik derecesi.
6. **Ortonormal** → **QR / kararlılık** → Ortogonal başlatma.
7. $A^T A$ **pozitif (yarı-)tanım** → Cholesky, GP kovaryans.

❗ Tek bir şey alıp gideceksen

Least squares = dik projeksiyon ($A^T A \hat{\mathbf{x}} = A^T \mathbf{b}$) = lineer regresyon. $\mathbf{b} = \mathbf{p} + \mathbf{e}$, $\mathbf{e} \perp C(A)$. Yalnız bağımsız kolonlarda benzersiz; ortonormal en sade ve kararlı.

23 Ortogonal Matrisler ve Gram-Schmidt

$Q^T Q = I$, $A = QR$ — sayısal LA'nın kararlılık motoru

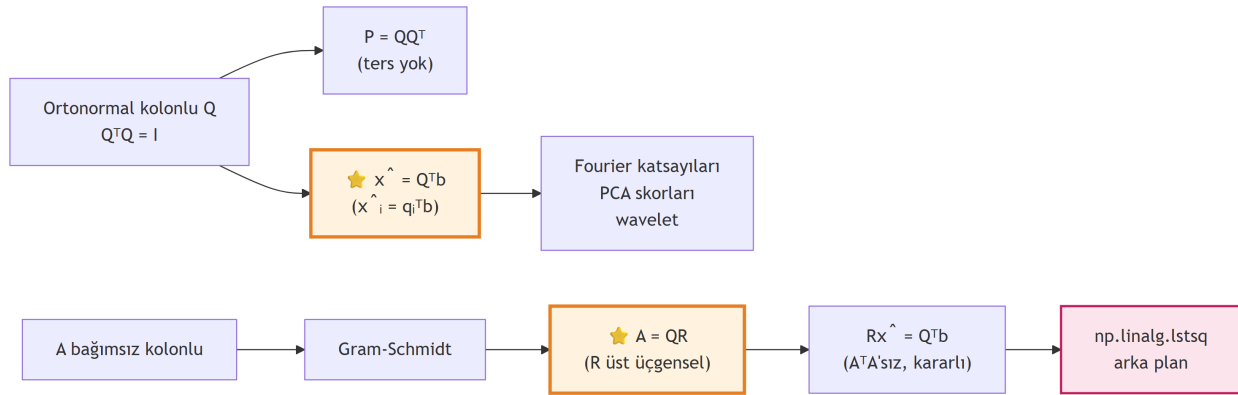
i Bölüm bilgisi

- **Strang'ın videosu:** [YouTube — Lecture 17: Orthogonal Matrices and Gram-Schmidt](#) (≈48 dk)
- **OCW sayfası:** [MIT 18.06SC — Lecture 17](#)
- **Okuma süresi:** ≈40 dk

23.1 Bu Derste Ne Var?

1. **Ortonormal vektörler** ($Q^T Q = I$).
2. **Ortogonal matris** (kare Q): $Q^T = Q^{-1}$.
3. **Projeksiyon** $P = QQ^T$, normal denklemler $\hat{\mathbf{x}} = Q^T \mathbf{b}$.
4. **Gram-Schmidt** $\rightarrow A = QR$ ayrışımı.

“A equals QR is the magic formula — the expression of Gram-Schmidt.” — Strang, 45:25



Şekil 23.1: Ortonormallik \rightarrow ters alma yok. Gram-Schmidt herhangi bir bazı ortonormale çevirir.

💡 Builder Notu — QR Sayısal LA + ML Kararlılık

- **Ortogonal Q = rotasyon/yansıma** $\rightarrow \|Q\mathbf{x}\| = \|\mathbf{x}\|$; ortogonal başlatma gradyanları korur.
- $\hat{\mathbf{x}}_i = \mathbf{q}_i^T \mathbf{b}$ \rightarrow Fourier, PCA, wavelet.
- $A = QR$ \rightarrow Regresyon $A^T A$ 'sız çözer; sayısal kararlı (`np.linalg.qr`).

- $A^T A$ 'nın koşul sayısı = $\kappa(A)^2 \rightarrow$ Doğrudan çözmek tehlikeli, QR/SVD tercih.

23.2 Ortonormal Vektörler ve Q

$$\mathbf{q}_i^T \mathbf{q}_j = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases}$$

Vektörler bir matrisin kolonları $\rightarrow Q$. Ortonormal vektörler her zaman bağımsız.

23.3 $Q^T Q = I$

$$(Q^T Q)_{ij} = \mathbf{q}_i^T \mathbf{q}_j \rightarrow Q^T Q = I.$$

Dikkat: Q kare olmak zorunda değil. 4×2 Q için $Q^T Q$ (2×2) = I , ama $Q Q^T$ farklı (projeksiyon).

Builder Notu: $Q^T Q = I =$ **izometri** (uzunluk korur): $\|Q\mathbf{x}\|^2 = \mathbf{x}^T Q^T Q \mathbf{x} = \|\mathbf{x}\|^2$. Spektral normun 1 olması (kararlı eğitim), normalizing flow hacim koruma.

23.4 Ortogonal Matris (Kare) — $Q^T = Q^{-1}$

Kare Q ortonormal kolonlu ise **ortogonal matris**. $Q^T Q = I \rightarrow Q^T = Q^{-1}$. Tersi bedava.

Örnekler:

- **Permütasyon:** birim vektörler yer değişmiş.
- **Rotasyon:** $\begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}$.
- **Hadamard** ($\frac{1}{2} \cdot \pm 1$ 'lerden).

Builder Notu: Rotasyonlar veri artırma; permütasyon shuffle; Hadamard fast transforms (Performer attention yaklaşımı).

23.5 Projeksiyon $P = Q Q^T$

$$P = A(A^T A)^{-1} A^T, \text{ de } A = Q, A^T A = I:$$

$$P = Q I Q^T = Q Q^T$$

Ters alma yok. Q kareyse $P = I$ (her şey zaten uzayda).

23.6 $\hat{\mathbf{x}} = Q^T \mathbf{b}$ — Normal Denklemler Çöker

$$Q^T Q \hat{\mathbf{x}} = Q^T \mathbf{b} \implies \hat{\mathbf{x}} = Q^T \mathbf{b}, \quad \hat{x}_i = \mathbf{q}_i^T \mathbf{b}$$

Her katsayı bağımsız bir dot product.

“The component along the i -th basis vector is just q_i transpose b — a dot product.” — Strang, 24:47

Builder Notu: Ortonormal bazda **koordinat = projeksiyon**. Bu, Fourier serisi, PCA skorları, wavelet katsayılarının ortak formülü.

23.7 Gram-Schmidt — Temel Fikir

İki bağımsız $\mathbf{a}, \mathbf{b} \rightarrow$ ortonormal $\mathbf{q}_1, \mathbf{q}_2$.

- $\mathbf{A} = \mathbf{a}$.
- $\mathbf{B} = \mathbf{b} - \frac{\mathbf{A}^T \mathbf{b}}{\mathbf{A}^T \mathbf{A}} \mathbf{A}$ (\mathbf{b} 'den \mathbf{A} yönündeki izdüşümü çıkar).

Kontrol: $\mathbf{A}^T \mathbf{B} = \mathbf{A}^T \mathbf{b} - (\mathbf{A}^T \mathbf{b}) = 0 \checkmark$.

Normalize: $\mathbf{q}_1 = \mathbf{A} / \|\mathbf{A}\|$, $\mathbf{q}_2 = \mathbf{B} / \|\mathbf{B}\|$.

Üç vektör için: $\mathbf{C} = \mathbf{c} - \frac{\mathbf{A}^T \mathbf{c}}{\mathbf{A}^T \mathbf{A}} \mathbf{A} - \frac{\mathbf{B}^T \mathbf{c}}{\mathbf{B}^T \mathbf{B}} \mathbf{B}$.

Sayısal örnek: $\mathbf{a} = (1, 1, 1)$, $\mathbf{b} = (1, 0, 2)$. $\mathbf{A}^T \mathbf{b} = 3$, $\mathbf{A}^T \mathbf{A} = 3$:

$$\mathbf{B} = (1, 0, 2) - (1, 1, 1) = (0, -1, 1)$$

$$\mathbf{q}_1 = \frac{1}{\sqrt{3}}(1, 1, 1), \quad \mathbf{q}_2 = \frac{1}{\sqrt{2}}(0, -1, 1)$$

Builder Notu: Saf Gram-Schmidt sayısal kararsız (kayan-nokta hatası birikir). Pratikte **modified Gram-Schmidt** veya **Householder**. ML'de: kararlı RNN, ortonormal baz öğrenme, seyrek kodlama.

23.8 $\mathbf{A} = \mathbf{QR}$ Ayrışımı

$$\mathbf{A} = \mathbf{QR}$$

- \mathbf{A} : orijinal bağımsız kolonlar.
- \mathbf{Q} : Gram-Schmidt'ten ortonormal kolonlar.
- $\mathbf{R} = Q^T \mathbf{A}$: üst üçgensel.

Neden üst üçgensel? $R_{ij} = \mathbf{q}_i^T \mathbf{a}_j$. $i > j$ için $\mathbf{q}_i, \mathbf{a}_j$ 'ye **dik** (Gram-Schmidt'in inşası): sonraki \mathbf{q} 'lar önceki \mathbf{a} 'lara dik \rightarrow alt üçgen sıfır.

\mathbf{A} ve \mathbf{Q} aynı kolon uzayına sahip; \mathbf{R} “geçiş matrisi”.

23 Ortogonal Matrisler ve Gram-Schmidt

Q =
[[-5.77350269e-01 -9.33980149e-17]
[-5.77350269e-01 -7.07106781e-01]
[-5.77350269e-01 7.07106781e-01]]

R (üst üçgensel) =
[[-1.73205081 -1.73205081]
[0. 1.41421356]]

Q^TQ = I? True
A = QR? True
QQ^T = A(A^TA)⁻¹A^T? True

23.9 QR ile Least Squares — Sayısal Üstünlük

$A^T A \hat{\mathbf{x}} = A^T \mathbf{b}$ 'ye $A = QR$ sok:

$$R^T Q^T Q R \hat{\mathbf{x}} = R^T Q^T \mathbf{b} \implies R \hat{\mathbf{x}} = Q^T \mathbf{b}$$

R üst üçgensel → geri yerine koymayla anında.

Avantajlar:

- $A^T A$ hiç oluşturulmaz — koşul sayısı $\kappa(A^T A) = \kappa(A)^2$. QR bundan kaçınır.
- **Kararlılık:** ortogonal Q hata biriktirmez.
- `np.linalg.lstsq` arka planda QR (veya daha kararlı SVD).

23.10 Bu Dersin Özeti

1. **Ortonormal** ($\mathbf{q}_i^T \mathbf{q}_j$).
2. $Q^T Q = I$.
3. **Ortogonal matris** ($Q^T = Q^{-1}$).
4. **Örnekler:** permütasyon, rotasyon, Hadamard.
5. $P = Q Q^T$.
6. $\hat{\mathbf{x}} = Q^T \mathbf{b}$.
7. **Gram-Schmidt** önceki yönlerin izdüşümünü çıkar.
8. **B formülü.**
9. $A = QR$.
10. **QR LS sayısal kararlı.**

! Tek bir cümle

Ortonormal kolonlar ($Q^T Q = I$) projeksiyonu ($P = Q Q^T$) ve least squares'i ($\hat{\mathbf{x}} = Q^T \mathbf{b}$) ters alması basitleştirir; **Gram-Schmidt** → **A = QR** modern regresyonun sayısal kararlı temeli.

23.11 Kontrol Soruları

i Soru 1: $Q = (1/\sqrt{2})(1 \ 1 \ / \ 1 \ -1)$ ortogonal mi?

$$Q^T Q = \frac{1}{2} \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} = I \checkmark$$

Ortogonal. $Q^T = Q^{-1}$.

i Soru 2: $\mathbf{a} = (1,0,1)$, $\mathbf{b} = (1,1,0) \rightarrow$ Gram-Schmidt.

$$\mathbf{A} = (1, 0, 1), \mathbf{q}_1 = \frac{1}{\sqrt{2}}(1, 0, 1).$$

$$\mathbf{A}^T \mathbf{b} = 1, \mathbf{A}^T \mathbf{A} = 2:$$

$$\mathbf{B} = (1, 1, 0) - \frac{1}{2}(1, 0, 1) = \left(\frac{1}{2}, 1, -\frac{1}{2}\right)$$

$$\|\mathbf{B}\| = \sqrt{3/2}, \mathbf{q}_2 = \frac{1}{\sqrt{3/2}}\left(\frac{1}{2}, 1, -\frac{1}{2}\right).$$

i Soru 3: Ortonormal Q ile P ve $\hat{\mathbf{x}}$?

$$P = Q(Q^T Q)^{-1} Q^T = Q Q^T \quad (Q^T Q = I \text{ sadeleşti}).$$

$$\hat{\mathbf{x}} = Q^T \mathbf{b}, \hat{x}_i = \mathbf{q}_i^T \mathbf{b}.$$

Ters alma yok; her katsayı bağımsız dot product.

i Soru 4: Regresyon neden QR kullanır?

1. Koşul sayısı: $\kappa(A^T A) = \kappa(A)^2$ — $A^T A$ çok daha kötü-koşullu.

2. QR: $R\hat{\mathbf{x}} = Q^T \mathbf{b}$, R üst üçgensel \rightarrow geri yerine koy, $A^T A$ hiç oluşturulmaz.

3. Kararlılık: ortonormal Q hata biriktirmez.

`np.linalg.lstsq` arka planda QR/SVD; “normal denklemleri elle ters alma” anti-pattern.

23.12 Egzersizler

Egzersiz 1. $\mathbf{a} = (1, 1, 0)$, $\mathbf{b} = (1, 0, 1)$, $\mathbf{c} = (0, 1, 1) \rightarrow$ ortonormal $\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3$.

Egzersiz 2. $Q = \frac{1}{3} \begin{pmatrix} 2 & -1 & 2 \\ 2 & 2 & -1 \\ -1 & 2 & 2 \end{pmatrix}$ ortogonal mi? Bir vektörü çarp, uzunluk korunsun.

Egzersiz 3. $\mathbf{a} = (3, 4) \rightarrow \mathbf{q}_1$, sonra $\mathbf{q}_2 \perp \mathbf{q}_1$ (2D’de iki yol).

Egzersiz 4. (Python) `np.linalg.qr` ile $A = QR$, projeksiyon karşılaştırması.

Egzersiz 5. İspatla: Ortogonal Q uzunluk ve dot product korur. (İpucu: $\|Q\mathbf{x}\|^2 = \mathbf{x}^T Q^T Q \mathbf{x}$.) Bu, SVD’nin U, V ’sinin neden “katı hareket” olduğunu açıklar.

23.13 Sonraki Ders İçin Hazırlık

Ders 18: Determinant ve Özellikleri

- Üç tanımlayıcı özellik ($\det I = 1$, satır takası işaret, lineerlik).
- Yedi türev sonuç.
- Tersinirlik testi: $\det = 0 \iff$ tekil.

⚠ Ders 18 öncesi

- Egzersiz 5 (uzunluk koruma).
- `np.linalg.qr` ile birkaç matris.

23.14 Anahtar Kavramlar (Cheat Sheet)

Kavram	Tanım	Strang'da
Ortonormal vektör $Q^T Q = I$	$\mathbf{q}_i^T \mathbf{q}_j = \delta_{ij}$ Ortonormal kolonların imzası	0m51 3m23
Ortogonal matris Rotasyon $P = QQ^T$	Kare Q , $Q^T = Q^{-1}$ ($\cos, -\sin; \sin, \cos$) Ters alma yok; Q kare \rightarrow $P = I$	6m59 9m53 17m50
$\hat{\mathbf{x}} = Q^T \mathbf{b}$ Gram-Schmidt	$\hat{x}_i = \mathbf{q}_i^T \mathbf{b}$ Önceki yönlerin izdüşümünü çıkar	23m10 25m39
B formülü $A = QR$	$\mathbf{b} - (\mathbf{A}^T \mathbf{b} / \mathbf{A}^T \mathbf{A}) \mathbf{A}$ R üst üçgensel	31m53 44m53
QR LS	$R\hat{\mathbf{x}} = Q^T \mathbf{b}$	46m45

23.15 ML Bağlantıları Özeti

💡 7 köprü

1. **Ortogonal $Q =$ rotasyon/yansıma** $\rightarrow \|\cdot\|$ korur; ortogonal başlatma.
2. $\hat{x}_i = \mathbf{q}_i^T \mathbf{b} \rightarrow$ Fourier, PCA, wavelet katsayıları.
3. $P = QQ^T \rightarrow$ SVD, PCA yapı taşı.
4. **Gram-Schmidt** \rightarrow Kararlı RNN, ortonormal baz; modified GS / Householder.
5. $A = QR \rightarrow$ `np.linalg.qr`, `lstsq` arka planı.
6. $\kappa(A^T A) = \kappa(A)^2 \rightarrow$ Normal denklemleri elle çözme anti-pattern.
7. **Hadamard / fast transforms** \rightarrow Performer attention, structured random projections.

! Tek bir Őey alıp gideceksen

Ortonormal kolonlar ($Q^T Q = I$) projeksiyon + LS'i ters almasız yapar; **Gram-Schmidt** $\rightarrow \mathbf{A} = \mathbf{QR}$
modern regresyonun kararlılık temeli.

24 Determinant ve Özellikleri

Üç özellikten her şey — hacim, tersinirlik, Jacobian

Bölüm bilgisi

- **Strang'ın videosu:** [YouTube — Lecture 18: Properties of Determinants](#) (≈48 dk)
- **OCW sayfası:** [MIT 18.06SC — Lecture 18](#)
- **Okuma süresi:** ≈40 dk

24.1 Bu Derste Ne Var?

Kare matrislere geçiş. **Determinant** = her kare matrise atanan bir sayı. Hedef: **A tersinir** \iff **det A \neq 0** + özdeşlere hazırlık (Ders 21).

3 tanımlayıcı özellik:

1. $\det(I) = 1$.
2. Satır takası \rightarrow işaret değişir.
3. Her satırda lineer.

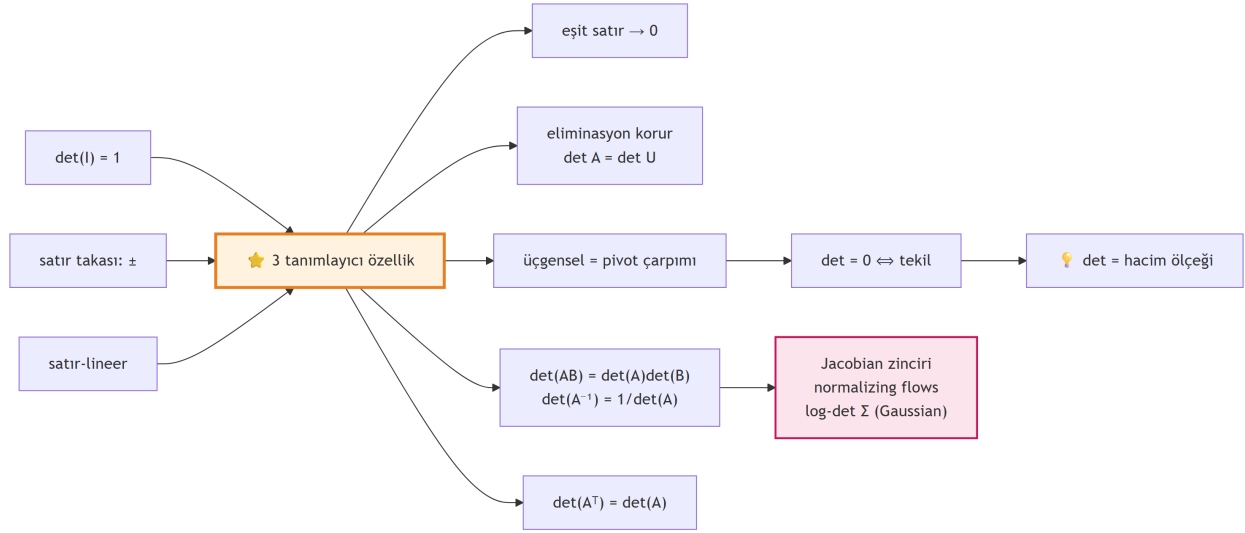
Bu üç özellikten tüm formüller ve sonuçlar türer.

“Determinant of A is zero exactly when A is singular.” — Strang, 28:28

Builder Notu — Determinant ML’ in Sessiz Kahramanı

- **det = Jacobian hacim ölçęęi** \rightarrow change of variables; **normalizing flows** log-det = olasılık düzeltmesi.
- **det(AB) = det(A)det(B)** \rightarrow zincir kuralı, flow katmanlarında log-det toplamı.
- **logdet Σ** \rightarrow Gaussian likelihood, entropi, Bayesian model seçimi.
- **det \approx 0** \rightarrow multicollinearity / kötü-koşullu / bilgi kaybı teşhisi.
- **det Q = \pm 1** \rightarrow ortogonal dönüşümler hacmi korur.

24 Determinant ve Özellikleri



Şekil 24.1: 3 özellik → 10 sonuç. det = hacim, tersinirlik testi, Jacobian, flow modellerinin temeli.

24.2 Özellik 1, 2 — $\det(I) = 1$, Satır Takası

Permütasyon matrisi: $\det(P) = \pm 1$ (takas paritesine göre).

“The determinant of a permutation is one or minus one, depending whether the number of exchanges was even or odd.” — Strang, 5:03

24.3 Özellik 3 — Satır-Lineerlik

$$3A: \begin{vmatrix} ta & tb \\ c & d \end{vmatrix} = t \begin{vmatrix} a & b \\ c & d \end{vmatrix}.$$

$$3B: \begin{vmatrix} a + a' & b + b' \\ c & d \end{vmatrix} = \begin{vmatrix} a & b \\ c & d \end{vmatrix} + \begin{vmatrix} a' & b' \\ c & d \end{vmatrix}.$$

UYARI: $\det(A + B) \neq \det A + \det B$! Lineerlik sadece **tek satırda**, diğerleri sabitken.

24.4 Özellik 4 — Eşit Satır → det = 0

İki eşit satırı takas et → işaret değişmeli ($\det \rightarrow -\det$) ama matris aynı kaldı ($\det \rightarrow \det$):

$$\det = -\det \implies \det = 0$$

Saf mantık ispatı, $n \times n$ için geçerli.

24.5 Özellik 5, 6 — Eliminasyon ve Sıfır Satır

5: Eliminasyon (bir satırdan diğerinin katımı çıkar) det'i **korur**. $\rightarrow \det A = \det U$.

6: Sıfır satır $\rightarrow \det = 0$ (Özellik 3A, $t = 0$).

24.6 Özellik 7 — Üçgensel = Pivot Çarpımı

$$\det(U) = d_1 \cdot d_2 \cdots d_n$$

Pratik hesap: eliminasyon \rightarrow pivotları çarp (satır takası varsa \pm). 100×100 için bu kullanılır, ad - bc formülü değil.

Builder Notu: $\log |\det A| = \sum \log |d_i|$ — Gaussian likelihood, normalizing flow her katman log-det, Bayesian model seçimi. **Cholesky/LU** pivotları verir.

24.7 Özellik 8 — $\det = 0 \iff$ Singular

- **Singular:** eliminasyonda sıfır satır \rightarrow Özellik 6 $\rightarrow \det = 0$.
- **Tersinir:** tam pivot kümesi \rightarrow Özellik 7 $\rightarrow \det \neq 0$.

$$\det A = \pm(d_1 d_2 \cdots d_n)$$

2×2 doğrulama: $\det = a(d - cb/a) = ad - bc \checkmark$.

24.8 Özellik 9 — $\det(AB) = \det(A) \cdot \det(B)$

Sonuçlar:

- $\det(A^{-1}) = 1/\det(A)$.
- $\det(A^2) = \det(A)^2$.
- $\det(cA) = c^n \det(A)$ (her satırdan bir c).

$$\det(A) = 2.0000$$

$$\det(2A) = 16.0000 = 2^3 \cdot \det(A) = 16.0000$$

$$\det(A^T) = 2.0000$$

$$\det(A^{-1}) = 0.5000 = 1/\det(A) = 0.5000$$

$$\det(AB) = 0.0000 = \det(A) \cdot \det(B) = 0.0000$$

Builder Notu: $\det(cA) = c^n \det(A) = \text{hacim}$ (3D'de $2 \times \rightarrow 8 \times$). $\det(AB) = \det(A) \det(B) = \text{Jacobian}$ zincir kuralı. Normalizing flow'da log-det'ler toplanır.

24.9 Özellik 10 — $\det(\mathbf{A}^T) = \det(\mathbf{A})$

İspat: $A = LU \rightarrow A^T = U^T L^T$; L, L^T köşegen 1, U, U^T aynı köşegen.

Büyük sonuç: Satır kuralları **kolonlar için de** geçerli (sıfır kolon $\rightarrow 0$, kolon takası $\rightarrow \pm$, vs.).

24.10 Determinant = Hacim

- **2×2:** paralelkenar alanı.
- **3×3:** paralelyüz hacmi.
- **n×n:** n-boyutlu kutu hacmi.

$\det = 0 \rightarrow$ yassı kutu (sıfır hacim) \rightarrow bağımlı.

$\det(2A) = 2^n \det(A) \rightarrow$ her kenar 2 kat \rightarrow hacim 2^n kat.

Builder Notu: Change of variables: $p_Y(\mathbf{y}) = p_X(\mathbf{x}) |\det J|$. Normalizing flow'lar Jacobian'ı üçgensel ($\det =$ köşegen çarpımı, ucuz) tutar — **RealNVP, Glow, autoregressive flows**.

24.11 Bu Dersin Özeti

1. Determinant kare matrise atanan sayı; **tersinir** \iff **det $\neq 0$** .
2. **3 özellik:** $\det I = 1$, satır takası \pm , satır-lineerlik.
3. **Eşit satır $\rightarrow 0$.**
4. **Eliminasyon korur.**
5. **Üçgensel = pivot çarpımı.**
6. **det = 0 \iff singular.**
7. **det(AB) = det(A)det(B).**
8. **det(A^T) = det(A).**
9. **det = hacim ölçęi.**

! Tek bir cümle

Determinant üç özellikle ($\det I = 1$, satır takası \pm , satır-lineerlik) tanımlanır; **A tersinir \iff det A $\neq 0$** , $\det A =$ pivot çarpımı, $\det(AB) = \det(A) \det(B)$ — ve geometrik olarak **hacim ölçęi** (Jacobian, normalizing flows).

24.12 Kontrol Soruları

i Soru 1: $A = ((1,2,0),(2,5,1),(0,1,3))$ determinanı.

$r_2 - 2r_1 = (0, 1, 1)$. $r_3 - r_2(\text{yeni}) = (0, 0, 2)$.
 U : pivotlar 1, 1, 2. **det = 2**.

i Soru 2: $\det A = 3$ (4×4). $\det(2A)$, $\det(A^{-1})$, $\det(A^T)$, $\det(A^2)$?

- $\det(2A) = 2^4 \cdot 3 = 48$.
- $\det(A^{-1}) = 1/3$.
- $\det(A^T) = 3$.
- $\det(A^2) = 9$.

Dikkat: $\det(2A) = 48$, $2 \cdot 3 = 6$ **değil**.

i Soru 3: $A = ((1,2,3),(4,5,6),(7,8,9))$ singular mı?

$r_1 + r_3 = (8, 10, 12) = 2r_2 \rightarrow$ bağımlı.
 Eliminasyon sıfır satır verir \rightarrow **det = 0** \rightarrow singular.

i Soru 4: Normalizing flow'larda Jacobian determinanı niye önemli?

Change of variables: $p_Y(\mathbf{y}) = p_X(\mathbf{x}) \cdot |\det J|$ (hacim koruması).

Normalizing flow: Basit dağılım (Gaussian) \rightarrow karmaşık dağılım, tersinir zincir.

Her adımın $\log |\det J|$ log-olabilirliğe eklenir; $\det(\prod) = \prod \det$ sayesinde **toplama** olur.

Flow'lar Jacobian'ı **üçgensel** tutar ($\det =$ köşegen çarpımı, ucuz). $\det \rightarrow 0 =$ bilgi kaybı yönü.

RealNVP, Glow, autoregressive flows bu temele dayanır.

24.13 Egzersizler

Egzersiz 1. $A = \begin{pmatrix} 2 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 2 \end{pmatrix}$ determinanı.

Egzersiz 2. $\det A = 5$, $\det B = 2$ (3×3): $\det(AB)$, $\det(2A)$, $\det(A^{-1}B)$, $\det(A^T A)$.

Egzersiz 3. İki eşit kolon \rightarrow det? (Özellik 10.)

Egzersiz 4. (Python) Özellikleri doğrula.

Egzersiz 5. İspatla: Ortogonal Q için $\det Q = \pm 1$. (İpucu: $Q^T Q = I + \det Q^T = \det Q$.) Rotasyon $+1$, yansıma -1 ; hacim korunur.

24.14 Sonraki Ders İçin Hazırlık

Ders 19: Determinant Formülleri ve Kofaktörler

- Büyük formül ($n!$ terim, permütasyonlar).
- Kofaktör açılımı.
- Ters matris formülünün habercisi (Ders 20).

⚠ Ders 19 öncesi

- Egzersiz 5 ($\det Q = \pm 1$).
- np.linalg.det ile özellikleri doğrula.

24.15 Anahtar Kavramlar (Cheat Sheet)

Özellik	İfade	Strang'da
1. $\det I = 1$	Birim	2m33
2. Satır takası	\pm	3m27
3. Satır-lineer	Sadece bir satır	7m08
4. Eşit satır $\rightarrow 0$	—	11m43
5. Eliminasyon korur	$\det A = \det U$	14m34
6. Sıfır satır $\rightarrow 0$	—	19m07
7. Üçgensel	Pivot çarpımı	22m42
8. $\det = 0 \iff$ singular	—	28m28
9. $\det(AB)$	$= \det(A) \det(B)$, $\det(cA) = c^n \det A$	34m01
10. $\det(A^T) = \det(A)$	Satır \rightarrow kolon	41m44

24.16 ML Bağlantıları Özeti

💡 7 köprü

1. **$\det =$ hacim / Jacobian** \rightarrow Change of variables.
2. **$\det(AB) = \det(A)\det(B)$** \rightarrow Flow log-det toplamı.
3. **$\det = 0 \iff$ tekil** \rightarrow Multicollinearity / kötü-koşullu.
4. **$\log|\det \Sigma|$** \rightarrow Gaussian likelihood, entropi.
5. **$\det(cA) = c^n \det A$** \rightarrow Hacim üstel.
6. **$\det Q = \pm 1$** \rightarrow Rotasyon/yansıma hacim korur.
7. **Permütasyon paritesi** \rightarrow Antisimetrik fonksiyonlar, DPP.

! Tek bir Őey alıp gideceksen

3 özellikten tüm determinant teorisi türer; **det = 0** \Leftrightarrow **singular**, det = pivot çarpımı, $\det(AB) = \det(A) \det(B)$; geometrik olarak **hacim ölçęi** (Jacobian, normalizing flows).

25 Determinant Formülleri ve Kofaktörler

Üç yüz — pivot, büyük formül, kofaktör

Bölüm bilgisi

- **Strang'ın videosu:** [YouTube — Lecture 19: Determinant Formulas and Cofactors](#) (≈53 dk)
- **OCW sayfası:** [MIT 18.06SC — Lecture 19](#)
- **Okuma süresi:** ≈40 dk

25.1 Bu Derste Ne Var?

Ders 18'in özelliklerinden açık formüller.

1. **Büyük formül:** $\det A = \sum \pm a_{1\alpha} a_{2\beta} \dots, n!$ **terim.**
2. **Permütasyon işareti** (parite).
3. **Kofaktör açılımı:** $n \times n \rightarrow (n-1) \times (n-1)$ recursive.
4. **Üç formülün karşılaştırması;** tridiagonal recursion.

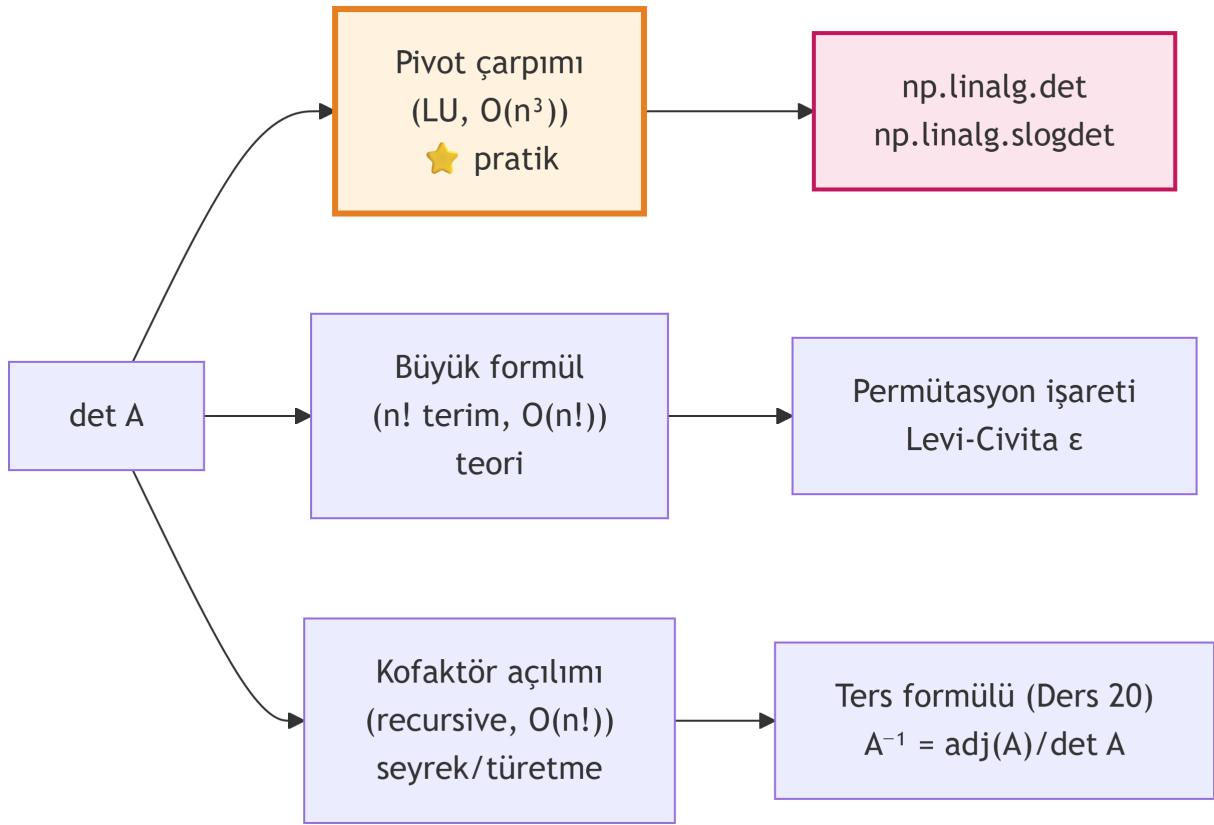
“The survivors have one entry from each row and each column.” — Strang, 9:10

Builder Notu — ML'de Determinant

- **Pratik:** `np.linalg.det` her zaman LU. Büyük matriste **slogdet** (kararlı log-det).
- **Determinantal point processes (DPP):** çeşitlilik (diversity); determinant kolay olduğu için verimli.
- **Permanent** (işaretsiz det) **#P-tam** → boson sampling, eşleştirme sayma.
- **Permütasyon paritesi = Levi-Civita** → antisimetrik tensörler, **fermiyonik sinir ağları (Fermi-Net)**.
- **Recursion (tridiagonal)** → HMM forward-backward, DP prototipi.

25.2 2×2'yi Özelliklerden Türetme

İlk satır $(a, b) = (a, 0) + (0, b)$:



Şekil 25.1: Determinantın üç yüzü ve maliyetleri.

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix} + bc \begin{vmatrix} 0 & 1 \\ 1 & 0 \end{vmatrix} = ad - bc$$

(Sıfır kolonlu terimler 0; köşegen 1; anti-köşegen takasla -1.)

25.3 Survivor'lar ve Büyük Formül

3×3'te $3^3 = 27$ terim, ama çoğu sıfır (tekrarlı kolon). **Survivor** = her satır + kolondan tam bir giriş = **permütasyon** = $n!$ tane.

$$\det A = \sum_{\sigma \in S_n} (\text{sgn } \sigma) a_{1,\sigma(1)} \cdots a_{n,\sigma(n)}$$

Boyut	Terim sayısı
2	2
3	6
4	24
10	3.6M

25.4 Permütasyon İşareti

Permütasyon birim sıraya getirmek için gereken **takas sayısının paritesi**.

Örnek (4×4 anti-köşegen): kolon sırası (4, 3, 2, 1). $1 \leftrightarrow 4, 2 \leftrightarrow 3 \rightarrow 2$ takas $\rightarrow +$.

3×3 anti-köşegen (3, 2, 1): 1 takas $\rightarrow -$. (Boyuta bağlı; “ana köşegen +, anti-köşegen -” kuralı sadece 3×3 için.)

Builder Notu: Parite = **Levi-Civita sembolü** $\varepsilon_{ijk\dots}$. Antisimetrik tensörler, çapraz çarpım, fermiyonik ağlar bununla yazılır.

25.5 Çok Sıfır = Az Survivor

Strang'ın 4×4 örneği (köşelerde 1'ler) — 24 terimin çoğu sıfır, +1 ve -1 birbirini götürür $\rightarrow \det = 0$ (eşit satır).

Seyrek matrislerde büyük formül elle hızlı.

25.6 Kofaktör Açılımı

Büyük formülü ilk satırın girişlerine göre grupta:

$$\det A = a_{11}C_{11} + a_{12}C_{12} + \dots + a_{1n}C_{1n}$$

$C_{ij} = a_{ij}$ 'nin **kofaktörü** = $(n - 1) \times (n - 1)$ minörün determinanı \times işaret:

$$C_{ij} = (-1)^{i+j} \det(M_{ij})$$

Checkerboard işaret: $(i + j)$ çift $\rightarrow +$, tek $\rightarrow -$.

3×3 ilk satır kofaktörü:

$$\det A = a_{11}(a_{22}a_{33} - a_{23}a_{32}) - a_{12}(\dots) + a_{13}(\dots)$$

2×2 doğrulama: $\det = a \cdot d + b \cdot (-c) = ad - bc \checkmark$.

İpucu: En çok sıfır içeren satır/kolon boyunca aç.

Builder Notu: Recursive \rightarrow dinamik programlama prototipi. Saf hâli $O(n!)$, pratik değil; ama **seyrek matriste** ve **simgesel türetmede** (Ders 20 ters formülü) kullanışlı.

25.7 Üç Formülün Karşılaştırması

Yöntem	Maliyet	Ne için?
Pivot (LU)	$O(n^3)$	Pratik hesap
Büyük formül	$O(n!)$	Teori, ispat
Kofaktör	$O(n!)$	Seyrek, türetme

$\det(A)$ (LU) = 3.0000
 sign = 1.0, $|\det| = \exp(1.0986) = 3.0000$

25.8 Tridiagonal Örnek — Recursion ve Periyot 6

Tüm girişleri 1 olan tridiagonal A_n . İlk satır kofaktörüyle:

$$\det(A_n) = \det(A_{n-1}) - \det(A_{n-2})$$

$$\det(A_1) = 1, \det(A_2) = 0:$$

$$1, 0, -1, -1, 0, 1, 1, 0, -1, \dots$$

Periyot 6!

tridiagonal det: [1, 0, -1, -1, 0, 1, 1, 0, -1, -1, 0, 1]

Builder Notu: Linear recurrence (Fibonacci akrabası) = **DP prototipi**. Tridiagonal matrisler 1-D Laplacian, spline, HMM forward-backward; tridiagonal sistemler $O(n)$ (Thomas).

25.9 Bu Dersin Özeti

1. 2×2 lineerlikle türetildi.
2. **Survivor** = permütasyon.
3. **Büyük formül:** $n!$ terim.
4. **İşaret** = parite.
5. **Seyreklik** → az survivor.
6. **Kofaktör** = $(n-1)$ recursive indirgeme.
7. **İşaret** $(-1)^{i+j}$.
8. **Formül:** $\det A = \sum a_{ij} C_{ij}$.
9. **Üç formül.**
10. **Tridiagonal recursion**, periyot 6.

! Tek bir cümle

Determinantın üç yüzü: **pivot çarpımı** ($O(n^3)$, pratik), **büyük formül** ($n!$ terim, teori), **kofaktör** (recursive, seyrek/türetme); pratikte hep LU (np.linalg.det / slogdet).

25.10 Kontrol Soruları

i Soru 1: $A = ((2,0,1),(1,3,2),(0,1,1))$ — ilk satır kofaktörü.

$$\det = 2 \cdot C_{11} + 0 + 1 \cdot C_{13}.$$

$$C_{11} = + \det \begin{pmatrix} 3 & 2 \\ 1 & 1 \end{pmatrix} = 1. \quad C_{13} = + \det \begin{pmatrix} 1 & 3 \\ 0 & 1 \end{pmatrix} = 1.$$

$$\det = 2 + 1 = 3.$$

i Soru 2: 4×4 ve 3×3 anti-köşegen işareti.

- **4×4** $(4, 3, 2, 1)$: $1 \leftrightarrow 4, 2 \leftrightarrow 3 \rightarrow 2$ takas $\rightarrow +$.
- **3×3** $(3, 2, 1)$: $1 \leftrightarrow 3 \rightarrow 1$ takas $\rightarrow -$.

Boyuta bağlı; parite genel.

i Soru 3: 4×4 , sadece köşegen $+$ $(1,4), (4,1)$ — kaç survivor?

1. Köşegen $1, 2, 3, 4 \rightarrow$ bir survivor.
2. $(1, 4)$ ve $(4, 1) + a_{22}, a_{33} \rightarrow$ permütasyon $(4, 2, 3, 1) \rightarrow$ ikinci survivor.

2 survivor. $1 \leftrightarrow 4$ tek takas \rightarrow işaret $-$. $\det = a_{11}a_{22}a_{33}a_{44} - a_{14}a_{22}a_{33}a_{41}$.

i Soru 4: Permanent neden zor? Recursion ML'de nerede?

Det vs permanent: Aynı $n!$ terim, fark işaretler.

- **Det:** işaretler eliminasyonu kullanılabilir kılar $\rightarrow O(n^3)$.
- **Permanent:** işaret yok \rightarrow **#P-tam** (boson sampling, bipartite eşleştirme).

ML'de:

- **DPP:** çeşitlilik modelleme (det kolay \rightarrow verimli).
- **Recursion (Tridiagonal):** HMM forward-backward, RNN, DP.
- **Pratik:** slogdet her zaman LU.

25.11 Egzersizler

Egzersiz 1. $A = \begin{pmatrix} 1 & 0 & 0 & 2 \\ 0 & 3 & 0 & 0 \\ 4 & 0 & 5 & 0 \\ 0 & 0 & 0 & 6 \end{pmatrix} \rightarrow$ en çok sıfır içeren satırla kofaktör.

Egzersiz 2. 3×3 büyük formül 6 terimi; permütasyon $(3, 1, 2)$ işareti.

Egzersiz 3. Köşegen 2, komşular 1 tridiagonal recursion: $\det(A_n) = ?$

Egzersiz 4. (Python) Üç yöntem karşılaştırma + slogdet.

Egzersiz 5. *İspatla:* Kofaktör formülü = büyük formül. Ders 20 (Cramer) temeli.

25.12 Sonraki Ders İçin Hazırlık

Ders 20: Cramer Kuralı, Ters Matris ve Hacim

- $A^{-1} = \frac{1}{\det A} C^T$ (adjugate formülü).
- **Cramer:** $A\mathbf{x} = \mathbf{b} \rightarrow x_i = \det(B_i) / \det(A)$.
- **Hacim:** det = paralelyüz hacmi.

⚠ Ders 20 öncesi

- Egzersiz 5 (formül eşdeğerliği).
- `np.linalg.slogdet` ile kararlı log-det dene.

25.13 Anahtar Kavramlar (Cheat Sheet)

Kavram	Tanım	Strang'da
2×2 türetme	Lineerlik $\rightarrow ad - bc$	2m04
Survivor	Permütasyon	6m13
Büyük formül	$\sum \pm a_{1\alpha} \dots, n!$	16m39
Parite	Takas sayısı (\pm)	19m50
Seyreklik	Az survivor	22m36
Kofaktör fikri	$(n - 1)$ recursive	28m53
$C_{ij} = (-1)^{i+j} \det M_{ij}$	Checkerboard	36m09
Kofaktör formülü	$\det A = \sum a_{ij} C_{ij}$	39m49
Üç formül	Pivot / büyük / kofaktör	43m01
Tridiagonal periyot 6	$A_n = A_{n-1} - A_{n-2}$	52m48

25.14 ML Bağlantıları Özeti

💡 7 köprü

1. **Det = işaretli eşleştirmeler** \rightarrow Permanent zor; DPP kolay.
2. **Kofaktör recursive** \rightarrow DP prototipi.
3. **Pratik det = LU** \rightarrow det/slogdet.
4. **Parite = Levi-Civita** \rightarrow Antisimetrik tensör, FermiNet.
5. **DPP** \rightarrow Çeşitlilik modelleme.
6. **Tridiagonal recursion** \rightarrow HMM, spline, 1-D Laplacian.
7. **slogdet** \rightarrow Gaussian likelihood, normalizing flows kararlı log-det.

! Tek bir şey alıp gideceksen

Üç eşdeğer formül: pivot ($O(n^3)$, pratik), büyük formül ($n!$ terim), kofaktör (recursive). Pratikte hep LU; **slogdet** büyük matriste kararlı log-det.

26 Cramer Kuralı, Ters Matris ve Hacim

Üç uygulama — kapalı-form vs hacim sezgisi

i Bölüm bilgisi

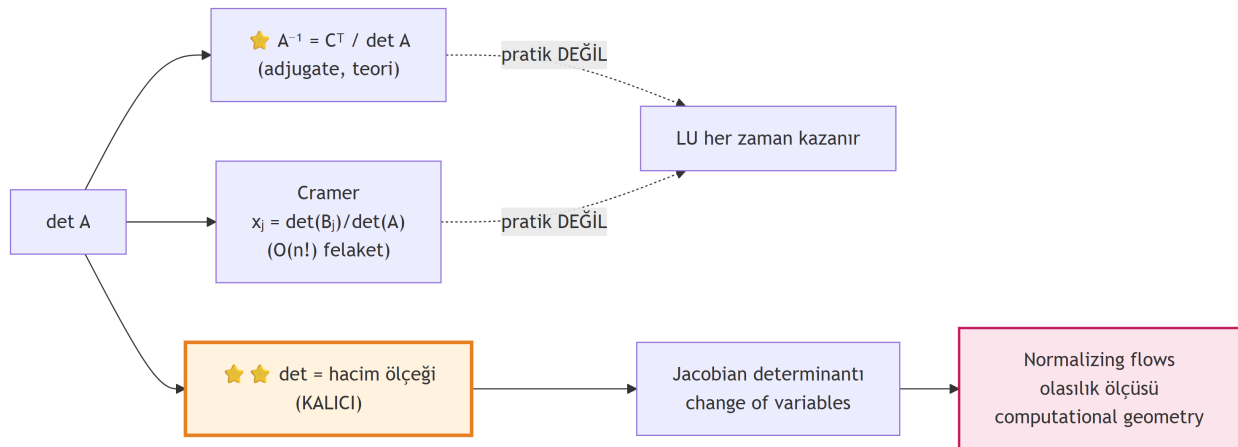
- **Strang'ın videosu:** [YouTube — Lecture 20: Cramer's Rule, Inverse Matrix, and Volume](#) (≈50 dk)
- **OCW sayfası:** [MIT 18.06SC — Lecture 20](#)
- **Okuma süresi:** ≈40 dk

26.1 Bu Derste Ne Var?

Determinantın üç uygulaması:

1. **Ters matris formülü:** $A^{-1} = \frac{1}{\det A} C^T$ (adjugate).
2. **Cramer kuralı:** $x_j = \det(B_j) / \det(A)$ — güzel ama pratikte felaket.
3. **det = hacim:** kalıcı fikir, Jacobian'ın temeli.

“The determinant actually equals the volume of something.” — Strang, 28:22



Şekil 26.1: Determinantın üç uygulaması — formüller (teorik) vs hacim (kalıcı).

 Builder Notu — Güzel Formül ≠ Pratik Algoritma

- **A^{-1} formülü ve Cramer** teorik şık, sayısal felaket — eliminasyon (LU) her zaman kazanır.
- **det = hacim** → Jacobian değişken değişimi, normalizing flow yoğunluk düzeltmesi.
- **det Q = ±1** → ortogonal hacim korur, kararlı flow/RNN katmanları.
- **Koordinatlardan alan/hacim** → computational geometry, point cloud (PointNet), mesh.

26.2 Ters Matris Formülü

$$A^{-1} = \frac{1}{\det A} C^T$$

C = kofaktör matrisi, C^T = adjugate. 2×2 'deki tanıdık formülün ($\frac{1}{ad-bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$) genellemesi.

Neden çalışır: $A \cdot C^T = \det(A) \cdot I$. Köşegen girişler kofaktör formülü ile $\det A$:

$$a_{i1}C_{i1} + \dots + a_{in}C_{in} = \det A$$

Köşegen-dışı sıfır: $i \neq j$ ise çarpım, “**iki eşit satırlı bozulmuş matrisin**” determinantına eşittir → 0 (Ders 18, Özellik 4).

“It’s as if I’m taking the determinant of a screwed up matrix with two identical rows.” — Strang, 17:09

Builder Notu: Adjugate **sembolik/parametrik** durumlarda değerli (implicit function theorem, hiperparametre gradyanları). Hesap için her zaman LU.

26.3 Cramer Kuralı

$$x_j = \frac{\det(B_j)}{\det A}$$

$B_j = A$ 'nın j -inci kolonu \mathbf{b} ile değiştirilmiş.

Türetme: $\mathbf{x} = A^{-1}\mathbf{b} = \frac{1}{\det A} C^T \mathbf{b}$. j -inci bileşen tam olarak $\det(B_j)/\det(A)$ (kolon kofaktör açılımı).

Pratik değil: $n + 1$ determinant. $n = 10$ 'da bile felaket. **Eliminasyon** $O(n^3)$ **kazanır**.

“Cramer’s Rule is a disastrous way to go, because to compute these determinants, it takes approximately forever.” — Strang, 27:05

Ders: Kapalı-form **algebra** içindir (anlama, türetme); **algoritma** (eliminasyon) hesap içindir.

26.4 Determinant = Hacim

$|\det A|$ = satırların geldiği **kutunun (paralelyüz) işaretli hacmi**.

- **2×2:** paralelkenar alanı.
- **3×3:** paralelyüz hacmi.
- **n×n:** n-boyutlu kutu hacmi.

İşaret = **orientasyon** (sağ-el/sol-el).

Birim küp: $A = I \rightarrow$ kutu birim küp, hacim $1 = \det(I) \checkmark$.

Ortogonal Q : kolonlar ortonormal \rightarrow kutu döndürülmüş birim küp, hacim 1. $Q^T Q = I \rightarrow (\det Q)^2 = 1 \rightarrow \det Q = \pm 1$. Rotasyon +1 (orientasyon korunur), yansıma -1 .

Builder Notu: Jacobian determinanti = hacim ölççeği = olasılık ölçüsü düzeltmesi. Normalizing flow'larda her katman $\log |\det J|$ olabirliğe eklenir; ortogonal katmanlar $\log |\det| = 0 \rightarrow$ ucuz ve kararlı.

26.5 Hacim = Determinant İspatı

Hacim, determinanti tanımlayan 3 özelliği sağlar \rightarrow hacim = determinant.

- **Özellik 1:** birim küp hacmi 1 \checkmark .
- **Özellik 2:** kenar takası \rightarrow orientasyon (işaret) değişir, hacim aynı \checkmark .
- **Özellik 3A:** kenar $\times t \rightarrow$ hacim $\times t \checkmark$.
- **Özellik 3B:** lineer ayrışım (shear ile gösterilir) \checkmark .

Aynı 3 özellik \rightarrow tek fonksiyon \rightarrow **hacim = determinant**.

Builder Notu: Bu “aksiyom karakterizasyonu” matematiğin güçlü tekniği. ML’de “değişmez (invariant) çekirdekleri” tanımlamak için kullanılır.

26.6 Alan Formülü — Paralelkenar ve Üçgen

2-D’de: kenarlar $(a, b), (c, d)$ paralelkenar:

$$\text{Alan} = |ad - bc|$$

Üçgen (orijinde başlıyorsa): yarısı $= \frac{1}{2}|ad - bc|$.

Genel üçgen (köşeler (x_i, y_i)):

$$\text{Alan} = \frac{1}{2} \left| \det \begin{pmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{pmatrix} \right|$$

Kareköksüz, sadece koordinatlardan — geometriden çok daha temiz.

26 Cramer Kuralı, Ters Matris ve Hacim

Paralelkenar alanı: 11.000000000000002

Üçgen (0,0), (3,1), (1,4): 5.500000000000001

Üçgen (1,1), (4,2), (2,5): 5.500000000000001

Builder Notu: **Computational geometry** ve **graphics**'in temel: üçgen mesh alanı, shoelace formülü (poligon), barycentric, point-in-triangle (işaret kontrolü). **PointNet**, **mesh CNN** bu nicelikleri özellik olarak kullanır. **3D tetrahedron hacmi** sonlu eleman mesh kalitesi $\rightarrow \det \approx 0 =$ çarpık eleman.

26.7 Bu Dersin Özeti

1. $A^{-1} = C^T / \det A$.
2. **Köşegen-dışı sıfır** ("bozulmuş" matris ispatı).
3. **Cramer:** $x_j = \det(B_j) / \det(A)$.
4. **Cramer pratik değil** \rightarrow eliminasyon.
5. **det = hacim**.
6. $\det Q = \pm 1$.
7. **Hacim = det** aksiyom ispatı.
8. **Alan** = $|ad - bc|$.
9. **Genel üçgen** 3×3 det.

! Tek bir cümle

$A^{-1} = C^T / \det A$ ve **Cramer** zarif kapalı-formlar, ama pratikte hep LU. Asıl kalıcı fikir: **det = hacim ölçüğü** (Jacobian, normalizing flows, computational geometry).

26.8 Kontrol Soruları

i Soru 1: $A = \begin{pmatrix} 2 & 1 \\ 1 & 3 \end{pmatrix} \rightarrow A^{-1}$ kofaktör formülüyle.

$$\det = 5. C = \begin{pmatrix} 3 & -1 \\ -1 & 2 \end{pmatrix} = C^T \text{ (simetrik)}. A^{-1} = \frac{1}{5} \begin{pmatrix} 3 & -1 \\ -1 & 2 \end{pmatrix}.$$

i Soru 2: Cramer ile $2x_1 + x_2 = 3$, $x_1 + 3x_2 = 5$.

$$\det A = 5.$$
$$x_1 = \frac{1}{5} \begin{vmatrix} 3 & 1 \\ 5 & 3 \end{vmatrix} = \frac{9-5}{5} = \frac{4}{5}.$$
$$x_2 = \frac{1}{5} \begin{vmatrix} 2 & 3 \\ 1 & 5 \end{vmatrix} = \frac{10-3}{5} = \frac{7}{5}.$$

i Soru 3: Köşeler (0,0), (3,1), (1,4) → üçgen alanı.

Orijinde → iki kenar (3, 1), (1, 4).
 $|ad - bc| = |12 - 1| = 11$. Üçgen = **5.5**.

i Soru 4: Jacobian = hacim ölçeği nasıl ML'de? Ortogonal katman avantajı? Cramer neden ML'de yok?

Jacobian: $\mathbf{y} = f(\mathbf{x})$ → küçük hacim ögesi $|\det J|$ kat ölçeklenir → yoğunluk $|\det J|^{-1}$ ile düzeltilir.

Normalizing flow: $\log |\det J|$ log-olabilirliğe eklenir; toplam zincir = $\sum \log |\det |$.

Ortogonal katman ($\det = \pm 1$): $\log |\det| = 0$, hacim korunur, $\|Q\mathbf{x}\| = \|\mathbf{x}\|$ → gradyan kararlı.

Cramer pratik değil: $n + 1$ determinant; LU $O(n^3)$ ile aynı sonuç. `np.linalg.solve` her zaman LU.

Özet: hacim fikri derin (Jacobian, ortogonalite); Cramer/adjugate sadece teorik.

26.9 Egzersizler

Egzersiz 1. $A = \begin{pmatrix} 1 & 2 & 3 & 4 \end{pmatrix} \rightarrow C, C^T, A^{-1}$. `np.linalg.inv` ile karşılaştır.

Egzersiz 2. Cramer ile: $x + y + z = 6, 2y + z = 4, z = 2$.

Egzersiz 3. Köşeler (1, 1), (4, 2), (2, 5) → üçgen alanı (3×3 det).

Egzersiz 4. (Python) solve vs Cramer karşılaştırma.

Egzersiz 5. *İspatla:* $\det(AB) = \det A \det B$ + hacim yorumu → bileşke hacim ölçekleri çarpılır. Ders 21'de özdeğer çarpımı = det.

26.10 Sonraki Ders İçin Hazırlık

Ders 21: Özdeğerler ve Özvektörler

- **Özvektör:** $A\mathbf{x} = \lambda\mathbf{x}$.
- **Karakteristik denklem:** $\det(A - \lambda I) = 0$.
- $\sum \lambda = \text{trace}$, $\prod \lambda = \det$.

⚠ Ders 21 öncesi

- Egzersiz 5 (det = özdeğer çarpımı habercisi).
- solve vs Cramer'i Python'da gör.

26.11 Anahtar Kavramlar (Cheat Sheet)

Kavram	Tanım	Strang'da
$A^{-1} = C^T / \det A$	Adjugate	3m26

Kavram	Tanım	Strang'da
$AC^T = (\det A)I$	Köşegende kofaktör	7m58
Köşegen-dışı 0	“Bozulmuş” matris det	17m09
Cramer	$x_j = \det(B_j) / \det(A)$	22m14
Cramer pratik değil	$n + 1$ det; LU kazanır	27m05
det = hacim	Kutu hacmi	28m22
det Q = ±1	Ortogonal = döndürülmüş küp	35m38
Hacim = det ispatı	3 özellik	33m42
Alan = $\ ad - bc\$	Paralelkenar; üçgen yarısı	46m03
Koordinatlardan alan	3×3 det	50m34

26.12 ML Bağlantıları Özeti

💡 7 köprü

1. $A^{-1} = \text{adj}/\text{det}$ → Sembolik/implicit theorem; hesap için LU.
2. **Cramer** → Pratik değil; kavramsal kapalı-form.
3. **det = hacim** → Jacobian, normalizing flow yoğunluk düzeltmesi.
4. **det Q = ±1** → Ortogonal flow/RNN katmanları; log-det = 0.
5. **Koordinatlardan alan/hacim** → PointNet, mesh CNN.
6. **Aksiyom karakterizasyonu** → Değişmez çekirdekler.
7. **det ≈ 0** → Kötü-koşullu / dejenere / bilgi kaybı teşhisi.

❗ Tek bir şey alıp gideceksen

A^{-1} ve **Cramer** zarif ama pratikte hep LU. **Asıl kalıcı fikir: det = hacim ölçeği** — Jacobian, olasılık ölçüsü, ortogonalite, computational geometry.

27 Özdeğerler ve Özvektörler

Kursun zirvesi — $Ax = \lambda x$, doğal eksenler

i Bölüm bilgisi

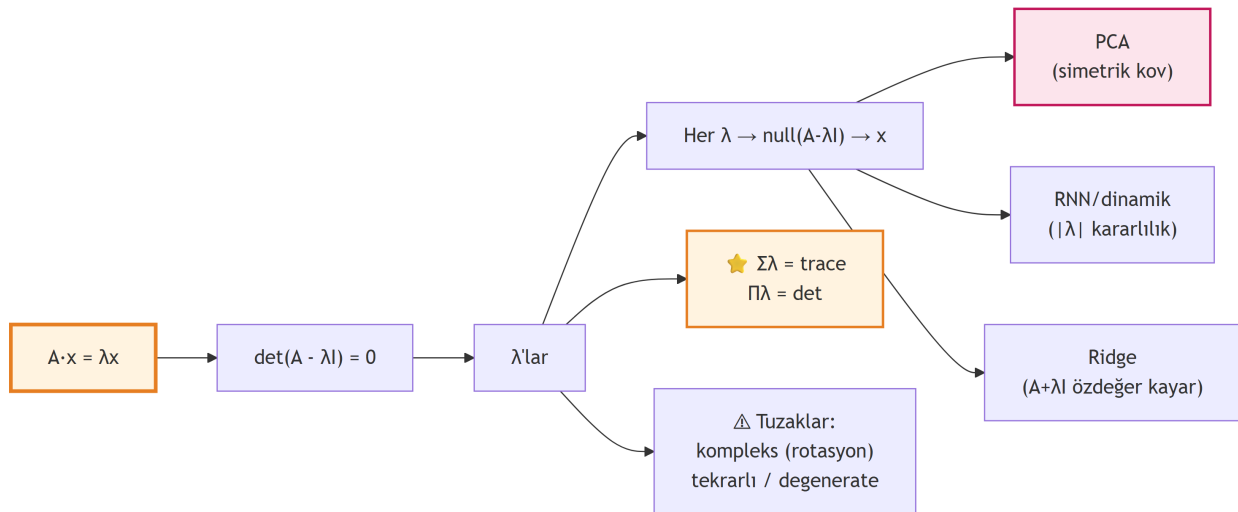
- **Strang'ın videosu:** [YouTube — Lecture 21: Eigenvalues and Eigenvectors](#) (≈51 dk)
- **OCW sayfası:** [MIT 18.06SC — Lecture 21](#)
- **Okuma süresi:** ≈40 dk

27.1 Bu Derste Ne Var?

Kursun zirvesi: **özdeğer/özvektör** — geri kalan derslerin çoğunu kaplar.

1. **Özvektör:** $Ax = \lambda x$.
2. **Karakteristik denklem:** $\det(A - \lambda I) = 0$.
3. $\sum \lambda = \text{trace}$, $\prod \lambda = \det$.
4. **Üç tuzak:** kompleks (rotasyon), tekrarlı, özvektör eksikliği.

“ Ax is some multiple lambda of x — that's our big equation.” — Strang, 2:36



Şekil 27.1: Özdeğer = doğal eksen ölçęi. PCA, RNN kararlılıęı, ridge regularization hepsi buradan.

💡 Builder Notu — Özdeğer ML'in Belkemiği

- **PCA** = kovaryans özvektörleri.
- **Spectral clustering / PageRank** = graf matrisleri özvektörleri.
- **RNN/dinamik kararlılık** = $|\lambda|$.
- **Trace** = $\sum \lambda$, **det** = $\prod \lambda \rightarrow \log\text{-det} = \sum \log \lambda$, etkin rank.
- **Simetrik** \rightarrow gerçel + ortogonal \rightarrow PCA, kernel yöntemleri sağlam.

27.2 $Ax = \lambda x$ — Özvektör

Ax genelde x 'ten farklı yönde. Bazı özel vektörler için **aynı doğrultuda, sadece ölçeklenmiş**:

$$Ax = \lambda x$$

λ pozitif/negatif/sıfır/kompleks olabilir. $n \times n$ matrisin (genelde) n özdeğer/özvektörü vardır — matrisin **doğal eksenleri**.

27.3 $\lambda = 0 \iff$ Singular

$$\lambda = 0 \rightarrow Ax = \mathbf{0} \rightarrow x \in N(A).$$

$$\lambda = 0 \text{ özdeğer} \iff A \text{ singular}$$

Builder Notu: **Graph Laplacian** sıfır özdeğeri (Ders 12); sayısı = bağlı bileşen sayısı. Sıfıra yakın özdeğerler = “neredeysen tekil” yönler = kararsızlık.

27.4 Geometriden Özdeğer Okuma

Projeksiyon P : düzlemdeki $x \rightarrow \lambda = 1$; dik $x \rightarrow \lambda = 0$.

Permütasyon $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$: $(1, 1) \rightarrow \lambda = 1$; $(-1, 1) \rightarrow \lambda = -1$.

“Those vectors are perpendicular. That'll happen for symmetric matrices.” — Strang, 22:33

Builder Notu: Mimari biliyorsan özdeğerleri tahmin et — projeksiyon $\{0, 1\}$, ortogonal $\{\pm 1, \text{birim çember}\}$, idempotent $\{0, 1\}$.

27.5 Trace = $\Sigma\lambda$, det = $\Pi\lambda$

$$\sum_i \lambda_i = \text{trace}(A) = \sum_i a_{ii}$$

$$\prod_i \lambda_i = \det(A)$$

2x2'de bu ikisi özdeğerleri **tamamen belirler**.

Permütasyon: trace = 0, det = -1 $\rightarrow \lambda_1 + \lambda_2 = 0$, $\lambda_1\lambda_2 = -1 \rightarrow \lambda = \pm 1$.

27.6 Karakteristik Denklem

$A\mathbf{x} = \lambda\mathbf{x} \rightarrow (A - \lambda I)\mathbf{x} = \mathbf{0}$. Sıfır-olmayan \mathbf{x} için $(A - \lambda I)$ singular:

$$\det(A - \lambda I) = 0$$

Strateji: (1) λ 'ları bul. (2) Her λ için null uzayı $\rightarrow \mathbf{x}$.

Builder Notu: Pratikte **asla** polinom kökünden değil — **QR iterasyonu** (np.linalg.eig) iteratif/kararlı.

27.7 Tam Örnek — $A = \begin{pmatrix} 3 & 1 \\ 1 & 3 \end{pmatrix}$

$$\det(A - \lambda I) = (3 - \lambda)^2 - 1 = \lambda^2 - 6\lambda + 8 = (\lambda - 4)(\lambda - 2)$$

$\lambda = 4, 2$. Trace 6 ✓, det 8 ✓.

- $\lambda = 4$: $A - 4I = \begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix} \rightarrow \mathbf{x} = (1, 1)$.
- $\lambda = 2$: $\mathbf{x} = (-1, 1)$.

Dik özvektörler (simetrik matris).

özdeğerler: [4. 2.]

özvektörler (kolonlar):

[[0.70710678 -0.70710678]

[0.70710678 0.70710678]]

trace = 6.0 = $\Sigma\lambda = 6.0000$

det = 8.0000 = $\Pi\lambda = 8.0000$

27.8 $A + cI$ ve $A + B$

$A + cI$: Özvektörler **değişmez**, özdeğerler $+c$ kayar.

$$(A + cI)\mathbf{x} = A\mathbf{x} + c\mathbf{x} = (\lambda + c)\mathbf{x}$$

$A + B$: Özdeğerler **toplanmaz** (genelde) — farklı özvektörler.

Builder Notu: Ridge regularization $A^T A + \lambda I$ tüm özdeğerleri λ kadar yukarı iter \rightarrow sifıra yakın özdeğerler pozitifleşir \rightarrow iyi-koşullu, tersinir. Özvektörler korunur.

27.9 Kompleks Özdeğerler — Rotasyon

$Q = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$ (90° döndür). Hiçbir gerçel vektör aynı doğrultuda kalmaz.

$$\det(Q - \lambda I) = \lambda^2 + 1 = 0 \implies \lambda = \pm i$$

Antisimetrik ($Q^T = -Q$) \rightarrow saf imajiner. **Simetrik** ($A^T = A$) \rightarrow gerçel (Ders 25).

27.10 Tekrarlı Özdeğer — Degenerate

$A = \begin{pmatrix} 3 & 1 \\ 0 & 3 \end{pmatrix}$ — özdeğerler 3, 3 (tekrarlı).

$A - 3I = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$ \rightarrow tek özvektör $(1, 0)$. **İkinci yok.**

Defective / degenerate \rightarrow diagonalize edilemez (Jordan formu, Ders 28).

“A repeated lambda is the source of all trouble.” — Strang, 50:11

Builder Notu: Üç tuzak:

- **Kompleks** \rightarrow osilasyon (RNN, dinamik).
- $\|\lambda\|$ \rightarrow kararlılık (< 1 sönümlü, > 1 patlar).
- **Degenerate** \rightarrow diagonalize edilemez; iyi haber: **simetrik matrisler asla degenerate olmaz.**

27.11 Bu Dersin Özeti

1. $A\mathbf{x} = \lambda\mathbf{x}$.
2. $\lambda = 0$ iff singular.
3. Projeksiyon (1, 0); permütasyon (± 1).
4. **Trace + det** özdeğerleri belirler.
5. **Karakteristik denklem.**
6. $A + cI$ özdeğer kayar (ridge).
7. **Kompleks** (rotasyon).
8. **Degenerate** (tekrarlı, eksik özvektör).

! Tek bir cümle

Özvektörler matrisin doğal eksenleri ($A\mathbf{x} = \lambda\mathbf{x}$); $\det(A - \lambda I) = 0$ ile λ bulunur, $\sum \lambda = \text{trace}$, $\prod \lambda = \det$. PCA, RNN kararlılığı, ridge — hepsi buradan.

27.12 Kontrol Soruları

i Soru 1: $A = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$ özdeğer/özvektör.

$(2 - \lambda)^2 - 1 = \lambda^2 - 4\lambda + 3 = (\lambda - 3)(\lambda - 1)$. $\lambda = 3, 1$.
 $\lambda = 3$: (1, 1). $\lambda = 1$: (-1, 1). Dik (simetrik).

i Soru 2: trace = 5, det = 6 — özdeğerler? 7 olabilir mi?

$\lambda^2 - 5\lambda + 6 = (\lambda - 2)(\lambda - 3) = 0 \rightarrow \mathbf{2, 3}$.
 7 olamaz: diğeri -2 olur, $7 \cdot (-2) = -14 \neq 6$.

i Soru 3: $A = \begin{pmatrix} 4 & 1 \\ 0 & 4 \end{pmatrix}$ bağımsız özvektör sayısı?

$\lambda = 4, 4$ (tekrarlı). $A - 4I$ rank 1 \rightarrow tek özvektör (1, 0).
1 bağımsız özvektör. Defective, diagonalize edilemez.

i Soru 4: Özdeğer PCA, RNN, ridge ile nasıl?

PCA: Simetrik kovaryans \rightarrow gerçel + ortogonal özvektör; en büyük $\lambda =$ en çok varyans yönü.

RNN: $A^k \mathbf{x}$ davranışı $|\lambda|$ ile: < 1 sönümlü, > 1 patlar, kompleks = osilasyon. Ortogonal ağırlık ($|\lambda| = 1$) kararlı.

Ridge: $A^T A + \lambda I \rightarrow$ tüm özdeğer $+\lambda$ kayar; sifıra yakın değerler pozitifleşir, iyi-koşullu.

27.13 Egzersizler

Egzersiz 1. $A = \begin{pmatrix} 1 & 2 \\ 2 & 4 \end{pmatrix}$ — det? Bir özdeğer?

Egzersiz 2. trace = 7, det = 12 → özdeğerler.

Egzersiz 3. 60° rotasyon → özdeğerler. Gerçek mi?

Egzersiz 4. (Python) `np.linalg.eig` + trace/det doğrulama.

Egzersiz 5. İspatla: $\sum \lambda = \text{trace}$, $\prod \lambda = \text{det}$. (İpucu: karakteristik polinom $\det(A - \lambda I) = \prod (\lambda_i - \lambda)$; $\lambda = 0 \rightarrow \text{det } A$; λ^{n-1} katsayısı trace.)

27.14 Sonraki Ders İçin Hazırlık

Ders 22: Diagonalization ve A^k

- $A = S\Lambda S^{-1}$.
- $A^k = S\Lambda^k S^{-1}$.
- Fibonacci, fark denklemleri.

⚠ Ders 22 öncesi

- Egzersiz 5 (trace/det = $\sum \lambda$, $\prod \lambda$).
- `np.linalg.eig` ile birkaç matris.

27.15 Anahtar Kavramlar (Cheat Sheet)

Kavram	Tanım	Strang'da
$A\mathbf{x} = \lambda\mathbf{x}$	Özvektör/özdeğer	2m36
$\lambda = 0 \iff$ singular	Null özvektörleri	4m12
Projeksiyon	$\{1, 0\}$	6m03
Permütasyon	$\{\pm 1\}$, dik	10m39
$\sum \lambda =$ trace	Köşegen toplamı	14m50
$\prod \lambda =$ det	—	14m50
$\det(A - \lambda I) = 0$	Karakteristik	16m51
$A + cI$	Özdeğer $+c$, özvektör aynı	31m40
Kompleks	Rotasyon $\rightarrow \pm i$	42m36
Degenerate	Tekrarlı + eksik özvektör	50m11

27.16 ML Bağlantıları Özeti

💡 7 köprü

1. **PCA = kovaryans özvektörleri.**
2. $\Sigma\lambda, \Pi\lambda \rightarrow \log\text{-det}$, etkin rank.
3. $\|\lambda\| = \mathbf{RNN}$ kararlılık.
4. **Kompleks $\lambda = \text{osilasyon}$.**
5. $A + cI = \mathbf{ridge} \rightarrow$ özdeğerleri yukarı kaydır.
6. **Degenerate** \rightarrow Jordan (Ders 28).
7. **Simetrik** \rightarrow PCA, spectral, kernel sağlam temeli.

❗ Tek bir şey alıp gideceksen

Özvektörler doğal eksenler ($A\mathbf{x} = \lambda\mathbf{x}$); $\sum \lambda = \text{trace}$, $\prod \lambda = \text{det}$. PCA, RNN kararlılığı, ridge — modern ML'in belkemiği.

28 Diagonalization ve A^k

$A = SAS^{-1}$ — kuvvetler, dinamik, Fibonacci

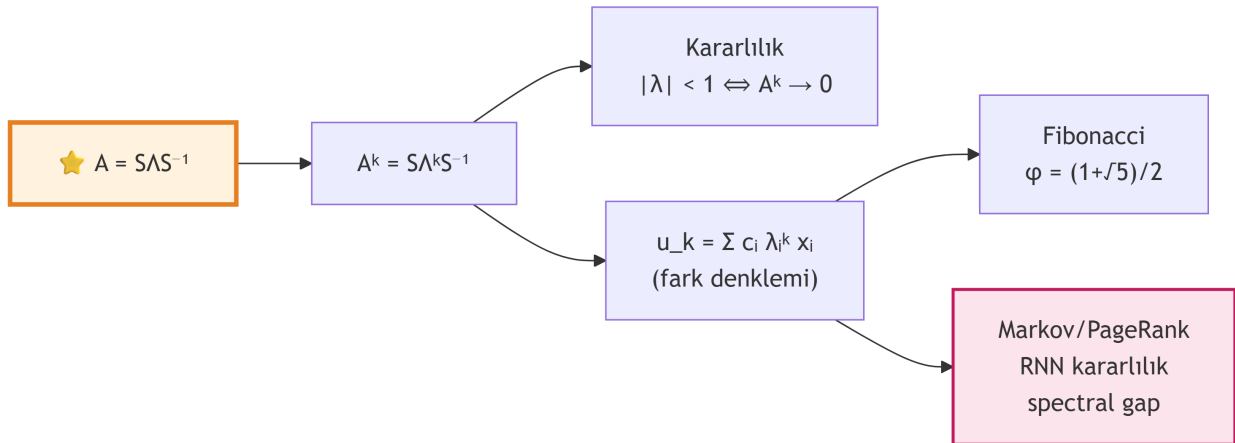
i Bölüm bilgisi

- **Strang'ın videosu:** [YouTube — Lecture 22: Diagonalization and Powers of A](#) (≈52 dk)
- **OCW sayfası:** [MIT 18.06SC — Lecture 22](#)
- **Okuma süresi:** ≈40 dk


28.1 Bu Derste Ne Var?

1. **Diagonalization:** $A = SAS^{-1}$.
2. $A^k = S\Lambda^k S^{-1}$ — kuvvetler kolaylaşır.
3. **Kararlılık:** $A^k \rightarrow 0$ iff tüm $|\lambda| < 1$.
4. **Fark denklemi:** $u_{k+1} = Au_k$, **Fibonacci** ve altın oran.

“A to the K power is S lambda to the K S inverse.” — Strang, 13:08



Şekil 28.1: $A = SAS^{-1} \rightarrow A^k$, kararlılık, fark denklemleri, Fibonacci/altın oran.

 Builder Notu — Dinamik Sistemlerin Omurgası

- **Markov/PageRank** → dominant $\lambda = 1$ + kararlı durum özvektörü.
- **RNN kararlılığı** → spektral yarıçap; $|\lambda| > 1$ patlar, < 1 söner.
- **Fark denklemi** → dizi modelleri, linear recurrence.
- **Spectral gap** $1 - |\lambda_2|$ → MCMC karışma hızı.

28.2 $AS = SA$ ve $A = SAS^{-1}$

Özvektörleri S 'nin kolonlarına koy:

$$AS = SA, \quad \Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$$

S tersinirse:

$$A = SAS^{-1}$$

Koşul: n bağımsız özvektör. Tekrarlı + eksik özvektör → diagonalize edilemez (degenerate).

Builder Notu: Bir matrisi **kendi doğal koordinatlarına** çevirir; PCA verinin kovaryansını köşegenleştirir.

28.3 $A^k = S\Lambda^k S^{-1}$

$$A^2 = SA \underbrace{S^{-1}S}_I \Lambda S^{-1} = S\Lambda^2 S^{-1}$$

Ortadaki sadeleşir, k kez tekrar → $A^k = S\Lambda^k S^{-1}$.

$\Lambda^k = \text{diag}(\lambda_1^k, \dots)$ — köşegen, bedava.

A^{100} doğrudan imkânsız; $S\Lambda^{100}S^{-1}$ ile anında.

özdeğerler (altın oran): [1.61803399 -0.61803399]

A^{10} (spektral):

[[89. 55.]

[55. 34.]]

A^{10} (doğrudan):

[[89. 55.]

[55. 34.]]

28.4 Kararlılık

$$A^k \rightarrow 0 \iff \text{tüm } |\lambda_i| < 1$$

“A to the K approaches zero if all eigenvalues are less than one in absolute value.” — Strang, 14:53

Spektral yarıçap = en büyük $|\lambda|$ = uzun-vadeli davranış.

💡 Builder Notu — ML’de Kararlılık

- **RNN spektral yarıçapı** $> 1 \rightarrow$ exploding gradient; $< 1 \rightarrow$ vanishing \rightarrow **ortogonal başlatma** ($|\lambda| = 1$).
- **Markov:** dominant $\lambda = 1$ kararlı durum, diğerleri söner.
- **Spectral gap:** karışma hızı (MCMC).

28.5 Multiplicity

- **Cebirsel:** karakteristik polinomda kök tekrar sayısı.
- **Geometrik:** $\dim N(A - \lambda I)$ — bağımsız özvektör sayısı.

Geometrik $<$ **cebirsel** \rightarrow degenerate, diagonalize edilemez.

Builder Notu: Simetrik matrisler asla degenerate olmaz (spektral teorem, Ders 25) — kovaryans, Gram, kernel, Laplacian güvenle diagonalize edilir.

28.6 Fark Denklemleri ve Spektral Çözüm

$$\mathbf{u}_{k+1} = A\mathbf{u}_k \implies \mathbf{u}_k = A^k \mathbf{u}_0$$

Spektral numara: \mathbf{u}_0 ’ı özvektörlere ayır \rightarrow her özvektör bağımsız evrilir:

$$\mathbf{u}_0 = \sum c_i \mathbf{x}_i \implies \mathbf{u}_k = \sum c_i \lambda_i^k \mathbf{x}_i$$

Uzun vadede en büyük $|\lambda|$ baskın.

Builder Notu: Markov, dolaşım, difüzyon, Fourier mod analizi — hepsi spektral ayrışım.

28.7 Fibonacci Örneği

$F_{k+2} = F_{k+1} + F_k$. Vektörleştir: $\mathbf{u}_k = (F_{k+1}, F_k)^T$:

$$A = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}, \quad \det(A - \lambda I) = \lambda^2 - \lambda - 1 = 0$$

(Karakteristik denklem = Fibonacci recursion'unun kendisi!)

$$\lambda_1 = \frac{1 + \sqrt{5}}{2} \approx 1.618 = \varphi \text{ (altın oran)}, \quad \lambda_2 \approx -0.618$$

Çözüm: $F_k \approx c_1 \varphi^k + c_2 \lambda_2^k$. $|\lambda_2| < 1 \rightarrow$ söner $\rightarrow F_k \approx c_1 \varphi^k$.

Her adımda ~1.618 kat büyür.

28.8 Bu Dersin Özeti

1. $AS = S\Lambda$.
2. $A = S\Lambda S^{-1}$.
3. $A^k = S\Lambda^k S^{-1}$.
4. **Kararlılık** $|\lambda| < 1$.
5. **Distinct $\lambda \rightarrow$ diagonalize edilebilir.**
6. **Multiplicity** (degenerate).
7. **Fark denklemi.**
8. **Spektral çözüm.**
9. **Fibonacci $\rightarrow \varphi$.**
10. **Büyüme = $\max |\lambda|$.**

! Tek bir cümle

$A = S\Lambda S^{-1}$ matrisin doğal eksenlerini açar; $A^k = S\Lambda^k S^{-1}$ kuvvetleri çözer. Fark denklemi çözümü $\mathbf{u}_k = \sum c_i \lambda_i^k \mathbf{x}_i$; uzun-vadeli davranış **en büyük** $|\lambda|$ ile (kararlılık, büyüme, kararlı durum).

28.9 Kontrol Soruları

i Soru 1: $A = \begin{pmatrix} 2 & 0 \\ 0 & 3 \end{pmatrix} \rightarrow S, \Lambda$?

Zaten köşegen: $S = I, \Lambda = A$. Özvektörler $(1, 0), (0, 1)$.

i Soru 2: $A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \rightarrow A^{10}$?

$$\lambda = \pm 1. \Lambda^{10} = \text{diag}(1, 1) = I.$$

$$A^{10} = S I S^{-1} = I. (\text{Çift kuvvetler } I, \text{ tek kuvvetler } A.)$$

i Soru 3: $\lambda = 0.5, -0.9$ kararlı mı? 0.5, 1.2?

- $|0.5|, |0.9| < 1 \rightarrow \mathbf{A^k} \rightarrow \mathbf{0}$, **kararlı** (en büyük 0.9 söndürür).
- $|1.2| > 1 \rightarrow$ **patlar**.

Kural: spektral yarıçap.

i Soru 4: Markov / PageRank \rightarrow özdeğer/özvektör.

Markov: kolonları olasılık. En büyük $\lambda = 1$, özvektörü = **kararlı durum**.

$\mathbf{u}_k \rightarrow c_1 \mathbf{x}_1$ (sönenler kaybolur).

PageRank: $\lambda = 1$ özvektörü = sayfa skorları. Power iteration ile hesap.

Karışma hızı: $1 - |\lambda_2|$ (spectral gap); MCMC verimliliği.

28.10 Egzersizler

Egzersiz 1. $A = \begin{pmatrix} 4 & 1 \\ 0 & 3 \end{pmatrix} \rightarrow$ diagonalize, A^5 .

Egzersiz 2. Lucas dizisi ($L_0 = 2, L_1 = 1$) — büyüme oranı?

Egzersiz 3. $\lambda = 1, 0.5, -0.3$ — \mathbf{u}_k uzun vadede?

Egzersiz 4. (Python) eig + matrix_power ile A^{10} .

Egzersiz 5. İspatla: $A = S \Lambda S^{-1} \rightarrow$ polinom $p(A) = S \cdot p(\Lambda) \cdot S^{-1}$. Ders 23 (e^{At}) temeli.

28.11 Sonraki Ders İçin Hazırlık

Ders 23: Diferansiyel Denklemler ve e^{At}

- Sürekli: $\frac{d\mathbf{u}}{dt} = \mathbf{A}\mathbf{u} \rightarrow \mathbf{u}(t) = e^{At}\mathbf{u}(0)$.
- $e^{At} = S e^{\Lambda t} S^{-1}$.
- Kararlılık: $\text{Re}(\lambda) < 0$.

⚠ Ders 23 öncesi

- Egzersiz 5 ($p(A)$).
- matrix_power ile diagonalizasyonu doğrula.

28.12 Anahtar Kavramlar (Cheat Sheet)

Kavram	Tanım	Strang'da
$AS = SA$	Özvektör + özdeğer matrisi	1m19
$A = S\Lambda S^{-1}$	Diagonalize	8m44
$A^k = S\Lambda^k S^{-1}$	Kuvvetler	12m32
Kararlılık	$ \lambda < 1$	14m24
Distinct	Bağımsız özvektör garanti	18m05
Multiplicity	Geom < cebir \rightarrow degenerate	22m14
Fark denklemleri	$\mathbf{u}_{k+1} = A\mathbf{u}_k$	26m46
Spektral çözüm	$\sum c_i \lambda_i^k \mathbf{x}_i$	29m23
Fibonacci	$A = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}, \lambda^2 - \lambda - 1$	34m29
Altın oran	$\varphi \approx 1.618$	44m22

28.13 ML Bağlantıları Özeti

💡 7 köprü

1. $A = S\Lambda S^{-1} = \text{baz değişimi}$ \rightarrow PCA köşegenleştirme.
2. $A^k = \text{spektral}$ \rightarrow Dinamik sistem, iteratif algoritmalar.
3. $|\lambda| = \text{RNN kararlılık}$ \rightarrow Ortogonal başlatma.
4. **Markov/PageRank** \rightarrow Dominant λ .
5. **Fark denklemleri** \rightarrow Dizi modelleri.
6. **Spectral gap** \rightarrow MCMC karışma.
7. **Simetrik diagonalize** \rightarrow Kovaryans/Gram/Laplacian sağlam.

! Tek bir şey alıp gideceksen

$A = S\Lambda S^{-1}$ doğal eksenler; $A^k = S\Lambda^k S^{-1}$ kuvvetler; çözüm $\sum c_i \lambda_i^k \mathbf{x}_i$. **Max** $|\lambda| = \text{kararlılık/büyüme}$
— PageRank, RNN, MCMC kalbi.

29 Diferansiyel Denklemler ve e^{At}

Sürekli dinamik — neural ODE’lerin lineer çekirdeği

i Bölüm bilgisi

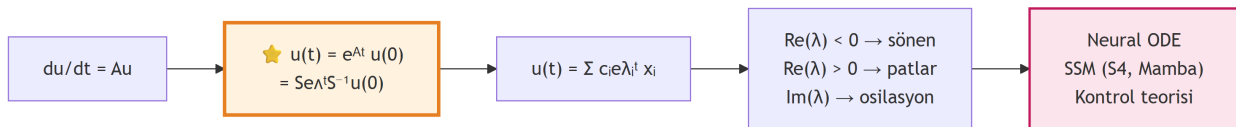
- **Strang’in videosu:** [YouTube — Lecture 23: Differential Equations and \$\exp\(At\)\$](#) (≈ 51 dk)
- **OCW sayfası:** [MIT 18.06SC — Lecture 23](#)
- **Okuma süresi:** ≈ 40 dk

29.1 Bu Derste Ne Var?

Ders 22’nin sürekli versiyonu: $\frac{du}{dt} = Au$.

1. **Üstel çözümler:** her özvektör $\rightarrow u = e^{\lambda t} x$.
2. **Kararlılık:** tüm $\text{Re}(\lambda) < 0$ (sol yarı düzlem).
3. **Matris üstel:** $e^{At} = Se^{\Lambda t}S^{-1}$, $u(t) = e^{At}u(0)$.
4. **İkinci-mertebe \rightarrow sistem** (companion matris).

“The solutions to constant coefficient linear equations are exponentials.” — Strang, 0:32



Şekil 29.1: $du/dt = Au \rightarrow e^{At}$ propagatör. $\text{Re}(\lambda)$ kararlılık, kompleks osilasyon, neural ODE/SSM çekirdeği.

💡 Builder Notu — Sürekli Dinamik ML’de

- **Neural ODE** — derin ağ = sürekli sistem $\frac{du}{dt} = f(u)$.
- **SSM (S4, Mamba)** — tam $\frac{du}{dt} = Au + Bx$ formunda; HiPPO başlatma özdeğer yapısı.
- **Kararlı RNN/SSM** \rightarrow özdeğerleri **sol yarı düzleme** ($\text{Re}(\lambda) < 0$) kısıtla.
- **Ayrık vs sürekli:** $|\lambda| < 1$ (birim çember) vs $\text{Re}(\lambda) < 0$ (sol yarı düzlem).

29.2 $du/dt = Au$ ve Üstel Çözümler

Her özvektör pür üstel çözüm verir: $Ax = \lambda x \rightarrow u = e^{\lambda t} x$ ($du/dt = \lambda e^{\lambda t} x = Au$ ✓).

Genel çözüm:

$$u(t) = \sum c_i e^{\lambda_i t} x_i$$

29.3 Örnek — Steady State

$$A = \begin{pmatrix} -1 & 2 \\ 1 & -2 \end{pmatrix}, \quad u(0) = (1, 0)^T$$

Singular (kolon 2 = $-2 \times$ kolon 1) $\rightarrow \lambda_1 = 0$. Trace = $-3 \rightarrow \lambda_2 = -3$.

- $\lambda_1 = 0$: $x_1 = (2, 1)$.
- $\lambda_2 = -3$: $x_2 = (1, -1)$.

$$u(0) = \frac{1}{3}x_1 + \frac{1}{3}x_2:$$

$$u(t) = \frac{1}{3}(2, 1)^T + \frac{1}{3}e^{-3t}(1, -1)^T \xrightarrow{t \rightarrow \infty} \left(\frac{2}{3}, \frac{1}{3}\right)^T$$

$\lambda = 0$ modu kalıcı (steady state); $\lambda = -3$ söner.

29.4 Kararlılık — $\text{Re}(\lambda) < 0$

$$u(t) \rightarrow 0 \iff \text{Re}(\lambda_i) < 0 \forall i$$

Üç durum:

- Tüm $\text{Re} < 0 \rightarrow$ kararlı, sönen.
- Bir $\lambda = 0$, diğerleri $\text{Re} < 0 \rightarrow$ steady state.
- Bir $\text{Re} > 0 \rightarrow$ patlar.

Ayrık vs sürekli: $|\lambda| < 1$ (birim çember) $\leftrightarrow \text{Re}(\lambda) < 0$ (sol yarı düzlem).

29.5 Kompleks Özdeğerler — Osilasyon

$\lambda = a + bi \rightarrow |e^{\lambda t}| = e^{at}$. Sadece gerçel kısım büyüklüğü belirler; imajiner = salınım.

Antisimetrik matris \rightarrow saf imajiner $\lambda \rightarrow$ sönmeyen salınım.

29.6 2x2 Kararlılık — Trace ve Determinant

$$\text{trace} < 0 \text{ ve } \det > 0$$

İkisi de gerekli: trace < 0 (toplam negatif), det > 0 (aynı işaret → ikisi de negatif).

Karşı-örnek: $A = \text{diag}(-2, 1)$, trace = -1, det = -2 → patlar.

29.7 Uncoupling — $\mathbf{u} = S\mathbf{v}$

$\mathbf{u} = S\mathbf{v}$:

$$\frac{d\mathbf{v}}{dt} = S^{-1}AS\mathbf{v} = \Lambda\mathbf{v}$$

Bağımsız denklemler $dv_i/dt = \lambda_i v_i \rightarrow v_i(t) = e^{\lambda_i t} v_i(0)$.

29.8 Matris Üstel — e^{At}

$$e^{At} = I + At + \frac{(At)^2}{2!} + \dots = \sum_{n=0}^{\infty} \frac{(At)^n}{n!}$$

Her zaman yakınsar ($n!$ paydası). Diğer seri $(I - At)^{-1}$ sadece $|\lambda| < 1$ için yakınsar; üstel daha güvenli.

29.9 $e^{At} = Se^{\Lambda t}S^{-1}$

Taylor serisine $A = S\Lambda S^{-1}$ koy $\rightarrow A^n = S\Lambda^n S^{-1}$, ortadakiler sadeleşir:

$$e^{At} = Se^{\Lambda t}S^{-1}, \quad e^{\Lambda t} = \text{diag}(e^{\lambda_1 t}, \dots, e^{\lambda_n t})$$

özdeğerler: [0. -3.]

$u(2) = [0.66749292 \ 0.33250708]$

$u(100) \approx [0.66666667 \ 0.33333333]$ (= (2/3, 1/3))

29.10 İkinci-Mertebe → Sistem (Companion)

$$y'' + by' + ky = 0 \rightarrow \mathbf{u} = (y', y)^T:$$

$$\mathbf{u}' = \begin{pmatrix} -b & -k \\ 1 & 0 \end{pmatrix} \mathbf{u}$$

Companion matrix — state-space temsili.

Builder Notu: Kontrol teorisi, robotik, Kalman filtreleri, **neural ODE**, **SSM** hep state-space formunda. Özdeğerler = orijinal karakteristik kökleri.

29.11 Bu Dersin Özeti

1. $d\mathbf{u}/dt = A\mathbf{u}$ üstel çözüm.
2. **Genel çözüm** = mod toplamı.
3. **Steady state** ($\lambda = 0$).
4. **Kararlılık** $\text{Re}(\lambda) < 0$.
5. **Kompleks osilasyon**.
6. **2x2:** trace < 0 , det > 0 .
7. **Uncoupling**.
8. e^{At} Taylor.
9. $e^{At} = S e^{\Lambda t} S^{-1}$.
10. **Companion (state-space)**.

! Tek bir cümle

$\mathbf{u}(t) = e^{At} \mathbf{u}(0) = S e^{\Lambda t} S^{-1} \mathbf{u}(0)$; özdeğerler tüm davranışı belirler — $\text{Re}(\lambda) < 0$ söndürür, > 0 patlatır, $\text{Im}(\lambda)$ salındırır. Neural ODE'den SSM'e modern ML'in sürekli omurgası.

29.12 Kontrol Soruları

i Soru 1: $A = \begin{pmatrix} 1 & 0 \\ 0 & -2 \end{pmatrix}$, $\mathbf{u}(0) = (3, 5)$. $\mathbf{u}(t)$ ve $t \rightarrow \infty$?

$$\mathbf{u}(t) = (3e^t, 5e^{-2t})$$

$t \rightarrow \infty$: $5e^{-2t} \rightarrow 0$, $3e^t \rightarrow \infty \rightarrow$ **patlar** ($\lambda = 1 > 0$).

i Soru 2: Hangileri kararlı? (a) -1, -2 (b) $-1 \pm 3i$ (c) 0, -5 (d) 2, -3

- (a) kararlı (her ikisi $\text{Re} < 0$).
- (b) kararlı ($\text{Re} = -1$, salınarak söner).
- (c) steady state'e gider (sıfıra değil ama patlamaz).
- (d) kararsız ($\text{Re} = 2 > 0$).

i Soru 3: $A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \rightarrow e^{At}$?

$A^2 = 0$ (nilpotent!). Taylor kısadır:

$$e^{At} = I + At = \begin{pmatrix} 1 & t \\ 0 & 1 \end{pmatrix}$$

Polinom çıkması = Jordan formu imzası (Ders 28). Degenerate (tekrarlı $\lambda = 0$, tek özvektör).

i Soru 4: Neural ODE / SSM / kontrol $\rightarrow du/dt = Au$?

Neural ODE: $du/dt = f(\mathbf{u}, \theta)$, lineer hâli $A\mathbf{u}$, $\mathbf{u}(T) = e^{AT}\mathbf{u}(0)$.

SSM (S4, Mamba): $du/dt = A\mathbf{u} + B\mathbf{x}$, $\mathbf{y} = C\mathbf{u}$; e^{At} ile ayrıştırılır.

Kararlılık: $\text{Re}(\lambda) < 0 \rightarrow$ kararlı RNN/SSM. **HiPPO** başlatma uzun-menzilli bağımlılıkları yakalar.

Kompleks λ : zaman serisi salınım modelleri.

29.13 Egzersizler

Egzersiz 1. $A = \begin{pmatrix} -2 & 1 \\ 1 & -2 \end{pmatrix}$, $\mathbf{u}(0) = (1, 0)$ — çözüm, kararlı mı?

Egzersiz 2. Kararlılık testi: (a) $\begin{pmatrix} -1 & 2 \\ 0 & -3 \end{pmatrix}$, (b) $\begin{pmatrix} 1 & 0 \\ 0 & -5 \end{pmatrix}$, (c) $\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$.

Egzersiz 3. $y'' + 4y = 0 \rightarrow$ companion + özdeğerler. Salınlı mı?

Egzersiz 4. (Python) `scipy.linalg.expm` + steady state.

Egzersiz 5. İspatla: $\frac{d}{dt}e^{At} = Ae^{At}$. (Taylor terim terim türev.)

29.14 Sonraki Ders İçin Hazırlık

Ders 24: Markov Matrisleri ve Fourier Serisi

- **Markov:** $\lambda = 1$ (kararlı durum), diğerleri sönen.
- **Fourier:** fonksiyon uzayında ortonormal sin/cos.

⚠ Ders 24 öncesi

- Egzersiz 5 (e^{At} türev).
- `expm` ile steady state.

29.15 Anahtar Kavramlar (Cheat Sheet)

Kavram	Tanım	Strang'da
$d\mathbf{u}/dt = A\mathbf{u}$	Üstel çözüm	0m32
Genel çözüm	$\sum c_i e^{\lambda_i t} \mathbf{x}_i$	8m54
Kararlılık	$\text{Re}(\lambda) < 0$	16m45
Kompleks λ	Re büyüklük, Im osilasyon	17m50
2x2 kararlılık	trace < 0, det > 0	22m02
Uncoupling	$\mathbf{u} = S\mathbf{v}$	29m14
Matris üstel	$\sum (At)^n / n!$	33m55
$e^{At} = S e^{\Lambda t} S^{-1}$	—	39m02
Companion	İkinci-mertebe \rightarrow sistem	47m38

29.16 ML Bağlantıları Özeti

💡 7 köprü

1. **Neural ODE / SSM** \rightarrow Sürekli derin öğrenme.
2. e^{At} **propagatör** \rightarrow Fizik, kontrol, Schrödinger.
3. **$\text{Re}(\lambda) < 0 =$ kararlı** \rightarrow S4/Mamba tasarımı.
4. **Kompleks $\lambda =$ osilasyon** \rightarrow Zaman serisi.
5. **Uncoupling** \rightarrow Mod analizi, Fourier akrabası.
6. **Companion** \rightarrow State-space, Kalman.
7. **Ayrık vs sürekli** \rightarrow Aynı fikir, iki geometri.

❗ Tek bir şey alıp gideceksen

$d\mathbf{u}/dt = A\mathbf{u} \rightarrow \mathbf{u}(t) = e^{At}\mathbf{u}(0) = S e^{\Lambda t} S^{-1}\mathbf{u}(0)$. Özdeğerler tüm davranışı belirler — $\text{Re}(\lambda)$ söndürme/büyüme, $\text{Im}(\lambda)$ osilasyon. Neural ODE, SSM, kontrol teorisi kalbi.

30 Markov Matrisleri ve Fourier Serisi

$\lambda = 1$ kararlı durum + ortonormal fonksiyon bazı

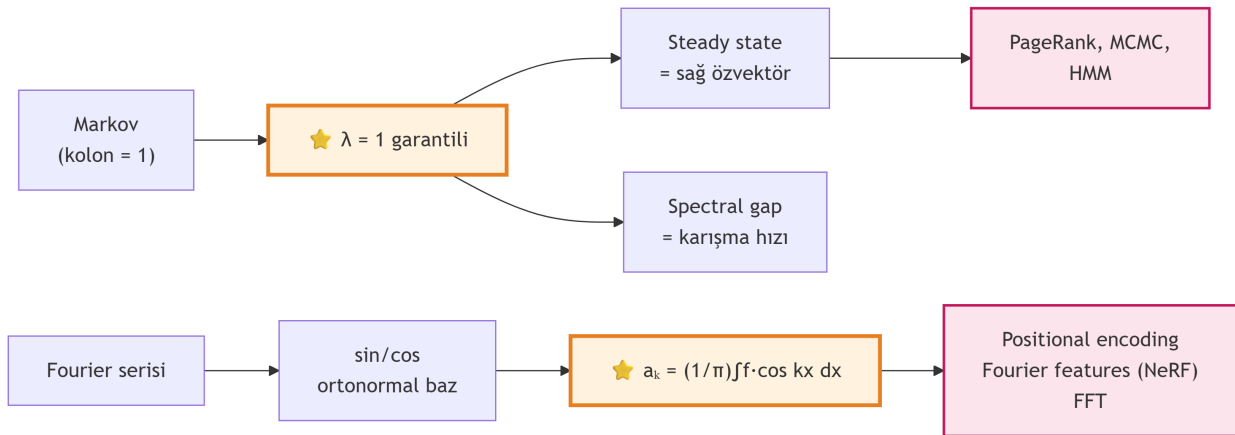
i Bölüm bilgisi

- **Strang'ın videosu:** [YouTube — Lecture 24: Markov Matrices, Fourier Series](#) (≈51 dk)
- **OCW sayfası:** [MIT 18.06SC — Lecture 24](#)
- **Okuma süresi:** ≈40 dk

30.1 Bu Derste Ne Var?

1. **Markov matrisi:** girişler ≥ 0 , kolonlar 1'e toplanır; her zaman $\lambda = 1$.
2. **Steady state** = $\lambda = 1$ özvektörü.
3. **A ve A^T aynı özdeğer.**
4. **Fourier serisi:** fonksiyonlar sonsuz boyutlu ortonormal sin/cos bazında.

“The steady state will be the eigenvector for that eigenvalue (lambda = 1).” — Strang, 3:55



Şekil 30.1: Markov + Fourier — özdeğer/ortogonalliğin iki güçlü uygulaması.

💡 Builder Notu — Markov + Fourier ML'de

- **Markov / PageRank / MCMC** — kararlı durum = $\lambda = 1$ özvektörü.
- **Spectral gap** $1 - |\lambda_2| \rightarrow$ MCMC karışma hızı.
- **Fourier serisi** \rightarrow transformer positional encoding, NeRF Fourier features, FFT konvolüsyon hızlandırma.
- **Ortonormal projeksiyon** ($x_i = \mathbf{q}_i^T \mathbf{v}$) \rightarrow PCA, wavelet, embedding ayrışımı.

30.2 Markov Matrisi

İki özellik: (1) girişler ≥ 0 , (2) her kolon 1'e toplanır.

A^k de Markov; özellikler korunur.

30.3 $\lambda = 1$ Garantili

A 'nın kolonları 1'e toplanır $\rightarrow A - I$ 'nin kolonları 0'a toplanır \rightarrow satırlar bağımlı (örn. $(1, 1, \dots, 1) \cdot (A - I) = \mathbf{0}$) $\rightarrow A - I$ singular $\rightarrow \lambda = 1$ özdeğer.

$(1, 1, \dots, 1)$ vektörü = A^T 'nin $\lambda = 1$ sol özvektörü (olasılık korunumu).

30.4 Diğer λ ve Steady State

Diğer tüm $|\lambda| \leq 1$ (1'i aşmaz, olasılık korunur).

$$A^k \mathbf{u}_0 = c_1 \mathbf{x}_1 + c_2 \lambda_2^k \mathbf{x}_2 + \dots$$

$\lambda_2^k, \dots \rightarrow 0$; sadece $\lambda = 1$ **modu** hayatta kalır:

$$A^k \mathbf{u}_0 \rightarrow c_1 \mathbf{x}_1 = \text{steady state}$$

(Pozitif özvektör — başlangıç pozitifse kalır.)

30.5 A ve A^T Aynı Özdeğer

$$\det(A - \lambda I) = \det((A - \lambda I)^T) = \det(A^T - \lambda I).$$

Özdeğerler aynı, özvektörler farklı (sol/sağ).

30.6 Markov Örneği — CA/MA

$$A = \begin{pmatrix} 0.9 & 0.2 \\ 0.1 & 0.8 \end{pmatrix}$$

trace = 1.7, det = 0.7 $\rightarrow \lambda^2 - 1.7\lambda + 0.7 = 0 \rightarrow \lambda = 1, 0.7$.

- $\lambda = 1$: $\mathbf{x}_1 = (2, 1)$.
- $\lambda = 0.7$: $\mathbf{x}_2 = (-1, 1)$.

$\mathbf{u}_0 = (0, 1000) \rightarrow c_1 = 1000/3, c_2 = 2000/3$.

Steady state: $\frac{1000}{3}(2, 1) = (667, 333)$. Toplama 1000 korunur.

özdeğerler: [1. 0.7]

steady state dağılımı: [0.66666667 0.33333333]

A^{100} u0: [666.66666667 333.33333333]

Builder Notu: Karışma hızı = $1 - |\lambda_2|$ = spectral gap. CA/MA: 0.3, hızlı. Web grafında power iteration ile PageRank.

30.7 Ortonormal Bazda Projeksiyon

$\mathbf{v} = \sum x_i \mathbf{q}_i$. Her iki yanı \mathbf{q}_i ile çarp:

$$\mathbf{q}_i^T \mathbf{v} = x_i \quad (\text{diğerleri } 0)$$

Her katsayı bağımsız dot product. Matris dilinde $\mathbf{x} = Q^T \mathbf{v}$.

30.8 Fourier Serisi

$$f(x) = a_0 + a_1 \cos x + b_1 \sin x + a_2 \cos 2x + \dots$$

Sonsuz boyutlu fonksiyon uzayı; baz = $\{1, \cos x, \sin x, \cos 2x, \sin 2x, \dots\}$.

Periyot 2π ile periyodik.

Builder Notu: Sinyal işleme, **transformer positional encoding** (sin/cos frekansları), **NeRF / Fourier features**, JPEG (DCT), Gaussian process RKHS.

30.9 Fonksiyon İç Çarpımı

$$\langle f, g \rangle = \int_0^{2\pi} f(x)g(x) dx$$

Toplamın sürekli karşılığı. Bu iç çarpımda sin, cos ortogonal:

$$\int_0^{2\pi} \sin x \cos x dx = 0$$

30.10 Fourier Katsayıları

$\mathbf{q}_i^T \mathbf{v}$ 'nin fonksiyon versiyonu:

$$\int_0^{2\pi} f(x) \cos(kx) dx = a_k \int_0^{2\pi} \cos^2(kx) dx = a_k \pi$$

$$a_k = \frac{1}{\pi} \int_0^{2\pi} f(x) \cos(kx) dx$$

Euler-Fourier formülü. Ortonormal bazda projeksiyon.

Builder Notu: Ayrık versiyonu **FFT** — konvolüsyonu frekansta çarpıma çevirir (sinyal işleme, FNet, uzun-konvolüsyon).

30.11 Bu Dersin Özeti

1. **Markov:** girişler ≥ 0 , kolon = 1.
2. $\lambda = 1$ **garantili.**
3. **Diğer** $|\lambda| \leq 1$, steady state.
4. $\mathbf{A} \leftrightarrow \mathbf{A}^T$ **özdeğer.**
5. **CA/MA örneği**, (667, 333).
6. **Çözüm spektral.**
7. **Ortonormal projeksiyon** $x_i = \mathbf{q}_i^T \mathbf{v}$.
8. **Fourier serisi.**
9. **Fonksiyon iç çarpımı.**
10. **Fourier katsayıları.**

! Tek bir cümle

Markov: $\lambda = 1$ + pozitif özvektör = steady state (PageRank, MCMC). **Fourier:** ortonormal sin/cos bazında projeksiyon ($a_k = \frac{1}{\pi} \int f \cos kx dx$) = transformer positional encoding, NeRF, FFT.

30.12 Kontrol Soruları

i Soru 1: $A = \begin{pmatrix} 0.8 & 0.3 \\ 0.2 & 0.7 \end{pmatrix}$ Markov mı? Özdeğer?

Kolonlar: 1.0, 1.0 ✓; girişler ≥ 0 ✓ → **Markov**. trace 1.5 → $\lambda = 1, 0.5$.

i Soru 2: Aynı A için steady state (toplam = 1).

$A - I = \begin{pmatrix} -0.2 & 0.3 \\ 0.2 & -0.3 \end{pmatrix}$. Null: (3, 2). Normalize: (0.6, 0.4).

i Soru 3: $f = \sin(x) \rightarrow b_1$ ve a_1 ?

Zaten baz fonksiyonu! $b_1 = 1, a_1 = 0$ ($\sin \perp \cos$). Diğer hepsi 0.

i Soru 4: Markov → PageRank/MCMC, Fourier → transformer.

PageRank: web grafi Markov → $\lambda = 1$ özvektörü = sayfa skorları. Power iteration; spectral gap = yakınsama.

MCMC: hedef dağılıma yakınsayan zincir; karışma hızı = spectral gap.

Transformer positional encoding: sin/cos frekansları → sıra bilgisi.

Fourier features (NeRF): koordinatları sin/cos bazında genişletme → yüksek-frekans detay.

FFT: konvolüsyon → frekansta çarpım (FNet, uzun-konv modeller).

30.13 Egzersizler

Egzersiz 1. $A = \begin{pmatrix} 0.5 & 0.4 \\ 0.5 & 0.6 \end{pmatrix}$ Markov mı? Özdeğer + steady state.

Egzersiz 2. Steady state (0.6, 0.4), $\lambda_2 = 0.9$ — yakınsama hızı?

Egzersiz 3. $f = \cos 2x - 3 \sin x \rightarrow$ Fourier katsayıları.

Egzersiz 4. (Python) Markov + power iteration.

Egzersiz 5. *İspatla*: Markov için $|\lambda| \leq 1$ tüm λ için. (İpucu: aksi takdirde A^k patlar, olasılık korunmaz.)

30.14 Sonraki Ders İçin Hazırlık

Ders 24b: Quiz 2 İncelemesi — Determinantlar + özdeğerler + uygulamalar tekrarı.

⚠ Ders 24b öncesi

- Egzersiz 5 ($|\lambda| \leq 1$).
- Power iteration ile steady state.

30.15 Anahtar Kavramlar (Cheat Sheet)

Kavram	Tanım	Strang'da
Markov	Girişler ≥ 0 , kolon = 1	1m01
$\lambda = 1$	$A - I$ kolon = 0 \rightarrow singular	9m08
Steady state	$\lambda = 1$ özvektör	7m01
$ \lambda \leq 1$	Olasılık korunumu	5m12
A, A^T özdeğer	Aynı, özvektör farklı	16m50
Ortonormal proj	$x_i = \mathbf{q}_i^T \mathbf{v}$	35m24
Fourier	$f = a_0 + \sum (a_k \cos + b_k \sin)$	41m08
Fonksiyon iç çarp	$\int fg dx$	44m40
$a_k = \frac{1}{\pi} \int f \cos kx$	Bazda projeksiyon	48m20

30.16 ML Bağlantıları Özeti

💡 7 köprü

1. **Markov** \rightarrow PageRank, MCMC, HMM.
2. **Steady state** \rightarrow Uzun-vadeli dağılım.
3. **Spectral gap** \rightarrow Karışma hızı.
4. **A, A^T** \rightarrow Sol/sağ özvektör.
5. **Fourier** \rightarrow Positional encoding, NeRF, JPEG.
6. **Ortonormal proj** \rightarrow PCA, wavelet.
7. **FFT** \rightarrow Konvolüsyon hızlandırma (FNet).

❗ Tek bir şey alıp gideceksen

Markov $\lambda = 1 =$ steady state (PageRank/MCMC). **Fourier** $a_k = \frac{1}{\pi} \int f \cos kx =$ ortonormal bazda projeksiyon (NeRF, positional encoding, FFT).

31 Quiz 2 İncelemesi

Ortogonalite + Determinant + Özdeğer — üç ayak

Bölüm bilgisi

- **Strang'ın videosu:** [YouTube — Quiz 2 Review](#) (≈48 dk)
- **OCW sayfası:** [MIT 18.06SC — Exam 2 Review](#)
- **Okuma süresi:** ≈35 dk

31.1 Bu Derste Ne Var?

Quiz 2 tekrar oturumu. Kapsam: Chapter 4 (ortogonalite), 5 (determinant), 6.1–6.2 (özdeğer/diagonalizasyon). **ODE hariç.**

1. **Projeksiyon** — matris, özdeğerler $\{0, 1\}$, idempotent.
2. **Least squares** — iki resim.
3. **Gram-Schmidt.**
4. **Özdeğer hüneryleri** — det/trace, kompleks, periyodiklik, recurrence.

“The magnitude of lambda — that’s the key point for stability.” — Strang, 36:22

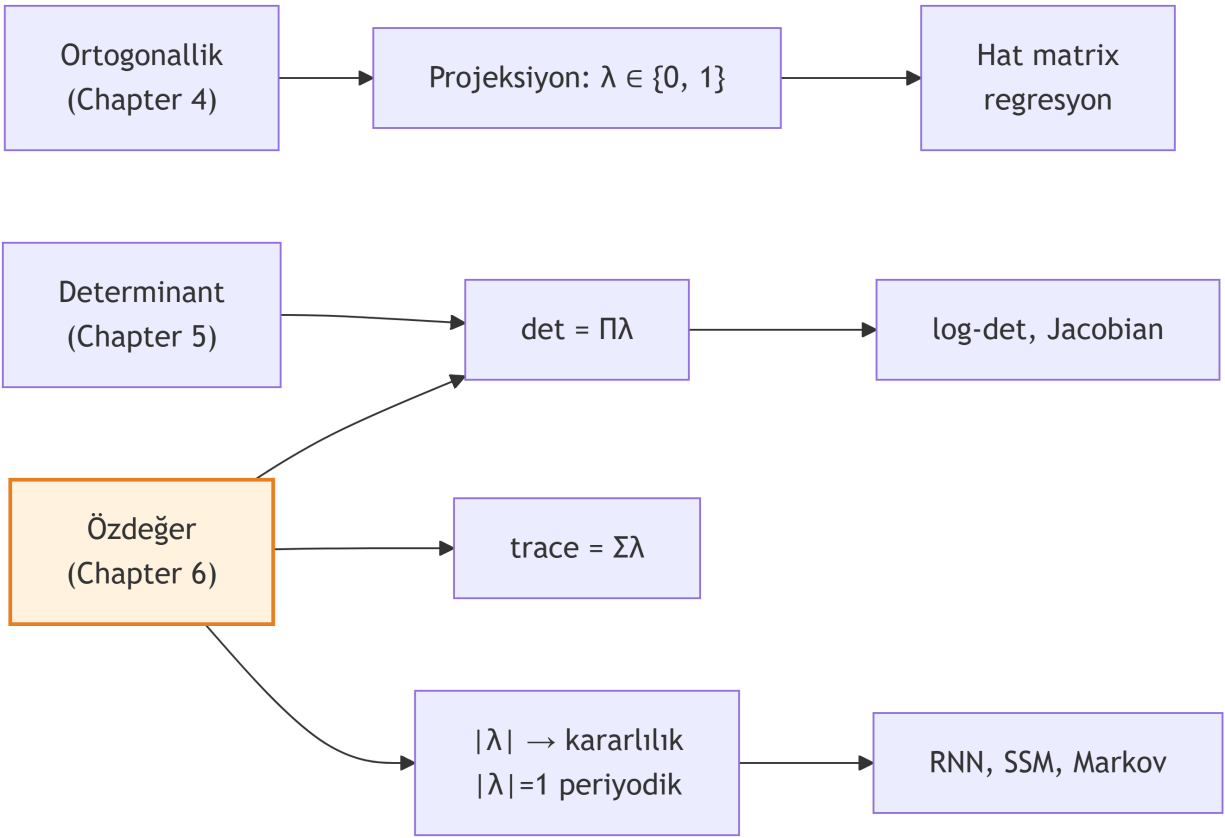
Builder Notu — Trace Kontrolü Disiplini

Strang'ın altın kuralı: özdeğer hesapladığında **her zaman trace ile doğrula** ($\sum \lambda =$ köşegen toplamı).
Bedava sigorta — ML sayısal sonuçlarını sanity-check etmek için aynı disiplin.

31.2 Kapsam

- **Ch 4:** Ortonormal ($Q^T Q = I$), projeksiyon, LS, Gram-Schmidt.
- **Ch 5:** Determinant (3 + 7 özellik), büyük formül, kofaktör, A^{-1} formülü.
- **Ch 6.1–6.2:** $Ax = \lambda x$, karakteristik, $A = S\Lambda S^{-1}$, A^k .

Kapsam dışı: ODE (6.3 — Quiz 3).



Şekil 31.1: Üç ayak birbirine bağlı: P özdeğerleri (ortogonallik + özdeğer), det/trace (det + özdeğer), $|\lambda|$ (dinamik).

31.3 Projeksiyon Özdeğerleri

$\mathbf{a} = (2, 1, 2)^T$ doğrusu projeksiyonu:

$$P = \frac{\mathbf{a}\mathbf{a}^T}{\mathbf{a}^T\mathbf{a}} = \frac{1}{9} \begin{pmatrix} 4 & 2 & 4 \\ 2 & 1 & 2 \\ 4 & 2 & 4 \end{pmatrix}$$

Özdeğerler — hesaplamadan:

- Rank 1 \rightarrow 2 boyutlu null $\rightarrow \lambda = 0, 0$.
- trace = $9/9 = 1 \rightarrow$ **üçüncü** $\lambda = 1$.

Genel kural: projeksiyon özdeğerleri **her zaman** $\{0, 1\}$ (idempotent imzası).

31.4 P + Fark Denklemi

$\mathbf{u}_{k+1} = P\mathbf{u}_k, \mathbf{u}_0 = (9, 9, 0)$:

$$\mathbf{u}_1 = \frac{27}{9}\mathbf{a} = (6, 3, 6)$$

\mathbf{u}_1 zaten kolon uzayında $\rightarrow P^2 = P \rightarrow \mathbf{u}_k = (6, 3, 6) \forall k \geq 1$.

31.5 Least Squares — İki Resim

$y = Dt$ (orijinden) ile $(1, 4), (2, 5), (3, 8)$.

$A = (1, 2, 3)^T, \mathbf{b} = (4, 5, 8), 14\hat{D} = 38 \rightarrow \hat{D} = 19/7$.

Resim 1 (ty): noktalar + doğru. **Resim 2 (vektör):** \mathbf{b} \mathbb{R}^3 'te, $\mathbf{a} = (1, 2, 3)$ doğrusuna izdüşer.

31.6 Gram-Schmidt

$\mathbf{a}_1 = (1, 2, 3), \mathbf{a}_2 = (1, 1, 1)$.

$$\mathbf{B} = \mathbf{a}_2 - \frac{6}{14}\mathbf{a}_1 = \frac{2}{14}(4, 1, -2)$$

$\mathbf{A} \cdot \mathbf{B} = 4 + 2 - 6 = 0 \checkmark$.

31.7 Özdeğer Cebri

$\lambda_1, \dots, \lambda_4$ olan 4×4 :

- **Tersinir** $\Leftrightarrow \lambda_i \neq 0$.
- $\det(A^{-1}) = 1/\det A = 1/\prod \lambda_i$.
- $\text{trace}(A + I) = \sum \lambda_i + n$ ($A + cI$ özdeğeri $\lambda + c$).

31.8 Tridiagonal Recurrence ve Kompleks Özdeğer

Köşegen 1, komşular 1 $\rightarrow D_n = D_{n-1} - D_{n-2}$.

Companion matris $\begin{pmatrix} 1 & -1 \\ 1 & 0 \end{pmatrix} \rightarrow \lambda^2 - \lambda + 1 = 0 \rightarrow \lambda = \frac{1 \pm \sqrt{3}i}{2}$.

$|\lambda|^2 = 1/4 + 3/4 = 1 \rightarrow |\lambda| = 1 \rightarrow$ birim çember, $\lambda = e^{\pm i\pi/3} \rightarrow$ **periyot 6**.

$$A^6 = I, \quad D_n : 1, 0, -1, -1, 0, 1, \text{tekrar}$$

D_n dizisi (periyot 6): $[1, 0, -1, -1, 0, 1, 1, 0]$

31.9 Trace Kontrolü — Strang'in Disiplini

$$A_3 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 2 \\ 0 & 2 & 0 \end{pmatrix}. \det(A_3 - \lambda I) = -\lambda(\lambda^2 - 5) = 0 \rightarrow \lambda = 0, \sqrt{5}, -\sqrt{5}.$$

Trace kontrolü: $\sum \lambda = 0 =$ köşegen toplamı $0 \checkmark$.

"I would never write down those three eigenvalues without checking the trace." — Strang, 44:17

31.10 Bu Tekrarın Özeti

1. **Kapsam** (Ch 4-6.2, ODE hariç).
2. **Projeksiyon** $\{0, 1\}$.
3. **P idempotent fark denk.** sabit.
4. **Least squares iki resim.**
5. **Gram-Schmidt B formülü.**
6. **Özdeğer cebri** ($A^{-1}, A + cI$).
7. **Tridiagonal recurrence.**
8. **Kompleks** $|\lambda| = 1$ **periyodik.**
9. **Bağımlı kolon** \rightarrow **singular.**
10. **Trace kontrolü.**

! Tek bir cümle

Üç ayak birbirine bağlı: projeksiyon $\{0, 1\}$, $\det = \prod \lambda$, $\text{trace} = \sum \lambda$, $|\lambda|$ dinamik kader. **Trace kontrolü** her özdeğer hesabının bedava sigortası.

31.11 Kontrol Soruları

i Soru 1: $\mathbf{a} = (1, 0, 1)$ için $P +$ özdeğer.

$$P = \frac{1}{2} \begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}.$$

Rank 1 $\rightarrow \lambda = 0, 0, 1$. $\text{trace} = 1 \checkmark$. $\lambda = 1$ özvektörü = \mathbf{a} .

i Soru 2: $(1,1),(2,3),(3,4)$ 'e $y = Dt$ fit.

$$A^T A = 14, A^T \mathbf{b} = 19 \rightarrow \hat{D} = 19/14.$$

i Soru 3: $\lambda = 2, 3, -1 \rightarrow \det, \text{trace}, \det(A^{-1}), \text{trace}(A - 2I)?$

- $\det = -6$.
- $\text{trace} = 4$.
- $\det(A^{-1}) = -1/6$.
- $\text{trace}(A - 2I) = (0, 1, -3) \rightarrow -2$.

i Soru 4: Üç bölüm ML'de hangi araçlara karşılık?

Ortogonalite: projeksiyon \rightarrow regresyon (hat matrix); QR \rightarrow kararlı LS, PCA, ortogonal başlatma.

Determinant: $|\det J| \rightarrow$ Jacobian (normalizing flows); log-det \rightarrow Gaussian likelihood, entropi; $\det = 0 \rightarrow$ tekil teşhis.

Özdeğer: PCA (kovaryans), Markov/PageRank ($\lambda = 1$), RNN/SSM kararlılık ($|\lambda|$), log-det = $\sum \log \lambda$.

Birleşim \rightarrow SVD (Ders 29) — üçü birden.

31.12 Egzersizler

Egzersiz 1. $\mathbf{a} = (2, 2, 1) \rightarrow P$, özdeğer, trace doğrulama.

Egzersiz 2. $(1, 2), (2, 3), (3, 5) \rightarrow y = C + Dt$ (sabitli) \rightarrow normal denklemler.

Egzersiz 3. $\mathbf{a}_1 = (1, 1, 0), \mathbf{a}_2 = (1, 0, 1) \rightarrow$ Gram-Schmidt.

Egzersiz 4. (Python) Quiz konuları tara.

Egzersiz 5. İspatla: Projeksiyon $\lambda \in \{0, 1\}$. (İpucu: $P^2 = P \rightarrow \lambda^2 = \lambda$.) Ders 25 (simetrik) habercisi.

31.13 Sonraki Ders İçin Hazırlık

Ders 25: Simetrik Matrisler ve Pozitif Tanımlılık

- **Spektral teorem:** simetrik \rightarrow gerçel λ + ortogonal özvektör; $A = Q\Lambda Q^T$.
- **Pozitif tanım:** $\lambda > 0$, pivot > 0 .
- Kovaryans, Gram, kernel — hepsi simetrik.

⚠ Ders 25 öncesi

- Egzersiz 5.
- Ders 14-24 + bu tekrar gözden geçir.

31.14 Sınav Formülleri (Cheat Sheet)

Soru tipi	Anahtar	Strang'da
Projeksiyon P	$P = \mathbf{a}\mathbf{a}^T / \mathbf{a}^T \mathbf{a}$; $\lambda \in \{0, 1\}$	4m35
$\lambda = 1$ özvektörü	\mathbf{a} ($P\mathbf{a} = \mathbf{a}$)	6m39
$P^2 = P$ fark denk.	İlk adımdan sonra sabit	9m22
Least squares	$A^T A \hat{\mathbf{x}} = A^T \mathbf{b}$	15m01
Gram-Schmidt	$\mathbf{B} = \mathbf{a}_2 - \frac{\mathbf{a}_1^T \mathbf{a}_2}{\mathbf{a}_1^T \mathbf{a}_1} \mathbf{a}_1$	21m14
det/trace	$\Pi\lambda, \Sigma\lambda$	24m38
Recurrence	$D_n = D_{n-1} - D_{n-2}$	28m14
$ \lambda = 1$	Periyodik	35m58
Trace kontrolü	Bedava doğrulama	44m17

31.15 ML Bağlantıları Özeti

💡 7 köprü

1. $P \in \{0, 1\} \rightarrow$ Hat matrix, regresyon.
2. **LS = projeksiyon** \rightarrow Lineer regresyon iki resim.
3. **GS/QR** \rightarrow Kararlı regresyon, ortogonal başlatma.
4. **det/trace** = $\Pi\lambda / \Sigma\lambda \rightarrow$ Log-det, iz regularization.
5. $|\lambda| \rightarrow$ RNN/SSM kararlılık.
6. **Trace kontrolü** \rightarrow Sanity-check disiplini.
7. **Üç bölüm** \rightarrow **SVD** (Ders 29) birleşim.

! Tek bir Őey alıp gideceksen

Üç ayak birbirine baęlı: projeksiyon $\{0, 1\}$, $\det = \prod \lambda$, $\text{trace} = \sum \lambda$, $|\lambda|$ dinamik kader. **Trace kontrolü** bedava sigorta.

32 Simetrik Matrisler ve Pozitif Tanımlılık

Spektral teorem — $A = Q\Lambda Q^T$, gerçel λ , ortogonal q

Bölüm bilgisi

- **Strang'ın videosu:** [YouTube — Lecture 25: Symmetric Matrices and Positive Definiteness](#) (≈44 dk)
- **OCW sayfası:** [MIT 18.06SC — Lecture 25](#)
- **Okuma süresi:** ≈40 dk

32.1 Bu Derste Ne Var?

Matrislerin en sağlam sınıfı: **simetrik** ($A = A^T$).

1. **İki gerçek:** özdeğerler **gerçel**, özvektörler **dik**.
2. **Spektral teorem:** $A = Q\Lambda Q^T$ (Q ortogonal, Λ gerçel köşegen).
3. **Spektral ayrışım:** $A = \sum \lambda_i \mathbf{q}_i \mathbf{q}_i^T$.
4. **Pozitif tanım:** tüm $\lambda > 0 \iff$ tüm pivot $> 0 \iff$ tüm sol-üst alt-det > 0 .

“The eigenvalues are real and the eigenvectors are perpendicular.” — Strang, 1:09

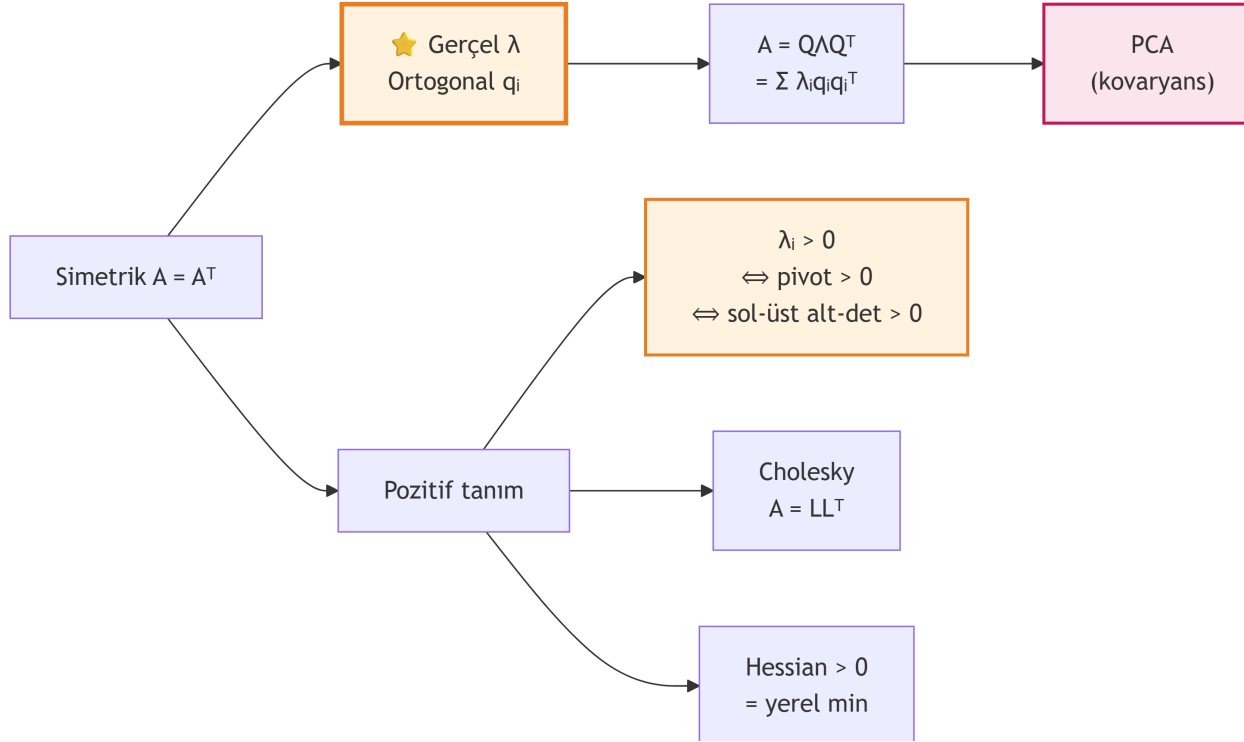
Builder Notu — Simetrik = ML'in Sağlam Zemini

- **Spektral teorem** → PCA: kovaryans simetrik → gerçel + ortogonal → ana bileşenler.
- **Pozitif tanım** → kovaryans, kernel (**Gram**), **Hessian** (konveks optimizasyon).
- **Cholesky** ($A = LL^T$) sadece pozitif tanım için — Gaussian örnekleme, GP regresyon.
- **Pivot-özdeğer işaret eşitliği** → Hessian'da saddle point teşhisi (Sylvester inertia).

32.2 İki Ana Gerçek

1. **Özdeğerler gerçel.**
2. **Özvektörler dik** (ortonormal seçilebilir).

Farklı özdeğerlerde otomatik dik; tekrarlı özdeğerde özuzaydan dik baz seçilir. **Simetrik matris asla degenerate olmaz** — her zaman tam özvektör seti.



Şekil 32.1: Simetrik $\rightarrow A = Q\Lambda Q^T$ (spektral teorem). Pozitif tanım $\rightarrow \lambda$, pivot, alt-det > 0 ; PCA, Cholesky, Hessian.

32.3 Spektral Teorem — $A = Q\Lambda Q^T$

Genel $A = SAS^{-1}$ 'de $S = Q$ (ortogonal):

$$A = Q\Lambda Q^T, \quad Q^T Q = I, \quad Q^{-1} = Q^T$$

Simetri açık: $(Q\Lambda Q^T)^T = Q\Lambda^T Q^T = Q\Lambda Q^T = A$.

32.4 Gerçel Özdeğer İspatı

$Ax = \lambda x$. Eşlenik + transpoze:

$$\bar{x}^T A^T = \bar{x}^T \bar{\lambda}$$

Simetri ($A^T = A$):

$$\bar{x}^T A = \bar{x}^T \bar{\lambda}$$

İlk denklemi \bar{x}^T ile, ikinciye x ile çarp \rightarrow sol taraflar aynı:

$$\lambda \bar{\mathbf{x}}^T \mathbf{x} = \bar{\lambda} \bar{\mathbf{x}}^T \mathbf{x}$$

$$\bar{\mathbf{x}}^T \mathbf{x} = \sum |x_i|^2 > 0 \rightarrow \lambda = \bar{\lambda} \rightarrow \text{gerçel.}$$

Kompleks “iyi” matris: $\bar{A}^T = A$ (**Hermitian**). Gerçek durumda = simetrik.

32.5 Spektral Ayrışım

$A = Q\Lambda Q^T$, yi kolon \times satır:

$$A = \sum_i \lambda_i \mathbf{q}_i \mathbf{q}_i^T$$

Her $\mathbf{q}_i \mathbf{q}_i^T = \text{dik projeksiyon matrisi}$ (birim vektör, $\mathbf{q}_i^T \mathbf{q}_i = 1 \rightarrow P^2 = P$).

“Every symmetric matrix is a combination of mutually perpendicular projection matrices.” — Strang, 27:57

Builder Notu: Bu ayrışım PCA’nın özü — veri = ana bileşen projeksiyonlarının ağırlıklı toplamı, ağırlıklar = varyanslar (özdeğerler).

32.6 Pivot İşareti = Özdeğer İşareti

Sylvester’s law of inertia:

$$\#\{\text{pozitif pivot}\} = \#\{\text{pozitif özdeğer}\}$$

50 \times 50 matris için 50 özdeğer hesaplamak pahalı ve kararsız; pivotlar hızlı ve kararlı.

Kuşatma: $A - 7I$ ’nin pivotları \rightarrow kaç özdeğer 7’nin üstünde.

Builder Notu: **Hessian’da saddle point** ucuz teşhisi — LDL^T köşegen işaretleri.

32.7 Pozitif Tanım — Tanım ve Testler

Tanım: simetrik + tüm $\lambda > 0$.

Eşdeğer testler:

Test	Koşul
Özdeğerler	hepsi > 0
Pivotlar	hepsi > 0
Sol-üst alt-determinantlar	hepsi > 0

32.8 Örnek — $A = ((2, 1), (1, 2))$

Simetrik ✓. Pivot 2, ikinci pivot 3/2 (det = 3). $\lambda = 1, 3$. Sol-üst alt-det: 2, 3. **Pozitif tanımlı** ✓.

Karşı-örnek: $\begin{pmatrix} -1 & 3 \\ 3 & 2 \end{pmatrix}$ — sol-üst $1 \times 1 = -1 < 0 \rightarrow$ pozitif tanımlı **değil** (büyük det olsa bile).

“This fails the test because that minus one is negative.” — Strang, 42:12

Determinant tek başına yetmez — TÜM alt-determinantlar gerekir.

```
özdeğerler: [1. 3.]
sol-üst alt-det 1x1: 2.0
sol-üst alt-det 2x2: 2.9999999999999996
pozitif tanımlı mı? True
L (Cholesky):
[[1.41421356 0.          ]
 [0.70710678 1.22474487]]
L LT = A? True
```

32.9 Kursun Birleştiği Nokta

Üç ana araç simetrik matriste buluşur:

- **Pivotlar** (Ders 2 — elimination)
- **Determinantlar** (Ch 5)
- **Özdeğerler** (Ch 6)

Simetrik (özellikle PD) matriste her birinin işareti diğerlerini söyler. SVD (Ders 29) bunu $n \times n$ simetrik-olmayana genelleştirecek.

32.10 Bu Dersin Özeti

1. **Simetrik:** gerçel λ + dik q .
2. $A = Q\Lambda Q^T$.
3. Gerçel özdeğer ispatı (eşlenik + simetri).
4. **Spektral ayrışım** = $\sum \lambda_i \mathbf{q}_i \mathbf{q}_i^T$.
5. **Pivot işareti = özdeğer işareti.**
6. **Pozitif tanım** üç eşdeğer test.
7. Determinant tek başına yetmez.

! Tek bir cümle

Simetrik matrisler gerçel özdeğer + ortogonal özvektör garantiler ($A = Q\Lambda Q^T$); pozitif tanımlı olanlarda tüm λ , pivot, sol-üst alt-det > 0 — PCA, Cholesky, Hessian, kernel methods'un sağlam temeli.

32.11 Kontrol Soruları

i Soru 1: $A = ((2,1),(1,2))$ pozitif tanımlı mı?

Pivot 2, $3/2 > 0$. $\lambda = 1, 3 > 0$. Alt-det 2, $3 > 0$. ✓ **PD**.

i Soru 2: $A = 4q_1q_1^T + q_2q_2^T \rightarrow \lambda, \det$?

$\lambda = 4, 1$. $\det = 4$. $\text{trace} = 5$.

i Soru 3: 50×50 simetrik, 30 + pivot, 20 -. Kaç + λ ? PD mi?

30 +, 20 -. PD **değil** (negatif λ var).

i Soru 4: PCA ve Cholesky simetrik/PD ile.

PCA: kovaryans simetrik $\rightarrow A = Q\Lambda Q^T$, Q ana bileşenler, Λ varyanslar.

Cholesky ($A = LL^T$) sadece PD için — Gaussian örnekleme $\mathbf{x} = L\mathbf{z}$, GP regresyonu, lineer sistem çözümü.

32.12 Egzersizler

Egzersiz 1. $A = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$ PD doğrula (pivot + λ + alt-det).

Egzersiz 2. Spektral ayrışımından λ , det.

Egzersiz 3. Pivot işaret sayımıyla özdeğer sayısı.

Egzersiz 4. (Python) `eigvalsh`, `cholesky`, alt-det.

Egzersiz 5. *İspatla*: Simetrik + farklı özdeğerler \rightarrow özvektörler dik. (İpucu: $\mathbf{x}^T A \mathbf{y}$ 'yi iki yoldan hesapla.)

32.13 Sonraki Ders İçin Hazırlık

Ders 26: Kompleks Matrisler ve FFT

- Hermitian, üniter.
- Fourier matrisi F_n , FFT $O(n \log n)$.

⚠ Ders 26 öncesi

- Egzersiz 5 (ortogonal özvektör ispatı).
- `eigvalsh` ile simetrik matris dene.

32.14 Anahtar Kavramlar (Cheat Sheet)

Kavram	Tanım	Strang'da
Simetrik	$A = A^T$	0m30
İki gerçek	Gerçek λ + dik q	1m09
Spektral teorem	$A = Q\Lambda Q^T$	8m08
Spektral ayrışım	$\sum \lambda_i \mathbf{q}_i \mathbf{q}_i^T$	27m57
Gerçek ispatı	$\bar{\mathbf{x}}^T \mathbf{x} > 0$	12m29
Hermitian	$\bar{A}^T = A$	24m20
Pivot-λ işareti	Sylvester inertia	32m58
Pozitif tanım	$\lambda > 0 \iff \text{pivot} > 0 \iff$ alt-det > 0	36m20

32.15 ML Bağlantıları Özeti

💡 7 köprü

1. **PCA = spektral teorem** → Kovaryans → ana bileşenler.
2. **Pozitif tanım = konveks optim** → Hessian PD = yerel min.
3. **Cholesky** $A = LL^T$ → Gaussian örnekleme, GP regresyon.
4. **Kernel matrisler PD** → Kernel PCA, spektral clustering, GP.
5. **Pivot işaret testi** → Saddle point teşhisi (LDL^T inertia).
6. **Hermitian / üniter** → Kuantum, kompleks ağlar, FFT.
7. **Stable training** → Simetrik kısıtlar; degenerate olmaz.

❗ Tek bir şey alıp gideceksen

Simetrik → gerçek λ + dik q ; $A = Q\Lambda Q^T$ (spektral). PD: tüm $\lambda > 0$ (= pivot > 0 = sol-üst alt-det > 0).
PCA, Cholesky, Hessian, kernel — ML'in sağlam zemini.

33 Kompleks Matrisler ve FFT

Hermitian, unitary, Fourier matrisi → $n \log n$ hızlandırma

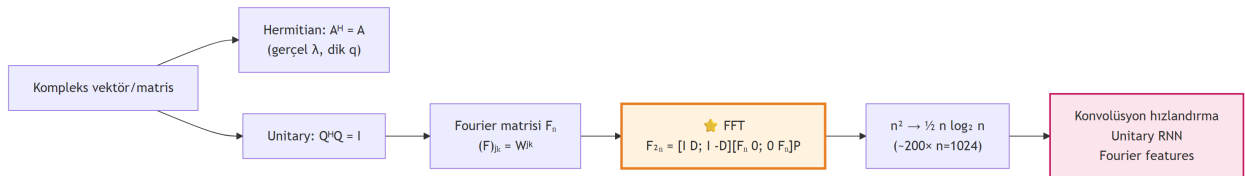
i Bölüm bilgisi

- **Strang'ın videosu:** [YouTube — Lecture 26: Complex Matrices; FFT](#) (≈48 dk)
- **OCW sayfası:** [MIT 18.06SC — Lecture 26](#)
- **Okuma süresi:** ≈40 dk

33.1 Bu Derste Ne Var?

1. **Kompleks uzunluk:** $\mathbf{z}^H \mathbf{z}$ ($\mathbf{z}^H = \bar{\mathbf{z}}^T$).
2. **Hermitian** ($A^H = A$) — simetri kompleks karşılığı; **unitary** ($Q^H Q = I$) — ortogonal kompleks karşılığı.
3. **Fourier matrisi** F_n : $(F_n)_{jk} = W^{jk}$, $W = e^{2\pi i/n}$.
4. **FFT:** n^2 yerine $\frac{1}{2}n \log_2 n$.

“FFT multiplies by an $n \times n$ matrix in one half $n \log n$ steps.” — Strang, 45:07



Şekil 33.1: Kompleks → Hermitian/unitary → Fourier → FFT ($n^2 \rightarrow n \log n$).

💡 Builder Notu — FFT ML’ in Sessiz Devi

- **Konvolüsyon teoremi** → büyük çekirdek konvolüsyon FFT ile $O(n \log n)$; WaveNet, FNO, diffusion.
- **Unitary RNN / ortogonal başlatma** — $|\lambda| = 1 \rightarrow$ gradyan ne patlar ne söner.
- **Fourier features** — NeRF positional encoding, FNO PDE çözücüler.
- **Divide-and-conquer** deseni → Strassen, FlashAttention bloklama.

33.2 Kompleks Uzunluk — $\mathbf{z}^H \mathbf{z}$

$\mathbf{z}^T \mathbf{z}$ yanlış: $\mathbf{z} = (1, i) \rightarrow 1 + i^2 = 0$ (saçma).

Doğrusu:

$$\mathbf{z}^H \mathbf{z} = \bar{z}_1 z_1 + \dots = \sum |z_i|^2 > 0$$

$\mathbf{z} = (1, i) \rightarrow 1 + 1 = 2$, uzunluk $\sqrt{2}$.

33.3 Hermitian Matris — $A^H = A$

$$A = \begin{pmatrix} 2 & 3+i \\ 3-i & 5 \end{pmatrix}$$

- Köşegen gerçel.
- Köşegen-dışı eşlenik çiftler.

Spektral teorem aynen: **gerçel özdeğer + dik özvektör.**

33.4 Unitary Matris — $Q^H Q = I$

Ortogonalin kompleks karşılığı. $Q^{-1} = Q^H$.

33.5 Fourier Matrisi F_n

$$(F_n)_{jk} = W^{jk}, \quad W = e^{2\pi i/n}, \quad W^n = 1$$

F_4 örneği ($W = i$):

$$F_4 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix}$$

F4 =

```
[[ 1.+0.j 1.+0.j 1.+0.j 1.+0.j]
 [ 1.+0.j 0.+1.j -1.+0.j -0.-1.j]
 [ 1.+0.j -1.+0.j 1.-0.j -1.+0.j]
 [ 1.+0.j -0.-1.j -1.+0.j 0.+1.j]]
```

F4^H F4 / n =

```
[[ 1.+0.j -0.+0.j 0.+0.j 0.+0.j]]
```

$$\begin{bmatrix} -0.-0.j & 1.+0.j & -0.+0.j & 0.+0.j \\ 0.-0.j & -0.-0.j & 1.+0.j & -0.+0.j \\ 0.-0.j & 0.-0.j & -0.-0.j & 1.+0.j \end{bmatrix}$$

33.6 Ters Matris

Kolonlar \sqrt{n} uzunluğunda, Hermitian iç çarpımda dik $\rightarrow \frac{1}{\sqrt{n}}F_n$ unitary:

$$F_n^{-1} = \frac{1}{n}F_n^H$$

Ters dönüşüm bedava.

33.7 FFT Fikri — F_{2n} ile F_n

$(W_{2n})^2 = W_n \rightarrow F_{2n}$ iki F_n 'e ayrıştırılabilir:

$$F_{2n} = \begin{pmatrix} I & D \\ I & -D \end{pmatrix} \begin{pmatrix} F_n & 0 \\ 0 & F_n \end{pmatrix} P$$

- P : çift-tek ayırma permütasyonu.
- **Orta**: iki bağımsız F_n (köşegen blok).
- $D = \mathbf{diag}(1, W, W^2, \dots, W^{n-1})$ — düzeltme.

Özyinelemeli: her F_n tekrar ikiye böl \rightarrow 1-noktaya iner.

33.8 $n^2 \rightarrow \frac{1}{2}n \log_2 n$

$\log_2 n$ adım, her birinde $\sim n/2$ çarpma:

$$\text{FFT maliyeti} = \frac{1}{2}n \log_2 n$$

n	Naif (n^2)	FFT
1024	$\sim 10^6$	5,120
Oran	—	$\sim 200\times$

“We’ve reduced the calculation by a factor of 200 just by factoring the matrix properly.” — Strang, 46:25

33.9 Bu Dersin Özeti

1. $\mathbf{z}^H \mathbf{z} = \sum |z_i|^2$.
2. **Hermitian iç çarpım** $\mathbf{y}^H \mathbf{x}$.
3. **Hermitian** $A^H = A$ — gerçel köşegen + eşlenik çiftler.
4. **Unitary** $Q^H Q = I$.
5. $F_n: (F_n)_{jk} = W^{jk}$.
6. $F_n^{-1} = \frac{1}{n} F_n^H$.
7. **FFT ayrıştırması**.
8. $\frac{1}{2} n \log_2 n$.

! Tek bir cümle

Kompleks dünyada “transpoze + eşlenik” (**Hermitian**); **Fourier matrisi** F_n kolonları unitary; **FFT** $F_{2n} = [I \ D/I \ -D][F_n/F_n]P$ özyinelemesi n^2 'den $\frac{1}{2}n \log_2 n$ 'e iner — modern hesaplamının temel hızlandırması.

33.10 Kontrol Soruları

i Soru 1: $\mathbf{z} = (1, i)$ uzunluğu? $\mathbf{z}^T \mathbf{z}$ neden yanlış?

$\mathbf{z}^T \mathbf{z} = 1 - 1 = 0$ (yanlış). $\mathbf{z}^H \mathbf{z} = 2$, uzunluk $\sqrt{2}$.

i Soru 2: Hermitian matrisin köşegeninde ne tür sayılar?

Gerçel — kendi eşleniğine eşit olmalı.

i Soru 3: F_4 'te W kuvvetleri?

$W = i$. Kuvvetler: $i, -1, -i, 1$. Birim çember çeyrek turları.

i Soru 4: $n = 1024$ FFT vs naif?

FFT: 5120. Naif: $\sim 10^6$. Oran $\sim 200\times$.

33.11 Egzersizler

Egzersiz 1. $\mathbf{z} = (2, 1 + i, i)$ uzunluk.

Egzersiz 2. $A = ((1, 2 - i), (2 + i, 3))$ Hermitian mi?

Egzersiz 3. F_2 ($W = -1$) yaz. (Hadamard.)

Egzersiz 4. (Python) `np.fft.fft` ile çarpım karşılaştır.

Egzersiz 5. İspatla: Unitary Q uzunluk korur ($\|Q\mathbf{z}\| = \|\mathbf{z}\|$). (İpucu: $\mathbf{z}^H Q^H Q \mathbf{z}$.)

33.12 Sonraki Ders İçin Hazırlık

Ders 27: Pozitif Tanımlı Matrisler ve Minimumlar

- **Enerji testi:** $\mathbf{x}^T A \mathbf{x} > 0$.
- Optimizasyon, eliptik geometri.

⚠ Ders 27 öncesi

- Egzersiz 5 (uzunluk koruma).
- fft ile dene.

33.13 Anahtar Kavramlar (Cheat Sheet)

Kavram	Tanım
Kompleks uzunluk	$\mathbf{z}^H \mathbf{z} = \sum z_i ^2$
Hermitian iç çarpım	$\mathbf{y}^H \mathbf{x}$
Hermitian matris	$A^H = A$, gerçel köşegen
Unitary	$Q^H Q = I$, $Q^{-1} = Q^H$
Fourier matrisi	$(F_n)_{jk} = W^{jk}$, $W^n = 1$
$F_4 W$	i
Ters	$F_n^{-1} = \frac{1}{n} F_n^H$
FFT ayrışım	$[I D/I - D][F_n/F_n]P$
Maliyet	$\frac{1}{2}n \log_2 n$

33.14 ML Bağlantıları Özeti

💡 5 köprü

1. **FFT = konvolüsyon hızlandırma** → WaveNet, FNO, diffusion.
2. **Fourier features** → NeRF positional encoding, FNO.
3. **Unitary RNN / ortogonal başlatma** → Gradyan kararlılığı.
4. **Kompleks-değerli ağlar** → MRI, radar, ses; Wirtinger calculus.
5. **Divide-and-conquer** → Strassen matris çarpımı, FlashAttention bloklama.

! Tek bir şey alıp gideceksen

“Transpoze + eşlenik” (Hermitian) — kompleks dünyanın kuralı. **FFT** $n^2 \rightarrow \frac{1}{2}n \log_2 n$ — modern hesaplamının temel hızlandırması.

34 Pozitif Tanımlı Matrisler ve Minimumlar

$\mathbf{x}^T \mathbf{A} \mathbf{x} > 0$ — enerji, Hessian, ellipsoid

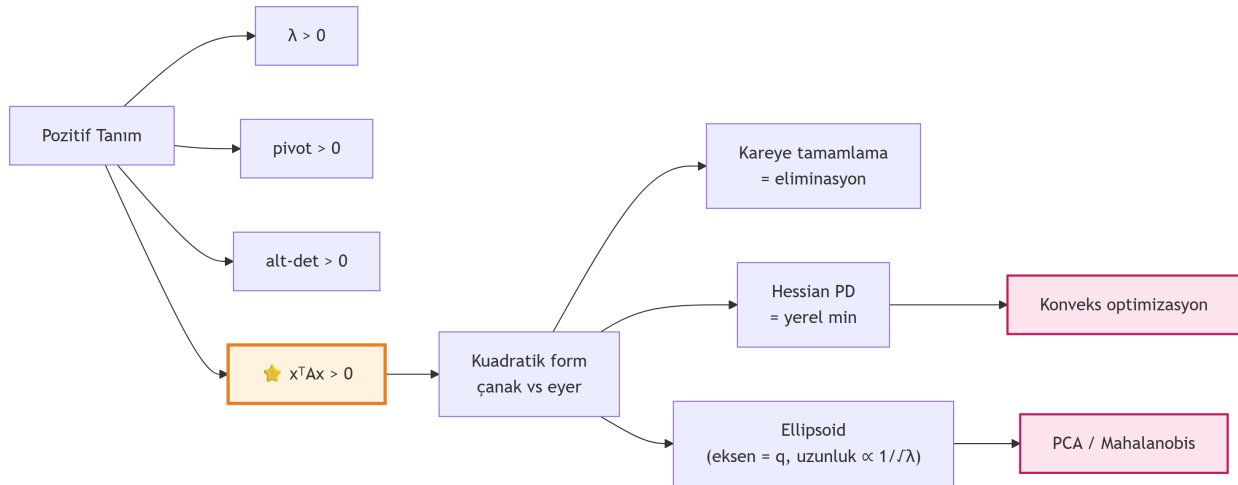
i Bölüm bilgisi

- Strang'ın videosu: [YouTube — Lecture 27: Positive Definite Matrices and Minima](#) (≈50 dk)
- OCW sayfası: [MIT 18.06SC — Lecture 27](#)
- Okuma süresi: ≈40 dk

34.1 Bu Derste Ne Var?

1. **Dört eşdeğer test:** $\lambda > 0$, alt-det > 0 , pivot > 0 , $\mathbf{x}^T \mathbf{A} \mathbf{x} > 0$.
2. **Kuadratik form** — çanak vs eyer.
3. **Kareye tamamlama = eliminasyon.**
4. **Hessian PD → minimum; ellipsoid = ana eksen teoremi.**

“Positive for a number translates into positive definite for a matrix.” — Strang, 26:29



Şekil 34.1: PD'nin dört testi → minimum/eyer ayrımı → Hessian → konveks optimizasyon.

💡 Builder Notu — Optimizasyon Kalbi

- $\mathbf{x}^T A \mathbf{x} = \text{enerji} = \text{loss}$. Min \iff Hessian PD.
- **Eyer noktaları** (negatif λ) = derin öğrenmenin gerçek düşmanı (gradyan = 0 ama min değil).
- **Kareye tamamlama = Cholesky** ($A = LL^T$).
- **Ellipsoid** = Mahalanobis konturu = **PCA yönleri**.

34.2 Dört Eşdeğer Test

Test	Koşul
Özdeğerler	hepsi > 0
Sol-üst alt-det	hepsi > 0
Pivotlar	hepsi > 0
Enerji	$\mathbf{x}^T A \mathbf{x} > 0 \forall \mathbf{x} \neq \mathbf{0}$

34.3 Kuadratik Form

$$A = \begin{pmatrix} a & b \\ b & c \end{pmatrix};$$

$$\mathbf{x}^T A \mathbf{x} = ax^2 + 2bxy + cy^2$$

Sınıflar:

- $\det > 0, a > 0 \rightarrow$ **çanak (minimum)** \rightarrow PD.
- $\det = 0 \rightarrow$ **yarı-tanım** ($\lambda = 0$ var).
- $\det < 0 \rightarrow$ **eyer** (negatif λ).

34.4 Üç Örnek

Örnek 1 — $A = \begin{pmatrix} 2 & 6 \\ 6 & 18 \end{pmatrix}$ (**yarı-tanım**): $\det = 0; \lambda = 0, 20$.

Örnek 2 — $\begin{pmatrix} 2 & 6 \\ 6 & 7 \end{pmatrix}$ (**eyer**): $\det = -22 < 0$. $\mathbf{x} = (1, -1) \rightarrow 2 - 12 + 7 = -3 < 0$.

Örnek 3 — $\begin{pmatrix} 2 & 6 \\ 6 & 20 \end{pmatrix}$ (**PD**): $\det = 4 > 0; \lambda$ ikisi pozitif; pivot 2, 2.

34.5 Kareye Tamamlama = Eliminasyon

$$2x^2 + 12xy + 20y^2 = 2(x + 3y)^2 + 2y^2$$

Pivotlar (2, 2) karelerin dışında; **multiplier (3)** içeride.

“Completing the square is elimination. The pivots outside, the multiplier inside.” — Strang, 33:05

Pozitif pivot → karelerin pozitif toplamı → her yerde pozitif.

34.6 Hessian ve Minimum

Çok değişkende minimum:

- Gradyan = 0 (kritik nokta).
- **Hessian PD** (eğrilik her yönde yukarı).

$$H = \begin{pmatrix} f_{xx} & f_{xy} \\ f_{xy} & f_{yy} \end{pmatrix}$$

Calculus testi $f_{xx}f_{yy} - f_{xy}^2 > 0 = 2 \times 2$ det.

Builder Notu: Deep learning’de Hessian çoğunlukla PD **değil** (eyer her yerde). SGD gürültüsü + momentum + Adam eyerden kaçır.

34.7 Strang’in 3×3 Favori

$$A = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}$$

Alt-det: 2, 3, 4. Pivot: 2, 3/2, 4/3. λ : $2 - \sqrt{2}$, 2, $2 + \sqrt{2}$. **PD.**

özdeğerler: [0.58578644 2. 3.41421356]

alt-det 1×1: 2.0

alt-det 2×2: 2.9999999999999996

alt-det 3×3: 4.0

Cholesky L:

$$\begin{bmatrix} 1.41421356 & 0. & 0. & \\ -0.70710678 & 1.22474487 & 0. & \\ 0. & -0.81649658 & 1.15470054 & \end{bmatrix}$$

34.8 Geometri — Ellipsoid

$\mathbf{x}^T A \mathbf{x} = 1 \rightarrow 2 \times 2$ elips, 3×3 ellipsoid.

Ana eksen teoremi ($A = Q \Lambda Q^T$):

- **Yönler:** özvektörler (Q).
- **Uzunluklar:** $\propto 1/\sqrt{\lambda_i}$.

Büyük $\lambda \rightarrow$ kısa eksen.

Builder Notu: Mahalanobis uzaklığı $\mathbf{x}^T \Sigma^{-1} \mathbf{x}$ — sabit kontur = ellipsoid; eksenler kovaryansın özvektörleri = PCA yönleri.

34.9 Bu Dersin Özeti

1. **4 eşdeğer test** (yeni: enerji $\mathbf{x}^T A \mathbf{x}$).
2. **Kuadratik form:** çanak/eyer.
3. **Yarı-tanım** ($\det = 0$), **indefinite** ($\det < 0$).
4. **Kareye tamamlama = eliminasyon.**
5. **Minimum:** gradyan = 0 + Hessian PD.
6. **Geometri:** ellipsoid + ana eksen.

! Tek bir cümle

PD = tüm özdeğer/pivot/alt-det $> 0 = \mathbf{x}^T A \mathbf{x} > 0$. Minimum testi: gradyan = 0 + **Hessian PD**. Geometri: $\mathbf{x}^T A \mathbf{x} = 1$ ellipsoidi, eksenler = q, uzunluk $\propto 1/\sqrt{\lambda}$. Konveks optimizasyon, Mahalanobis, PCA temeli.

34.10 Kontrol Soruları

i Soru 1: 4 eşdeğer test?

$\lambda > 0$, alt-det > 0 , pivot > 0 , $\mathbf{x}^T A \mathbf{x} > 0$. Biri sağlanırsa hepsi.

i Soru 2: ((2,6),(6,18)) niye yarı-tanım?

$\det = 0 \rightarrow \lambda = 0$, 20. $\mathbf{x}^T A \mathbf{x} \geq 0$ (sıfır olabilir).

i Soru 3: Kareye tamamlama niye pozitiflik ispatı?

Kareler ≥ 0 . Pivot katsayıları > 0 ise toplam > 0 (orijin hariç).

i Soru 4: nD minimum testi?

Gradyan = 0 + **Hessian PD** (1D'deki $f'' > 0$ genellemesi).

i Soru 5: Ellipsoid eksenleri.

Yön = özvektör, uzunluk $\propto 1/\sqrt{\lambda}$ (ana eksen teoremi $A = Q\Lambda Q^T$).

34.11 Egzersizler

Egzersiz 1. $A = \text{diag}(3, 5)$ PD mi? $\mathbf{x}^T A \mathbf{x}$.

Egzersiz 2. $((1, 2), (2, 1))$ PD mi? (det + enerji)

Egzersiz 3. Hangi c için $((1, 2), (2, c))$ PD?

Egzersiz 4. (Python) eigvalsh + Cholesky + alt-det.

Egzersiz 5. İspatla: A PD $\rightarrow A^{-1}$ PD. (İpucu: özdeğer $1/\lambda$.)

34.12 Sonraki Ders İçin Hazırlık

Ders 28: Benzer Matrisler ve Jordan Formu

- **Benzer:** $B = M^{-1}AM$ (aynı özdeğer).
- **Jordan formu** — diagonalize edilemeyen matrisler.

⚠ Ders 28 öncesi

- Egzersiz 5 (A^{-1} PD).
- cholesky ile dene.

34.13 Anahtar Kavramlar (Cheat Sheet)

Kavram	Tanım
PD (4 test)	$\lambda > 0 \iff \text{alt-det} > 0 \iff \text{pivot} > 0 \iff \mathbf{x}^T A \mathbf{x} > 0$
Kuadratik form	$ax^2 + 2bxy + cy^2$
Yarı-tamm	$\mathbf{x}^T A \mathbf{x} \geq 0$
İndefinite	$\det < 0$, eyer
Kareye tamamlama	= eliminasyon
Min testi	gradyan = 0 + Hessian PD
Hessian simetrik	$f_{xy} = f_{yx}$

Kavram	Tanım
2D Calculus	$f_{xx}f_{yy} - f_{xy}^2 > 0$
Geometri	Ellipsoid; eksen = q, uzunluk $\propto 1/\sqrt{\lambda}$

34.14 ML Bağlantıları Özeti

💡 5 köprü

1. **Konveks optim = Hessian PD** → Lineer/lojistik reg, SVM garantili min.
2. **Eyer noktaları** → DL'in gerçek düşmanı.
3. **Mahalanobis** → Gaussian şekli, PCA yönleri.
4. **Cholesky** → Gaussian örnekleme, GP regresyon.
5. **Newton, K-FAC** → İkinci-mertebe optim (PD olmazsa damping).

! Tek bir şey alıp gideceksen

PD = $\mathbf{x}^T A \mathbf{x} > 0$ + 3 eşdeğer test. Min = gradyan + Hessian PD. Ellipsoid eksenleri = özvektör (PCA).
Konveks optim, Cholesky, Mahalanobis ortak temeli.

35 Benzer Matrisler ve Jordan Formu

$B = M^{-1}AM$ — aynı özdeğer ailesi

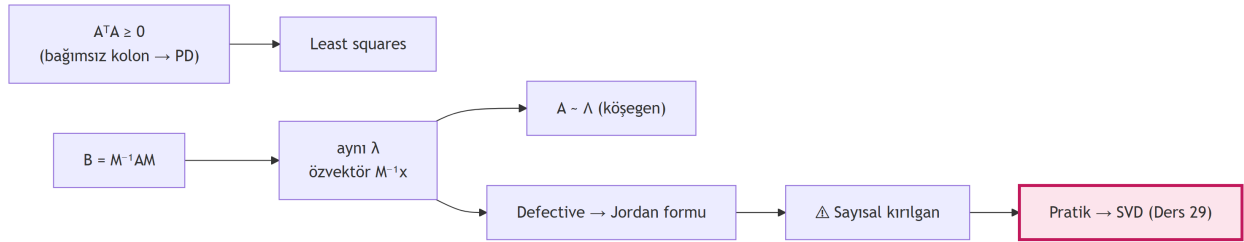
i Bölüm bilgisi

- **Strang'ın videosu:** [YouTube — Lecture 28: Similar Matrices and Jordan Form](#) (≈46 dk)
- **OCW sayfası:** [MIT 18.06SC — Lecture 28](#)
- **Okuma süresi:** ≈38 dk

35.1 Bu Derste Ne Var?

1. $A^T A$ **pozitif (yarı-)tanım** — least squares kalbi.
2. **Benzer:** $B = M^{-1}AM$, aynı özdeğer.
3. **Diagonalize = benzerlik:** $A \sim \Lambda$.
4. **Jordan formu** — defective matrisler için en yakın köşegen.

“*Similar matrices have the same eigenvalues.*” — Strang, 20:31



Şekil 35.1: $A^T A \rightarrow$ least squares; benzerlik aileleri; defective \rightarrow Jordan; pratik \rightarrow SVD.

💡 Builder Notu — Defective = Kırılğan

- $A^T A =$ **Gram matrisi** her yerde: kernel, kovaryans, normal denklemler, QK^T benzeri.
- **Benzerlik = baz değişimi;** özdeğerler **baz-bağımsız** (dinamik sistem içsel özellikleri).
- **Jordan sayısal kötü** \rightarrow en ufak pertürbasyon yapıyı bozar; pratik = **SVD** her matris için kararlı.
- **Tekrarlı λ + Jordan** \rightarrow polinom \times üstel çözümler (rezonans).

35.2 $A^T A$ Pozitif Tanım

Enerji testi:

$$\mathbf{x}^T A^T A \mathbf{x} = (A\mathbf{x})^T (A\mathbf{x}) = \|A\mathbf{x}\|^2 \geq 0$$

Tam PD $\iff A$ kolonları bağımsız ($N(A) = \{\mathbf{0}\}$). Least squares matemağının temeli.

35.3 Pozitif Tanım Ek Gerçekler

- A^{-1} **PD** ($1/\lambda > 0$).
- $A + B$ **PD** (enerji toplanır).

35.4 Benzer Matris

$$B = M^{-1} A M$$

Bu özel kombinasyonu zaten biliyoruz — diagonalize $S^{-1} A S = \Lambda$.

Köşegenleştirme = benzerlik: $A \sim \Lambda$.

35.5 Aynı Özdeğer İspatı

$$A\mathbf{x} = \lambda\mathbf{x} \rightarrow M^{-1} A M (M^{-1}\mathbf{x}) = \lambda(M^{-1}\mathbf{x}).$$

$$B(M^{-1}\mathbf{x}) = \lambda(M^{-1}\mathbf{x})$$

λ aynı; özvektör $M^{-1}\mathbf{x}$.

35.6 Aile Örneği — Özdeğer 3, 1

$A = ((2, 1), (1, 2))$, $\Lambda = \text{diag}(3, 1)$. Rastgele $M \rightarrow B = M^{-1} A M$ — iz 4, det 3, **özdeğer 3, 1**.

Sonsuz matris bu ailede; **en sade üye** Λ .

A özdeğer: [3. 1.]

B özdeğer: [3. 1.]

trace(A) = 4.0 , trace(B) = 4.0

det(A) = 2.9999999999999996 , det(B) = 3.0000000000000004

35.7 Tekrarlı Özdeğer — İki Aile

$\lambda = 4, 4$ için iki ayrı aile:

Aile 1 — yalnız üye: $4I$. $M^{-1}(4I)M = 4I$ her M için.

Aile 2 — büyük aile: $\begin{pmatrix} 4 & 1 \\ 0 & 4 \end{pmatrix}$ ve özdeğer 4, 4 olan tüm defective matrisler. Tek özvektör \rightarrow diagonalize edilemez.

35.8 Jordan Formu

Defective ailenin en güzel üyesi (köşegen değil, **köşegene en yakın**):

$$J = \begin{pmatrix} 4 & 1 \\ 0 & 4 \end{pmatrix}$$

Jordan bloğu: köşegende tekrarlı λ , üstte 1, tek özvektör.

“For every missing eigenvector, we put a one above the diagonal.” — Strang, 39:31

35.9 Jordan Teoremi

Her kare matris bir Jordan matrisine benzerdir.

Blok sayısı = bağımsız özvektör sayısı.

- **İyi durum** (distinct λ): her blok $1 \times 1 \rightarrow J = \Lambda$.
- **Kötü durum** (tekrarlı + eksik): Jordan bloklarıyla “tamamlanır”.

“I’m not that crazy about the Jordan form. But I’m very positive about the singular value decomposition.” — Strang, 45:28

Builder Notu: Jordan sayısal kırılğan \rightarrow en ufak pertürbasyon özdeğer ayırır, rank değişir. Pratik araçlar SVD veya Schur ayrışımı.

35.10 Bu Dersin Özeti

1. $A^T A$ PSD; bağımsız kolon \rightarrow PD.
2. A^{-1} PD, $A + B$ PD.
3. $B = M^{-1}AM$ benzer.
4. Aynı özdeğer, özvektör $M^{-1}\mathbf{x}$.
5. Diagonalize = benzerlik.
6. Tekrarlı $\lambda \rightarrow$ iki aile (cI yalnız; defective büyük aile).
7. Jordan formu = köşegene en yakın.

8. **Jordan teoremi**; blok = özvektör sayısı.
9. **Sayısal kırılğan** → SVD tercih.

! Tek bir cümle

Benzer matrisler **aynı özdeğer ailesi**; en sade üye Λ (diagonalize edilebilir) ya da **Jordan formu** (defective). Jordan kırılğan → pratik **SVD** (Ders 29).

35.11 Kontrol Soruları

i Soru 1: $A^T A$ PSD ispatı.

$$\mathbf{x}^T A^T A \mathbf{x} = \|A\mathbf{x}\|^2 \geq 0. \text{ Sıfır ancak } A\mathbf{x} = \mathbf{0}.$$

i Soru 2: $A^T A$ tam PD ne zaman?

A kolonları bağımsız ($N(A) = \{\mathbf{0}\}$).

i Soru 3: Benzer matrislerin ortak ve farklı özellikleri?

Ortak: özdeğerler, trace, det. **Farklı:** özvektörler ($\mathbf{x} \rightarrow M^{-1}\mathbf{x}$), giriş değerleri.

i Soru 4: $4I$ niye tek aile?

$$M^{-1}(4I)M = 4I \text{ her } M \text{ için.}$$

i Soru 5: Aynı λ + aynı özvektör sayısı → benzer mi?

Hayır. 4×4 örnek: $3 + 1$ blok vs $2 + 2$ blok, ikisi de 2 özvektör ama benzer değil. Blok yapısı eşleşmeli.

35.12 Egzersizler

Egzersiz 1. $A = ((2, 1), (1, 2))$ PD. A^{-1} PD mi?

Egzersiz 2. $((5, 1), (-1, 3))$ Jordan formu? Diagonalize edilebilir mi?

Egzersiz 3. 3×3 $\lambda = 5, 5, 5, 1$ özvektör → Jordan formu?

Egzersiz 4. (Python) Benzer matris ispatı.

Egzersiz 5. İspatla: Benzer matrisler aynı karakteristik polinom. ($\det(B - \lambda I) = \det(M^{-1}) \det(A - \lambda I) \det(M)$.)

35.13 Sonraki Ders İçin Hazırlık

Ders 29: Tekil Değer Ayrışımı (SVD) — kursun gerçek doruğu. $A = U\Sigma V^T$ her matris için (kare/dikdörtgen, simetrik/değil). LoRA, PCA, boyut indirme, öneri.

⚠ Ders 29 öncesi

- Egzersiz 5 (karakteristik polinom).

35.14 Anahtar Kavramlar (Cheat Sheet)

Kavram	Tanım
A^{-1} PD	$\lambda \rightarrow 1/\lambda$
$A + B$ PD	Enerji toplanır
$A^T A$ PSD	$\ Ax\ ^2 \geq 0$; bağımsız kolon \rightarrow PD
Benzer	$B = M^{-1}AM$
Aynı λ	Özvektör $M^{-1}\mathbf{x}$
Diagonalize	$S^{-1}AS = \Lambda$; $A \sim \Lambda$
Jordan bloğu	λ köşegen, 1 üst, tek özvektör
Jordan teoremi	Her A bir J'ye benzer; #blok = #özvektör
İyi durum	Distinct $\rightarrow J = \Lambda$

35.15 ML Bağlantıları Özeti

💡 5 köprü

1. $A^T A = \mathbf{Gram}$ \rightarrow Kernel, kovaryans, normal denklemler, attention.
2. **Benzerlik = baz değişimi** \rightarrow Özdeğer baz-bağımsız invariant.
3. **Jordan kırılğan** \rightarrow SVD \rightarrow Pratik araçlar SVD/Schur.
4. **Tekrarlı λ** \rightarrow polinom \times üstel \rightarrow Rezonans, RNN/dinamik.
5. **Defective Hessian** \rightarrow optim güçlükleri.

❗ Tek bir şey alıp gideceksen

Benzer matrisler = aynı özdeğer ailesi; en sade Λ ya da **Jordan formu** (defective). Pratikte **SVD** (Ders 29) — kararlı, evrensel.

36 Tekil Değer Ayırışımı (SVD)

Kursun doruğu — $A = U\Sigma V^T$, LoRA/PCA'nın matematiği

i Bölüm bilgisi

- **Strang'ın videosu:** [YouTube — Lecture 29: Singular Value Decomposition](#) (≈40 dk)
- **OCW sayfası:** [MIT 18.06SC — Lecture 29](#)
- **Okuma süresi:** ≈40 dk

36.1 Bu Derste Ne Var?

Kursun doruğu: her matrisin en iyi ayrışımı — **SVD**. Jordan formu kırılındı; **SVD her matris için kararlı**.

1. $A = U\Sigma V^T$ — iki ortogonal \times köşegen.
2. **Geometri:** satır uzayında ortonormal \mathbf{v} 'ler \rightarrow kolon uzayında ortonormal \mathbf{u} 'lar; $A\mathbf{v}_i = \sigma_i\mathbf{u}_i$.
3. **Hesap:** $V = A^T A$ özvektörleri, $U = AA^T$ özvektörleri, $\sigma = \sqrt{\lambda}$.
4. **Dört temel alt-uzay** için doğru ortonormal bazlar.

“This is the final and best factorization of a matrix.” — Strang, 0:27

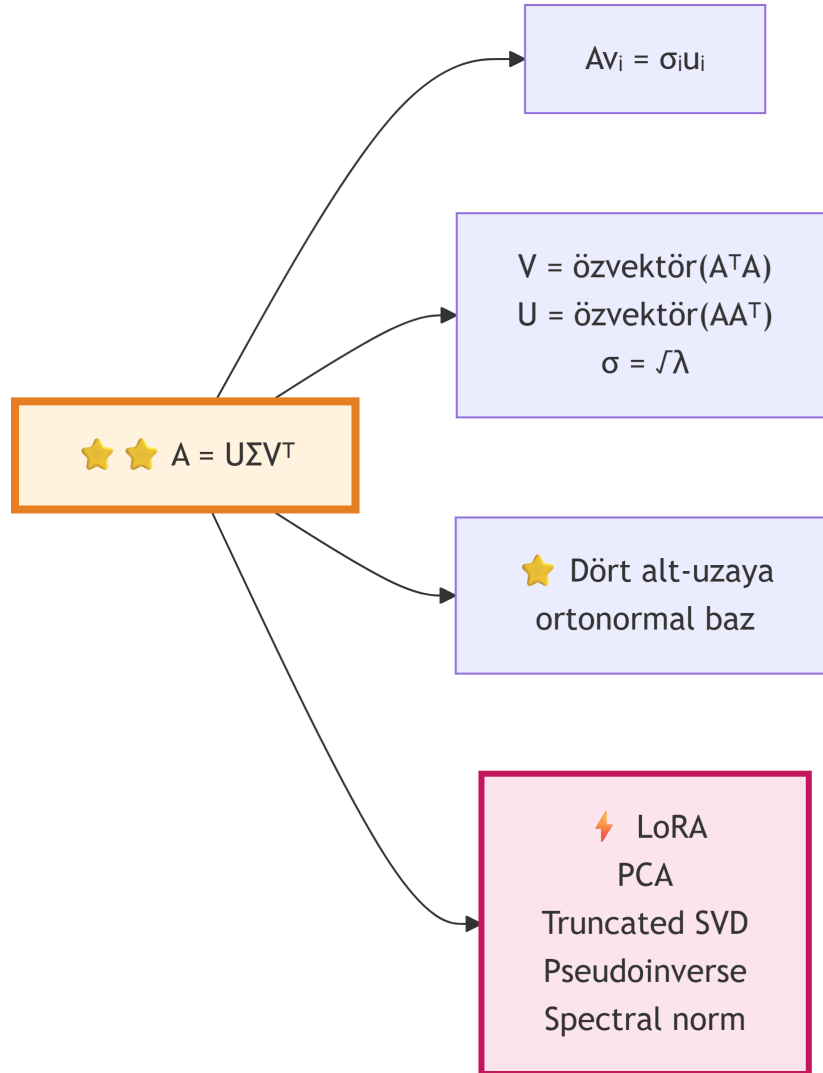
💡 Builder Notu — SVD = ML'in Tek En Önemli Aracı

- **LoRA** = $\Delta W = BA$, düşük-rank güncelleme; doğrudan SVD/Eckart-Young.
- **PCA** = veri matrisinin SVD'si; σ varyans, V ana bileşen.
- **Truncated SVD** = en iyi rank- k yaklaşım (Eckart-Young) \rightarrow görüntü/LSA sıkıştırma, öneri sistemleri.
- **Pseudoinverse** $A^+ = V\Sigma^+U^T$ (Ders 33).
- **Condition number** $\sigma_{\max}/\sigma_{\min}$ = sayısal kararlılık.
- **Spectral normalization** = σ_{\max} sınırı; GAN/Lipschitz.

36.2 SVD — $A = U\Sigma V^T$

$$A = U\Sigma V^T$$

- U ($m \times m$) ortogonal.



Şekil 36.1: SVD = kursun zirvesi: alt-uzay + ortogonalite + özdeğer birleşimi → LoRA, PCA, pseudoinverse.

- Σ köşegen, $\sigma_1 \geq \sigma_2 \geq \dots \geq 0$ (tekil değerler).
- V ($n \times n$) ortogonal.

Yeni: iki farklı ortogonal matris ($U \neq V$ genelde). Köşegenleştirmede S ortogonal değildi; SVD bunu kararlı yapar ve **her** matris için çalışır.

36.3 Geometrik Hedef

Satır uzayında ortonormal $\mathbf{v}_1, \dots, \mathbf{v}_r \rightarrow$ **kolon uzayında** ortonormal $\mathbf{u}_1, \dots, \mathbf{u}_r$:

$$A\mathbf{v}_i = \sigma_i \mathbf{u}_i$$

σ_i = gerdirme faktörü.

36.4 Matris Diline — $AV = U\Sigma$

$$AV = U\Sigma \implies A = U\Sigma V^T \quad (V \text{ ortogonal})$$

36.5 U 'ları Yok Etme — $A^T A = V\Sigma^2 V^T$

$$A^T A = V\Sigma^T \underbrace{U^T U}_I \Sigma V^T = V\Sigma^2 V^T$$

Bu özdeğer ayrışımı ($Q\Lambda Q^T$ biçimi):

- $V = A^T A$ özvektörleri.
- $\sigma^2 = A^T A$ özdeğerleri.

Aynı şekilde $AA^T = U\Sigma^2 U^T$.

36.6 Hesap Reçetesi

Bileşen	Kaynak
V	$A^T A$ özvektörleri
U	AA^T özvektörleri ($\mathbf{u}_i = A\mathbf{v}_i/\sigma_i$ tercih)
σ	$\sqrt{\lambda(A^T A)}$

İşaret tuzağı: U 'yu bağımsız hesaplıyorsan işaret tutmayabilir. Güvenli: $\mathbf{u}_i = A\mathbf{v}_i/\sigma_i$.

36.7 Örnek 1 — $A = \begin{pmatrix} 4 & 4 \\ -3 & 3 \end{pmatrix}$

$$A^T A = \begin{pmatrix} 25 & 7 \\ 7 & 25 \end{pmatrix}$$

Özvektörler $(1, 1)/\sqrt{2}, (1, -1)/\sqrt{2}$; özdeğerler 32, 18.

$$\sigma_1 = \sqrt{32}, \sigma_2 = \sqrt{18}, V = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

$AA^T = \text{diag}(32, 18) \rightarrow U = I$ (gerçekte $u_2 = -e_2$ işaret düzeltmesiyle).

U =

$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$$

$$\sigma = [5.65685425 \ 4.24264069]$$

V^T =

$$\begin{bmatrix} -0.7071 & -0.7071 \\ -0.7071 & 0.7071 \end{bmatrix}$$

$$\begin{bmatrix} -0.7071 & 0.7071 \\ -0.7071 & 0.7071 \end{bmatrix}$$

$$U \Sigma V^T - A = 1.4043333874306805e-15$$

36.8 Örnek 2 — Rank 1

$A = \begin{pmatrix} 4 & 3 \\ 8 & 6 \end{pmatrix}$ — her satır (4, 3) katı, rank 1.

$A^T A$ özdeğerleri: 125, 0 $\rightarrow \sigma_1 = \sqrt{125}, \sigma_2 = 0$. **Sıfır-olmayan σ sayısı = rank.**

36.9 SVD ve Dört Temel Alt-Uzay

Vektörler	Ortonormal baz	Boyut
$\mathbf{v}_1, \dots, \mathbf{v}_r$	Satır uzayı $C(A^T)$	r
$\mathbf{v}_{r+1}, \dots, \mathbf{v}_n$	Null uzayı $N(A)$	$n - r$
$\mathbf{u}_1, \dots, \mathbf{u}_r$	Kolon uzayı $C(A)$	r
$\mathbf{u}_{r+1}, \dots, \mathbf{u}_m$	Sol null $N(A^T)$	$m - r$

Ders 10'dan beri aranan **doğru ortonormal bazlar**. Hem ortonormal hem köşegenleştiren.

“It’s exactly the right basis for the four fundamental subspaces.” — Strang, 39:36

36.10 Simetrik PD — Özel Durum

A simetrik PD $\rightarrow A = Q\Lambda Q^T$ ile SVD çıkarılır:

$$U = V = Q, \quad \Sigma = \Lambda$$

36.11 Bu Dersin Özeti

1. $A = U\Sigma V^T$ her matris için.
2. $Av_i = \sigma_i u_i$.
3. $V = \text{özvektör}(A^T A)$, $U = \text{özvektör}(AA^T)$, $\sigma = \sqrt{\lambda}$.
4. **İşaret tuzağı:** $u_i = Av_i / \sigma_i$.
5. $A^T A$ ve AA^T aynı özdeğer (σ^2).
6. **Dört alt-uzay için ortonormal baz.**
7. **Simetrik PD:** $U = V = Q$.

! Tek bir cümle

$A = U\Sigma V^T$ her matrisi “dönüştür \rightarrow ölçekle \rightarrow dönüştür” şeklinde ayrıştırır; dört alt-uzaya doğru ortonormal bazlar. **LoRA, PCA, pseudoinverse, Eckart-Young, spectral norm — modern ML omurgası.**

36.12 Kontrol Soruları

i Soru 1: Niye iki ortogonal matris?

Genel A için özvektör matrisi ortogonal değil; satır uzayı (V) ve kolon uzayı (U) için ayrı bazlar.

i Soru 2: V nasıl bulunur?

$V = A^T A$ özvektörleri ($A^T A = V\Sigma^2 V^T$). Simetrik PSD \rightarrow ortogonal özvektör.

i Soru 3: σ ve $\lambda(A^T A)$ ilişkisi?

$\sigma_i = \sqrt{\lambda_i(A^T A)}$. AA^T aynı λ 'ya sahip.

i Soru 4: Rank 1 \rightarrow kaç σ ?

Bir sıfır-olmayan σ . Sıfır-olmayan σ sayısı = rank.

i Soru 5: Simetrik PD \rightarrow SVD?

$U = V = Q, \Sigma = \Lambda$. Tek ortogonal matris.

36.13 Egzersizler

Egzersiz 1. $A = \text{diag}(3, 2) \rightarrow$ SVD?

Egzersiz 2. $A = ((0, 2), (0, 0)) \rightarrow A^T A, \sigma, \text{rank}$.

Egzersiz 3. A 4×3 rank 2 $\rightarrow U, \Sigma, V$ boyutları, kaç $\sigma \neq 0$?

Egzersiz 4. (Python) `np.linalg.svd` + dört alt-uzay bazları.

Egzersiz 5. *İspatla:* En iyi rank- k yaklaşım = en büyük k tekil değeri tutmak (Eckart-Young, kavramsal). LoRA matematiği.

36.14 Sonraki Ders İçin Hazırlık

Ders 30: Lineer Dönüşümler ve Matrisleri

- Soyut lineer dönüşüm $T : V \rightarrow W$.
- Baz seçimine bağlı temsil.
- Türev, döndürme, projeksiyon — hep lineer dönüşüm.

! Ders 30 öncesi

- Egzersiz 5 (Eckart-Young).
- `np.linalg.svd` ile dene.

36.15 Anahtar Kavramlar (Cheat Sheet)

Kavram	Tanım
SVD	$A = U\Sigma V^T$
Temel ilişki	$A\mathbf{v}_i = \sigma_i \mathbf{u}_i$
V	$A^T A$ özvektörü
U	AA^T özvektörü; $\mathbf{u}_i = A\mathbf{v}_i / \sigma_i$
σ	$\sqrt{\lambda(A^T A)}$
Rank	# nonzero σ
Dört alt-uzay	\mathbf{v} satır+null, \mathbf{u} kolon+sol-null
Simetrik PD	$U = V = Q, \Sigma = \Lambda$
Condition number	$\sigma_{\max} / \sigma_{\min}$

36.16 ML Bağlantıları Özeti

💡 7 köprü

1. **LoRA** = $\Delta W = BA$ düşük-rank.
2. **PCA** = SVD veri matrisi.
3. **Eckart-Young** = en iyi rank- k yaklaşım.
4. **Pseudoinverse** $A^+ = V\Sigma^+U^T$.
5. **Condition number** = sayısal kararlılık.
6. **Spectral norm** = σ_{\max} ; GAN/Lipschitz.
7. **Truncated SVD** → görüntü sıkıştırma, LSA, öneri sistemleri.

! Tek bir şey alıp gideceksen

$A = U\Sigma V^T$ — kursun zirvesi. Dört alt-uzaya doğru ortonormal bazlar. LoRA, PCA, pseudoinverse, Eckart-Young, spectral norm — **modern ML omurgası**.

37 Lineer Dönüşümler ve Matrisleri

Matristen önce gelen soyut fikir; baz seçince matris doğar

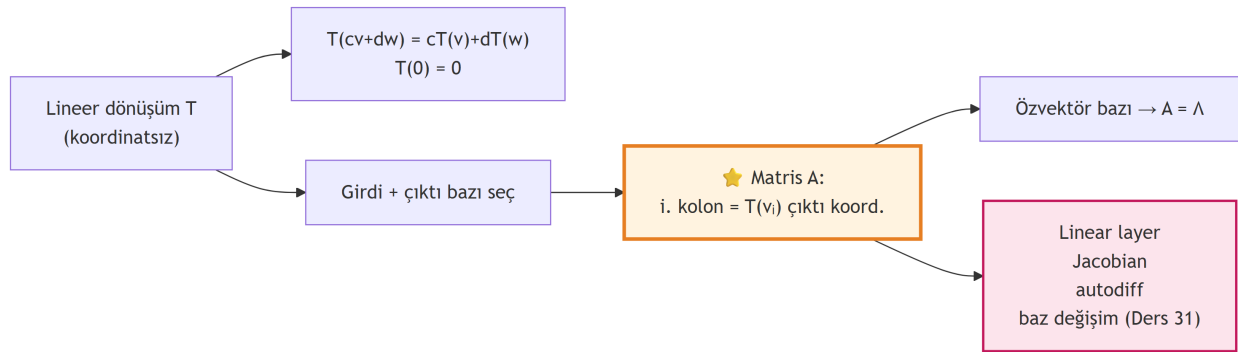
i Bölüm bilgisi

- **Strang'ın videosu:** [YouTube — Lecture 30: Linear Transformations and Their Matrices](#) (≈49 dk)
- **OCW sayfası:** [MIT 18.06SC — Lecture 30](#)
- **Okuma süresi:** ≈38 dk

37.1 Bu Derste Ne Var?

1. **Lineer dönüşüm:** $T(cv + dw) = cT(v) + dT(w)$. $T(\mathbf{0}) = \mathbf{0}$ **zorunlu**.
2. **Örnekler:** projeksiyon, rotasyon, türev (lineer); kaydırma, uzunluk (değil).
3. **Matris bazdan doğar:** girdi + çıktı bazı seç → matris belirlenir.
4. **Kurma kuralı:** A 'nın i . kolonu = $T(\mathbf{v}_i)$ 'nin çıktı bazındaki koordinatları.

“The way to understand linear transformations is to find the matrix that lies behind them.” — Strang, 17:15



Şekil 37.1: Lineer dönüşüm → baz seçimi → matris. İyi baz (özvektör) → köşegen Λ .

💡 Builder Notu — Lineer Dönüşüm ML’de

- **Lineer katman** $y = Wx$ saf lineer; **bias** ekleyince **affine** ($T(\mathbf{0}) = \mathbf{b} \neq \mathbf{0}$).
- **Baz seçimi = temsil** (representation); özvektör → köşegen → A^k, e^{At} ucuz.
- **Türev lineer** → otomatik türev (autodiff), backprop zincir kuralı.

- **Jacobian** = doğrusal-olmayan fonksiyonun yerel lineer dönüşüm matrisi.

37.2 Lineer Dönüşüm Tanımı

$$T(c\mathbf{v} + d\mathbf{w}) = cT(\mathbf{v}) + dT(\mathbf{w})$$

Zorunlu: $T(\mathbf{0}) = \mathbf{0}$ ($c = 3, \mathbf{v} = \mathbf{0} \rightarrow T(\mathbf{0}) = 3T(\mathbf{0})$).

37.3 Örnekler ve Karşı-Örnekler

Dönüşüm	Lineer?	Neden
Projeksiyon	✓	toplama/skala uyumlu
Rotasyon	✓	döndür-topla = topla-döndür
Türev	✓	$(f + g)' = f' + g'$
Kaydırma $\mathbf{v} \rightarrow \mathbf{v} + \mathbf{v}_0$	✗	$T(\mathbf{0}) = \mathbf{v}_0 \neq \mathbf{0}$
Uzunluk $\mathbf{v} \rightarrow \ \mathbf{v}\ $	✗	$\ -2\mathbf{v} \ = 2\ \mathbf{v}\ \neq -2\ \mathbf{v}\ $

37.4 $T(\mathbf{v}) = A\mathbf{v}$

Her matris bir lineer dönüşüm: $A(\mathbf{v} + \mathbf{w}) = A\mathbf{v} + A\mathbf{w}$ ✓.

Hedef: Bir dönüşümü anlamak = arkasındaki matrisi bulmak. Koordinat (baz) seçmek lazım.

37.5 Baz Yeter

T 'yi tam tanımak için tüm vektörlerine ne yaptığını bilmek gerekmez. **Bir baza** ne yaptığını bilmek yeter:

$$\mathbf{v} = \sum c_i \mathbf{v}_i \implies T(\mathbf{v}) = \sum c_i T(\mathbf{v}_i)$$

Sonsuz girdi, sonlu (n) bilgiyle kodlanır.

37.6 İki Baz — Girdi + Çıktı

$T : \mathbb{R}^n \rightarrow \mathbb{R}^m$ için:

- **Girdi bazı** $\mathbf{v}_1, \dots, \mathbf{v}_n$.
- **Çıktı bazı** $\mathbf{w}_1, \dots, \mathbf{w}_m$.

$$A \cdot (\text{girdi koord}) = (\text{çıktı koord})$$

37.7 Matris Kurma Kuralı

$$A\text{'nın } i. \text{ kolonu} = T(\mathbf{v}_i)\text{'nin çıktı bazındaki koordinatları}$$

Neden? Girdi koord $(1, 0, \dots, 0) \rightarrow \mathbf{v}_1 \rightarrow T(\mathbf{v}_1)$. A 'nın 1. kolonu çıkar.

37.8 Projeksiyon — İyi Baz vs Standart Baz

45° doğrusuna projeksiyon.

İyi baz (özvektör): \mathbf{v}_1 doğru üstü, \mathbf{v}_2 dik. $T(\mathbf{v}_1) = \mathbf{v}_1, T(\mathbf{v}_2) = \mathbf{0}$:

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} = \Lambda$$

Standart baz: $P = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix}$.

Aynı dönüşüm, farklı baz \rightarrow farklı matris.

“The eigenvector basis is the good basis — it leads to the diagonal matrix Λ .” — Strang, 37:27

37.9 Türev Lineer Dönüşümdür

Girdi: $\{1, x, x^2\}$; çıktı: $\{1, x\}$. $T(p) = p'$:

- $T(1) = 0 \rightarrow$ kolon $(0, 0)$.
- $T(x) = 1 \rightarrow$ kolon $(1, 0)$.
- $T(x^2) = 2x \rightarrow$ kolon $(0, 2)$.

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 2 \end{pmatrix}$$

Doğrula: $A(c_1, c_2, c_3)^T = (c_2, 2c_3)^T$ — tam türev \checkmark .

“We compute derivatives exactly because we know it’s a linear transformation.” — Strang, 47:19

Bonus: Matris çarpımı = dönüşüm bileşkesi; ters matris = ters dönüşüm.

37.10 Bu Dersin Özeti

1. **Lineer dönüşüm** koordinatsız; $T(\mathbf{0}) = \mathbf{0}$.
2. **Örnekler/karşı-örnekler** (projeksiyon \checkmark , kaydırma \times).
3. $T(\mathbf{v}) = A\mathbf{v}$.
4. **Baz yeter** (n bilgi).
5. **İki baz** seç.
6. **Kurma kuralı**: i . kolon = $T(\mathbf{v}_i)$ çıktı koord.
7. **İyi baz** \rightarrow köşegen.
8. **Türev** lineer; matrisi açık.

! Tek bir cümle

Lineer dönüşüm matristen önce gelir; baz seçince matris doğar (i . kolon = $T(\mathbf{v}_i)$ çıktı koord). **Özvektör bazı** en sade matrisi (Λ) verir; türev de bir lineer dönüşüm — bu autodiff'in temelidir.

37.11 Kontrol Soruları

i Soru 1: Lineer değil testi?

$T(\mathbf{0}) = \mathbf{0}$ mı? Kaydırma ihlal eder.

i Soru 2: Uzunluk niye lineer değil?

$\| -2\mathbf{v} \| = 2\|\mathbf{v}\| \neq -2\|\mathbf{v}\|$ (skala kuralı).

i Soru 3: Tam tanım için ne kadar bilgi?

Bir baza ne yaptığı ($T(\mathbf{v}_1), \dots, T(\mathbf{v}_n)$).

i Soru 4: Matris kurma kuralı?

i . kolon = $T(\mathbf{v}_i)$ 'nin çıktı bazındaki koordinatları.

i Soru 5: Aynı projeksiyon iki farklı matris?

Baz farklı. Özvektör bazı $\rightarrow \Lambda$; standart baz $\rightarrow P$.

37.12 Egzersizler

Egzersiz 1. $T(x, y) = (x, -y)$ standart matrisi? Lineer mi?

Egzersiz 2. $T(x, y) = (x + 1, y)$ lineer mi?

Egzersiz 3. İkinci türev matrisi $\{1, x, x^2\} \rightarrow \{1, x, x^2\}$ bazında.

Egzersiz 4. (Python) Matris çarpımı = bileşke.

Egzersiz 5. *İspatla:* İki lineer dönüşümün bileşkesi lineer; matrisi çarpımdır.

37.13 Sonraki Ders İçin Hazırlık

Ders 31: Baz Değişimi ve Görüntü Sıkıştırma

- Aynı dönüşüm farklı bazlarda \rightarrow **benzer matrisler** ($B = M^{-1}AM$).
- **JPEG / wavelet** sıkıştırma = iyi baza geç + küçük katsayıları at.

⚠ Ders 31 öncesi

- Egzersiz 5 (bileşke).
- Türev matrisini ($\{1, x, x^2\}$) yeniden hesapla.

37.14 Anahtar Kavramlar (Cheat Sheet)

Kavram	Kural
Lineerlik	$T(c\mathbf{v} + d\mathbf{w}) = cT(\mathbf{v}) + dT(\mathbf{w})$
Zorunlu	$T(\mathbf{0}) = \mathbf{0}$
$T(\mathbf{v}) = A\mathbf{v}$	Her matris lineer
Baz yeter	$T(\mathbf{v}) = \sum c_i T(\mathbf{v}_i)$
Matris kurma	$i.$ kolon = $T(\mathbf{v}_i)$ çıktı koord
İyi baz	Özvektör $\rightarrow \Lambda$
Projeksiyon (özvektör)	$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$
Projeksiyon (standart)	$P = A(A^T A)^{-1} A^T$
Türev $\{1, x, x^2\} \rightarrow \{1, x\}$	$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 2 \end{pmatrix}$
Matris çarpımı	= bileşke

37.15 ML Bağlantıları Özeti

💡 4 köprü

1. **Lineer katman + bias = affine** → Lineer kısım + aktivasyon mimarisi.
2. **Baz = temsil** → Embedding, PCA, autoencoder; özvektör bazında ucuz hesap.
3. **Türev lineer → autodiff** → Backprop zincir kuralı = Jacobian çarpımı.
4. **Jacobian = yerel lineer** → Newton, normalizing flows, duyarlılık analizi.

! Tek bir şey alıp gideceksen

Lineer dönüşüm matristen önce; baz seçince matris doğar (i . kolon = $T(\mathbf{v}_i)$ çıktı koord). Özvektör bazı = Λ . Türev de lineer = **autodiff temeli**.

38 Baz Değişimi ve Görüntü Sıkıştırma

JPEG/wavelet — sinyali iyi baza geçirip atmak

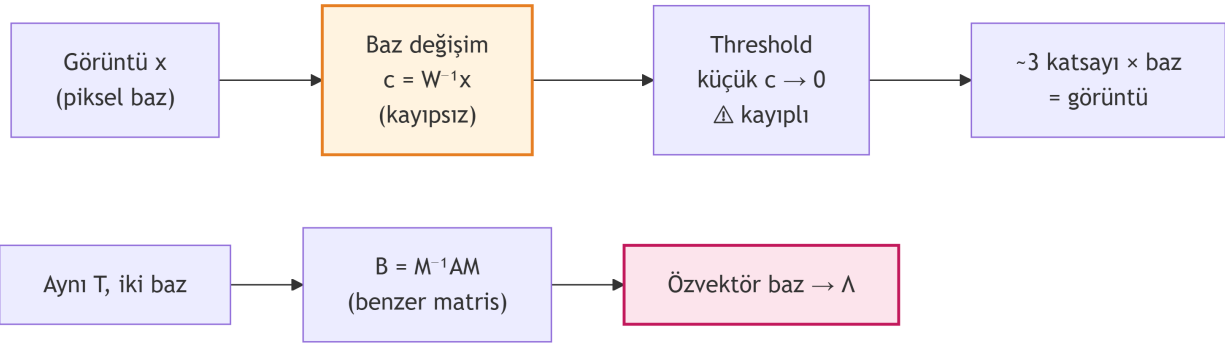
i Bölüm bilgisi

- **Strang'ın videosu:** [YouTube — Lecture 31: Change of Basis; Image Compression](#) (≈50 dk)
- **OCW sayfası:** [MIT 18.06SC — Lecture 31](#)
- **Okuma süresi:** ≈38 dk

38.1 Bu Derste Ne Var?

1. **Sıkıştırma = baz değişimi + thresholding** (JPEG: 64 → ~3 katsayı).
2. **Standart baz kötü, Fourier/wavelet iyi.**
3. $\mathbf{x} = W\mathbf{c}$, $\mathbf{c} = W^{-1}\mathbf{x}$.
4. **Aynı dönüşüm, iki baz** → $B = M^{-1}AM$ benzer; özvektör bazı → Λ .

“A good basis has a nice, fast inverse — and a few basis vectors come close to the signal.” — Strang, 28:31



Şekil 38.1: Sıkıştırma: $\mathbf{x} \rightarrow \mathbf{c}$ (kayıpsız) → threshold (kayıplı). Aynı dönüşüm farklı bazlarda → benzer matris.

💡 Builder Notu — Sıkıştırma → ML

- **Transform coding (JPEG/wavelet) → autoencoder/VAE:** öğrenilmiş encoder = W , latent = \mathbf{c} , bottleneck = threshold.

- **Embedding = öğrenilmiş baz** (Word2Vec, transformer latent).
- **PCA = en iyi (özvektör) baz**, ama pahalı; pratikte DCT/wavelet veya randomized SVD.
- **Fourier/wavelet öznelikleri** — scattering networks, FNO, spektrogram.

38.2 Görüntü = Vektör; Standart Baz Kötü

512×512 görüntü = $\mathbf{x} \in \mathbb{R}^{512^2}$. **Standart baz** komşu piksel korelasyonunu kullanmaz.

İyi baz vektörleri:

- **Hepsi-1** (1, 1, ..., 1) — düz görüntü.
- **Yarı-1/yarı(-1)** — düşük frekans.
- **Dama tahtası** (1, -1, 1, -1, ...) — yüksek frekans.

38.3 JPEG — Fourier Bazı, 8×8 Blok

JPEG 8×8 bloklara böler (64-boyut). Fourier bazına dönüştürür:

$$\underbrace{64 \text{ piksel } \mathbf{x}}_{\text{standart}} \xrightarrow{\text{kayıpsız}} \underbrace{64 \text{ katsayı } \mathbf{c}}_{\text{Fourier}}$$

Threshold (kayıplı): küçük $|c_i| < \epsilon \rightarrow 0$. Pürüzsüz blokta dama tahtası katsayıları küçük \rightarrow at.

$$64 \rightarrow \sim 3 \text{ katsayı} \quad (\sim 21 : 1 \text{ sıkıştırma})$$

38.4 Wavelet Bazı

Fourier rakibi — **ortogonal + lokal** (kenar/detay için iyi).

8-boyut için: (1, 1, 1, 1, 1, 1, 1, 1), (1, 1, 1, 1, -1, -1, -1, -1), (1, 1, -1, -1, 0, 0, 0, 0), ...

İç çarpımlar 0 \rightarrow ortogonal.

38.5 Baz Değişimi — $\mathbf{x} = W\mathbf{c}$

W kolonları = yeni baz vektörleri:

$$\mathbf{x} = W\mathbf{c} \implies \mathbf{c} = W^{-1}\mathbf{x}$$

İyi baz = hızlı W^{-1} + az vektörle iyi temsil. Ortonormal $\rightarrow W^{-1} = W^T$ (FFT, fast wavelet).

kolonlar dik? True

Wavelet katsayıları (çoğu sıfır): [75. 25. -0. -0. -0. -0. -0. -0.]

38.6 Benzer Matrisler — $B = M^{-1}AM$

Aynı T iki bazda: matrisler **benzer** (Ders 28). Aynı özdeğer. **Özvektör bazı** → matris Λ köşegen.

“To find the eigenvectors would be too expensive.” — Strang, 48:04

Görüntüde piksel matrisinin özvektörleri pahalı → sabit bazlar (Fourier/wavelet).

38.7 Bu Dersin Özeti

1. Sıkıştırma = baz değişimi + threshold.
2. Standart baz kötü; iyi: Fourier/wavelet.
3. JPEG: 8×8 Fourier, $64 \rightarrow \sim 3$.
4. Wavelet ortogonal + lokal.
5. $\mathbf{x} = W\mathbf{c}$, $\mathbf{c} = W^{-1}\mathbf{x}$.
6. Ortonormal → $W^{-1} = W^T$.
7. Benzer matrisler ($B = M^{-1}AM$).
8. Özvektör bazı → Λ (en iyi, pahalı).

! Tek bir cümle

Sıkıştırma = sinyali az büyük katsayıyla temsil eden baza geçirip gerisini atmaktır. Aynı dönüşüm farklı bazlarda **benzer matris** ($B = M^{-1}AM$); en sade = **özvektör bazında** Λ . Modern ML’de autoencoder/VAE = öğrenilmiş baz.

38.8 Kontrol Soruları

i Soru 1: Hangi adım kayıpsız/kayıplı?

Baz değişim **kayıpsız**; threshold **kayıplı**.

i Soru 2: Standart baz niye kötü?

Piksel korelasyonunu kullanmaz; küçük katsayı üretmez.

i Soru 3: İyi baz iki özellik?

(1) Hızlı W , W^{-1} . (2) Az vektörle iyi temsil.

i Soru 4: Ortonormal W^{-1} ?

W^T . İleri/geri aynı maliyet.

i Soru 5: İki bazdaki matris ilişkisi?

Benzer: $B = M^{-1}AM$; aynı özdeğer.

38.9 Egzersizler

Egzersiz 1. W kolonları $(1, 1), (1, -1)$. $\mathbf{x} = (3, 1)$ katsayıları?

Egzersiz 2. $(1, 1, 1, 1)$ ve $(1, 1, -1, -1)$ ortogonal mi?

Egzersiz 3. $A = ((2, 1), (0, 3)) \rightarrow$ baz değişiminde özdeğer aynı mı?

Egzersiz 4. (Python) Wavelet/DCT ile küçük sinyal sıkıştır.

Egzersiz 5. İspatla: Aynı T iki bazda $\rightarrow B = M^{-1}AM$. (M = baz değişim matrisi.)

38.10 Sonraki Ders İçin Hazırlık

Ders 32: Quiz 3 İncelemesi — Ders 21-31 (özdeğer, diagonalize, A^k , e^{At} , Markov, simetrik, PD, SVD, benzerlik) toplu tekrar.

! Ders 32 öncesi

- Egzersiz 5.
- DCT/wavelet sıkıştırmayı dene.

38.11 Anahtar Kavramlar (Cheat Sheet)

Kavram	Kural
Sıkıştırma	Baz değişim + threshold
JPEG	8×8 Fourier, $\sim 21:1$
Wavelet	Ortogonal + lokal
$\mathbf{x} = W\mathbf{c}$	$\mathbf{c} = W^{-1}\mathbf{x}$
İyi baz	Hızlı + iyi sıkıştırma
Ortonormal	$W^{-1} = W^T$
Benzer	$B = M^{-1}AM$
Özvektör baz	matris $\rightarrow \Lambda$

38.12 ML Bağlantıları Özeti

💡 4 köprü

1. **Transform coding** → autoencoder/VAE.
2. **Embedding** = öğrenilmiş baz.
3. **PCA** = özvektör baz (Ders 29 SVD ile).
4. **Fourier/wavelet öznitelikleri** → scattering, FNO, spektrogram.

! Tek bir şey alıp gideceksen

Sıkıştırma = iyi baza geçirip küçük katsayıları atmak. Aynı T farklı bazlarda benzer matrisler; **özvektör bazında köşegen** Λ . Modern ML: **autoencoder = öğrenilmiş baz**.

39 Quiz 3 İncelemesi

Özdeğer parmak izi: simetrik, ortogonal, PD, Markov, projeksiyon

Bölüm bilgisi

- **Strang'in videosu:** [YouTube — Quiz 3 Review](#) (≈47 dk)
- **OCW sayfası:** [MIT 18.06SC — Exam 3 Review](#)
- **Okuma süresi:** ≈35 dk

39.1 Bu Derste Ne Var?

Bölüm 6 (Ders 21–31) tekrarı.

1. **ODE + anti-simetrik** — saf sanal λ , periyodik.
2. **“Hangi c ?”** — diagonalize / simetrik / PD / Markov / projeksiyon.
3. **SVD + işaret tuzağı.**
4. **Simetrik + ortogonal** $\rightarrow \lambda = \pm 1$.

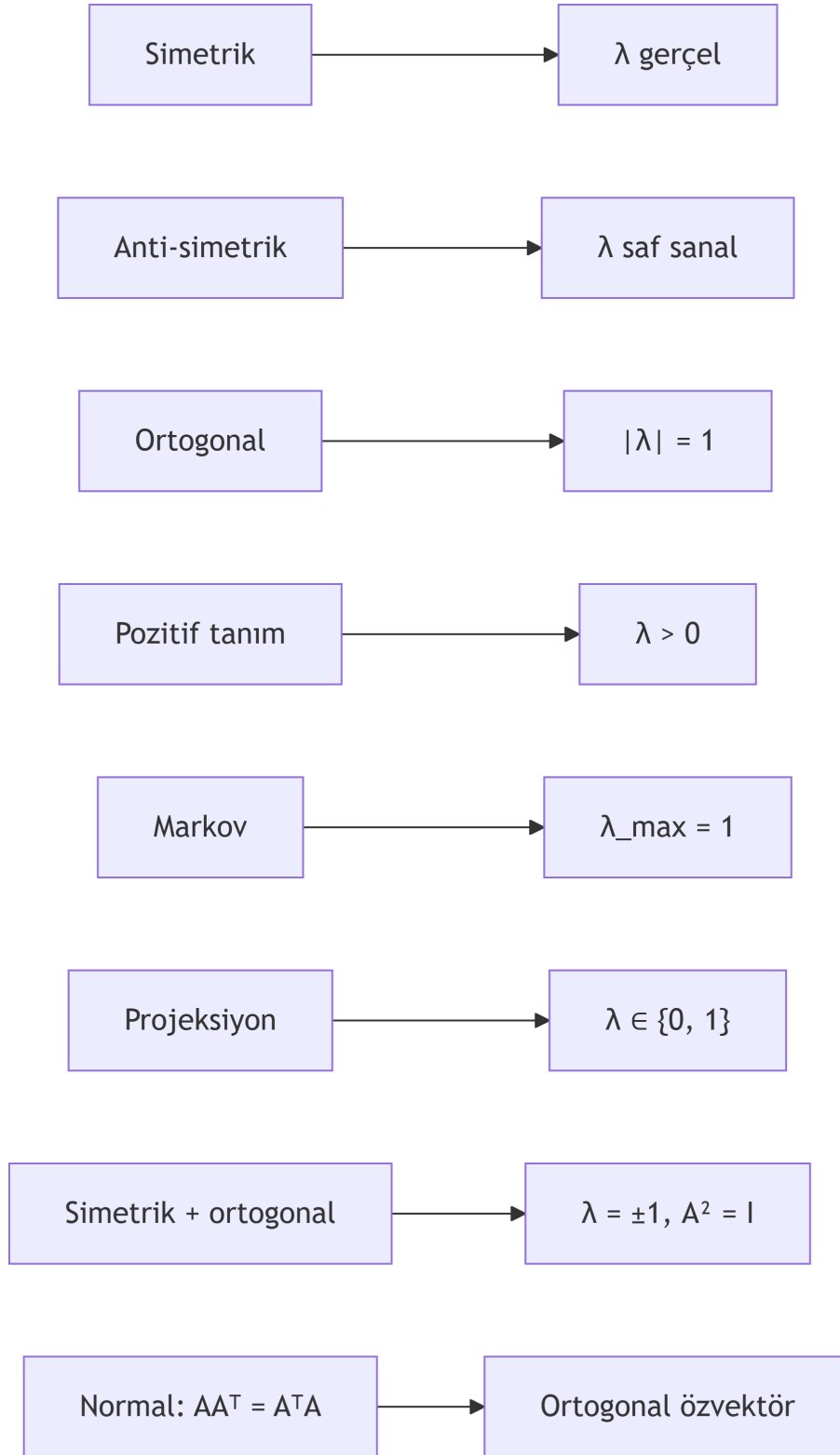
Builder Notu — Özdeğer Parmak İzi

ML'de matris sınıfını bilmek = davranışı bilmek.

- **Kovaryans/Gram** \rightarrow simetrik PSD \rightarrow PCA güvenli.
- **Ortogonal başlatma** $\rightarrow |\lambda| = 1 \rightarrow$ gradyan kararlı.
- **Markov** $\rightarrow \lambda_{\max} = 1 \rightarrow$ PageRank kararlı dağılım.
- **Anti-simetrik** \rightarrow salınım, enerji-koruyan (Hamiltonian NN, Lipschitz RNN).

39.2 Kapsam

Bölüm 6: özdeğer/özvektör, simetrik ($Q\Lambda Q^T$), PD, benzer matris, SVD.



Şekil 39.1: Özel matris sınıfları = özdeğer parmak izleri.

39.3 Problem 1: Anti-Simetrik ODE

$A^T = -A \rightarrow$ **anti-simetrik**. Karakteristik denklem $\lambda^3 + 2\lambda = 0$:

$$\lambda = 0, \pm\sqrt{2}i$$

Saf sanal $\rightarrow |e^{i\sqrt{2}t}| = 1$, çözüm **periyodik**. Periyot:

$$T = \frac{2\pi}{\sqrt{2}} = \pi\sqrt{2}$$

Ortogonal özvektör koşulu: $AA^T = A^T A$ (normal matris). Simetrik, anti-simetrik, ortogonal hepsi sağlar.

“A matrix has orthogonal eigenvectors exactly when $AA^T = A^T A$.” — Strang, 13:55

39.4 Problem 2: “Hangi c ?”

3×3 matris, ortogonal özvektörler, özdeğerler 0, 2, c :

- **Diagonalize:** her c (ortogonal özvektörler bağımsız).
- **Simetrik:** gerçek c (sanal olsaydı simetri yok).
- **PD:** hiçbir c ($\lambda = 0$ var).
- **Markov:** olamaz ($\lambda = 2 > 1$).
- $A/2$ **projeksiyon mu?** Özdeğerleri 0, 1, $c/2$. $c/2 \in \{0, 1\} \rightarrow c = 0$ veya $c = 2$.

39.5 SVD İşaret Tuzağı

V ve σ sabitle, sonra:

$$\mathbf{u}_i = \frac{A\mathbf{v}_i}{\sigma_i}$$

(Bağımsız AA^T özvektörleri \pm belirsizliği yaratır.)

39.6 SVD'den Matris Okuma

- $\Sigma = \text{diag}(3, 2) \rightarrow$ tersinir, rank 2.
- $\Sigma = \text{diag}(3, 0) \rightarrow$ rank 1, null boyut 1, null vektör = \mathbf{v}_2 .
- $\Sigma = \text{diag}(3, -5) \rightarrow$ geçersiz ($\sigma \geq 0$).

39.7 Simetrik + Ortogonal — $\lambda = \pm 1$

Simetrik \rightarrow gerçel; ortogonal $\rightarrow |\lambda| = 1$. $\lambda = \pm 1$.

İddia	D/Y	Neden
PD	Y	$\lambda = -1$ olabilir
Tekrarsız	Y	Büyük boyutta ± 1 tekrarlanır
Diagonalize	D	Simetrik/ortogonal her zaman
Tersinir	D	$\lambda = 0$ yok

$$A = A^T = A^{-1} \rightarrow A^2 = I.$$

39.8 $\frac{1}{2}(A + I)$ Projeksiyon mu?

A simetrik+ortogonal $\rightarrow A^2 = I$:

$$\left(\frac{A + I}{2}\right)^2 = \frac{A^2 + 2A + I}{4} = \frac{2A + 2I}{4} = \frac{A + I}{2} \checkmark$$

$P^2 = P + \text{simetrik} \rightarrow$ **projeksiyon**. Özdeğerler: A 'da $\pm 1 \rightarrow A + I$ 'de $0, 2 \rightarrow \frac{A+I}{2}$ 'de $0, 1 \checkmark$.

39.9 Özet — Parmak İzi Tablosu

Sınıf	Özdeğer parmak izi
Simetrik	λ gerçel, ortogonal özvektör
Anti-simetrik	λ saf sanal
Ortogonal	$ \lambda = 1$
PD	$\lambda > 0$
Markov	$\lambda_{\max} = 1$
Projeksiyon	$\lambda \in \{0, 1\}$
Sim + ort	$\lambda = \pm 1, A^2 = I$
Normal ($AA^T = A^T A$)	Ortogonal özvektör
e^{At}	$S e^{\Lambda t} S^{-1}$
SVD	$A = U \Sigma V^T, V = \text{eig}(A^T A), \mathbf{u}_i = A \mathbf{v}_i / \sigma_i$

! Tek bir cümle

Özel matris sınıfları **özdeğer parmak izleriyle** tanınır; bu kestirme matrisi hesaplamadan davranışı söyler. Simetrik + ortogonal $\rightarrow \lambda = \pm 1, A^2 = I$.

39.10 Kontrol Soruları

i Soru 1: Anti-simetrik özdeğer?

Saf sanal. Periyodik çözüm.

i Soru 2: Ortogonal özvektör koşulu?

$AA^T = A^T A$ (normal).

i Soru 3: Sim + ort özdeğer?

± 1 .

i Soru 4: SVD U işaret hatasız?

$\mathbf{u}_i = A\mathbf{v}_i/\sigma_i$.

i Soru 5: Sim + ort $\rightarrow A^2$?

$A = A^T = A^{-1} \rightarrow A^2 = I$.

39.11 Egzersizler

Egzersiz 1. $A = ((0, -1), (1, 0))$ — özdeğerler, geometrik anlam.

Egzersiz 2. $\lambda = 1, 1, 0$ simetrik — Markov? Projeksiyon?

Egzersiz 3. $\Sigma = \text{diag}(5, 0, 0)$ 3×3 — rank, null boyutu?

Egzersiz 4. (Python) Parmak izi tablosu doğrulama.

Egzersiz 5. *İspatla:* Normal matris ortogonal özvektör.

39.12 Sonraki Ders İçin Hazırlık

Ders 33: Pseudoinverse — sol/sağ ters, $A^+ = V\Sigma^+U^T$.

⚠ Ders 33 öncesi

- Parmak izi tablosu ezberle.
- SVD egzersizlerini gözden geçir.

39.13 Anahtar Kavramlar (Cheat Sheet)

Konu	Anahtar
Genel ODE çözüm	$\sum c_i e^{\lambda_i t} \mathbf{x}_i$
Anti-simetrik	λ saf sanal, periyodik
Periyot	$T = 2\pi/\omega$
Ortogonal özvektör	$AA^T = A^T A$
e^{At}	$S e^{\Lambda t} S^{-1}$
PD	$\lambda > 0$
Markov	$\lambda_{\max} = 1$
Projeksiyon	$\lambda \in \{0, 1\}$
Sim + ort	$\lambda = \pm 1, A^2 = I$
SVD rank	# nonzero σ ; $\sigma < 0$ imkansız

39.14 ML Bağlantıları Özeti

💡 3 köprü

1. **Parmak izi** → Mimari kararlar (PCA, ortogonal başlatma, Markov RNN).
2. **Anti-simetrik** → Hamiltonian NN, Lipschitz RNN (enerji-koruyan).
3. **Ortogonal/normal** → Spektral normalizasyon, unitary RNN, gradyan kararlılığı.

❗ Tek bir şey alıp gideceksen

Özdeğer parmak izi her özel sınıfı tanıır: simetrik gerçel, ortogonal $|\lambda| = 1$, PD $\lambda > 0$, Markov $\lambda_{\max} = 1$, projeksiyon $\{0, 1\}$. Sim+ort → $\pm 1, A^2 = I$.

40 Sol ve Sağ Tersler, Pseudoinverse

$A^+ = V\Sigma^+U^T$ — satır uzayı \leftrightarrow kolon uzayı

i Bölüm bilgisi

- **Strang'in videosu:** [YouTube — Lecture 33: Left and Right Inverses; Pseudoinverse](#) (≈ 42 dk)
- **OCW sayfası:** [MIT 18.06SC — Lecture 33](#)
- **Okuma süresi:** ≈ 36 dk

40.1 Bu Derste Ne Var?

Ters: rank'e göre 4 durum.

1. **İki taraflı:** $r = m = n$.
2. **Sol ters:** tam kolon rank $\rightarrow (A^T A)^{-1} A^T$.
3. **Sağ ters:** tam satır rank $\rightarrow A^T (A A^T)^{-1}$.
4. **Pseudoinverse** A^+ : genel; $A^+ = V\Sigma^+U^T$.

"From the row space to the column space, A is perfect — invertible. Its inverse is the pseudoinverse." — Strang, 26:09

💡 Builder Notu — Pseudoinverse = Regresyon Motoru

- `np.linalg.pinv` / `torch.linalg.pinv` \rightarrow least squares + minimum-norm.
- **Rank eksik** ($A^T A$ singüler) \rightarrow **ridge** ($A^T A + \lambda I$) ya da truncated pseudoinverse.
- **Modern aşırı-parametre rejimi** \rightarrow parametre $>$ veri = sağ ters + minimum-norm; gradient descent **implicit regularization**.

40.2 Dört Durum

Durum	Koşul	Ters
İki taraflı	$r = m = n$	A^{-1}
Sol	$r = n < m$	$(A^T A)^{-1} A^T$
Sağ	$r = m < n$	$A^T (A A^T)^{-1}$

Durum	Koşul	Ters
Pseudoinverse	$r < m, n$	$A^+ = V\Sigma^+U^T$

40.3 Sol Ters — Tam Kolon Rank

Kolonlar bağımsız $\rightarrow A^T A$ tersinir.

$$A_{\text{sol}}^{-1} = (A^T A)^{-1} A^T$$

Soldan çarpınca I . **Least squares'in kalbi** (Ders 16).

40.4 Sağ Ters — Tam Satır Rank

Satırlar bağımsız $\rightarrow AA^T$ tersinir.

$$A_{\text{sağ}}^{-1} = A^T (AA^T)^{-1}$$

Sağdan çarpınca I . $Ax = b$ her zaman çözülür, $n - m$ serbest.

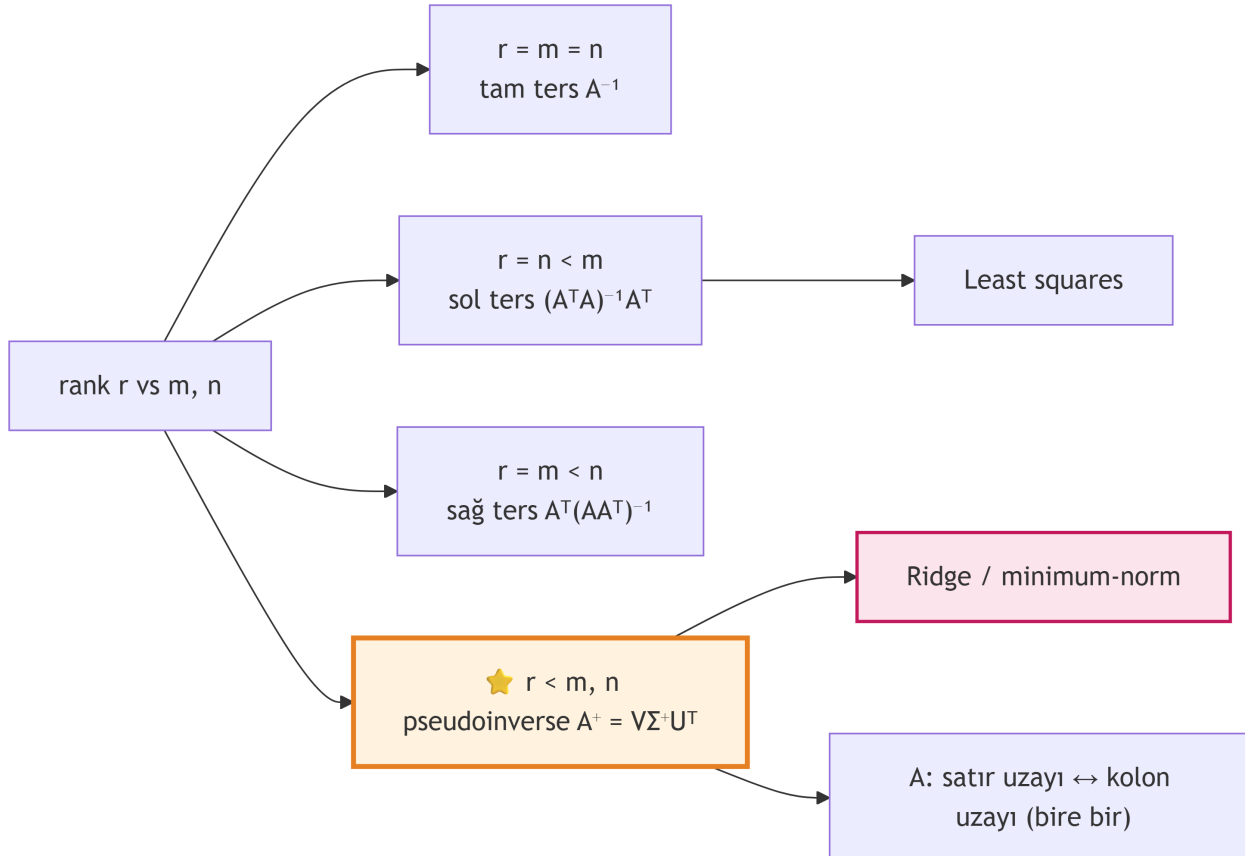
40.5 Yanlış Tarafa \rightarrow Projeksiyon

Sol tersi **sağdan**: $A(A^T A)^{-1} A^T = P$ (kolon uzayına projeksiyon). I değil — yapabildiği yerde I , yapamadığında 0 .

40.6 Pseudoinverse — Satır \leftrightarrow Kolon Mucizesi

A , **satır uzayını kolon uzayına** bire bir eşler. **İspat**: x, y satır uzayında, $Ax = Ay$ varsay $\rightarrow A(x - y) = \mathbf{0} \rightarrow x - y \in N(A)$. Ama satır uzayı ve null uzayı **dik tümleyen** (Ders 14) $\rightarrow x - y \in N(A) \cap C(A^T) = \{\mathbf{0}\} \rightarrow x = y$.

A^+ : bu eşlemenin tersi — kolon uzayından satır uzayına, null uzayını siler.



Şekil 40.1: Dört durum + pseudoinverse mucizesi: A , satır uzayını kolon uzayına bire bir eşler.

40.7 SVD'den A^+

$$A = U\Sigma V^T \rightarrow A^+ = V\Sigma^+U^T.$$

Σ^+ : sıfır-olmayan $\sigma_i \rightarrow 1/\sigma_i$, sıfırlar kalır, şekil $n \times m$.

$$A^+ = \begin{bmatrix} 0.04 & 0.08 \\ 0.08 & 0.16 \end{bmatrix}$$

$A A^+$ (kolon uzayına projeksiyon):

$$\begin{bmatrix} 0.2 & 0.4 \\ 0.4 & 0.8 \end{bmatrix}$$

$A^+ A$ (satır uzayına projeksiyon):

$$\begin{bmatrix} 0.2 & 0.4 \\ 0.4 & 0.8 \end{bmatrix}$$

min-norm çözüm $x = [0.6 \ 1.2]$

$A \cdot x = [3. \ 6.]$ (b'ye eşit)

40.8 Σ^+ ve Projeksiyonlar

- $\Sigma\Sigma^+$ ($m \times m$): köşegende r tane 1, gerisi 0 \rightarrow **kolon uzayı projeksiyonu**.
- $\Sigma^+\Sigma$ ($n \times n$): köşegende r tane 1 \rightarrow **satır uzayı projeksiyonu**.

$A^+A =$ satır uzayı projeksiyonu, $AA^+ =$ kolon uzayı projeksiyonu. I **değil, projeksiyon**.

40.9 İstatistik / Least Squares Bağlantısı

Klasik LS tam kolon rank gerektirir. Bağımlı özellikler $\rightarrow A^T A$ singüler \rightarrow pseudoinverse devreye girer (en iyi, **minimum-norm** çözüm).

“Statisticians discovered: oh boy, this is the thing we needed all our lives.” — Strang, 30:56

40.10 Bu Dersin Özeti

1. **Dört durum.**
2. **Yanlış taraf \rightarrow projeksiyon.**
3. **A satır \leftrightarrow kolon mucizesi** (ispat: dik tümleyen).
4. $A^+ = V\Sigma^+U^T$.
5. $A^+A, AA^+ =$ **projeksiyon**.
6. Rank-eksik LS / regresyon.

! Tek bir cümle

Bir matrisin tersi rank'e bağlı: tam $\rightarrow A^{-1}$, sol/sağ ters dikdörtgende, $A^+ = V\Sigma^+U^T$ genelde. A satır uzayını kolon uzayına bire bir eşler; A^+ bunun tersi. **Pseudoinverse = least squares + minimum-norm motoru.**

40.11 Kontrol Soruları

i Soru 1: Sol ters ne zaman, formül?

Tam kolon rank ($r = n < m$). $(A^T A)^{-1} A^T$.

i Soru 2: Sağ ters? Kaç çözüm?

Tam satır rank ($r = m < n$). Her \mathbf{b} için **sonsuz** çözüm.

i Soru 3: Dikdörtgen niye iki taraflı tersi yok?

Bir null uzay var \rightarrow vektör sıfıra gider \rightarrow ters geri getiremez.

i Soru 4: A^+ neden satır \leftrightarrow kolon tersi?

Dik tümleyen ispatı: $\mathbf{x} - \mathbf{y} \in N(A) \cap C(A^T) = \{\mathbf{0}\}$. $A^+A =$ satır projeksiyon, $AA^+ =$ kolon projeksiyon.

i Soru 5: SVD'den A^+ ?

$A^+ = V\Sigma^+U^T$, $\Sigma^+ = \sigma \rightarrow 1/\sigma$ (nonzero), şekil $n \times m$.

40.12 Egzersizler

Egzersiz 1. $A = (1, 1)^T$ (2×1) \rightarrow sol ters.

Egzersiz 2. $A = (1, 1)$ (1×2) \rightarrow sağ ters.

Egzersiz 3. $\Sigma = \text{diag}(4, 0) \rightarrow \Sigma^+, \Sigma^+\Sigma$.

Egzersiz 4. (Python) `pinv` ile rank-eksik LS.

Egzersiz 5. İspatla: A^+A satır uzayına projeksiyon (idempotent, simetrik).

40.13 Sonraki Ders İçin Hazırlık

Ders 34: Final İncelemesi — tüm kursun bir araya gelişi.

⚠ Ders 34 öncesi

- Egzersiz 5.
- `pinv` ile birkaç matrise dene.

40.14 Anahtar Kavramlar (Cheat Sheet)

Durum	Koşul	Ters
Tam	$r = m = n$	A^{-1}
Sol	$r = n < m$	$(A^T A)^{-1} A^T$
Sağ	$r = m < n$	$A^T (A A^T)^{-1}$
Pseudo	$r < m, n$	$V \Sigma^+ U^T$
$A^+ A$	—	Satır uzayı projeksiyon
$A A^+$	—	Kolon uzayı projeksiyon
Σ^+	—	$\sigma \rightarrow 1/\sigma$
Mucize	—	A satır↔kolon bire bir

40.15 ML Bağlantıları Özeti

💡 4 köprü

1. **pinv = regresyon motoru** → LS + min-norm.
2. **Rank-eksik** → **ridge** ($A^T A + \lambda I$).
3. **Truncated pinv** → ill-conditioned problem (deblurring, ters problem).
4. **Modern overparametrize** → sağ ters + min-norm; gradient descent **implicit regularization**.

❗ Tek bir şey alıp gideceksen

Ters rank'e bağlı: dört durum. $A^+ = V \Sigma^+ U^T$ = satır↔kolon bire-bir tersi. **Min-norm least squares motoru** — `np.linalg.pinv`.

41 Final İncelemesi

Tüm kursun bir araya gelişi — 34/34 🎓

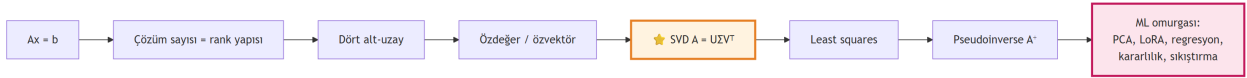
i Bölüm bilgisi

- **Strang'ın videosu:** [YouTube — Final Course Review](#) (≈43 dk)
- **OCW sayfası:** [MIT 18.06SC — Final Review](#)
- **Okuma süresi:** ≈35 dk

41.1 Bu Derste Ne Var?

Kursun son dersi — Bölüm 1–7'nin bir araya gelişi.

1. **Rank ve çözülebilirlik.**
2. $A^T A / A A^T$ gerçekleri.
3. **Null uzayı ve teklik.**
4. **Markov, least squares, 2x2 hızlı sorular.**



Şekil 41.1: Tüm kurs tek bir bağlı resim: rank → alt-uzay → özdeğer → SVD → LS → pseudoinverse.

💡 Builder Notu — ML'in Dili

Lineer cebir tek bir bağlı resim. ML'de:

- **Forward pass** = matris çarpımı.
- **Eğitim** = optimizasyon (PD Hessian).
- **Embedding** = baz değişimi.
- **Attention** = iç çarpım.
- **LoRA** = SVD / düşük-rank.

Bu 34 dersin her kavramı modern ML'de doğrudan karşılık bulur.

41.2 Soru 1: Rank ve Çözülebilirlik

$3 \times n$ matris. $Ax = (1, 0, 0)$ çözümsüz, $Ax = (0, 1, 0)$ tek çözüm.

- $m = 3$.
- Çözümsüz var \rightarrow bazı satırlar bağımlı $\rightarrow r < m$.
- Tek çözüm \rightarrow null = $\{0\}$ $\rightarrow r = n$.

Sonuç: $m = 3, r = n < 3$.

“No solution tells me $r < m$; exactly one solution tells me $r = n$.” — Strang, 3:50

41.3 Soru 1 T/F

İddia	D/Y	Neden
$\det(A^T A) = \det(AA^T)$	Y	A kare değil; $A^T A$ tersinir, AA^T singüler
$A^T A$ tersinir	D	$r = n$
AA^T PD	Y	rank < 3 , sadece PSD

Önemli: $\det(AB) = \det(BA)$ yalnız kare matriste.

41.4 $A^T y = c$ — Varlık + Çokluk

- A 'nın kolonları bağımsız $\rightarrow A^T$ satırları bağımsız \rightarrow her c için çözüm.
- A^T null boyutu $m - r > 0 \rightarrow$ sonsuz çözüm.

41.5 Soru 2: Kolon Kombinasyonu

$Ax = v_1 - v_2 + v_3 \rightarrow x = (1, -1, 1)$.

$v_1 - v_2 + v_3 = 0$ ise $(1, -1, 1) \in N(A) \rightarrow$ çözüm asla tek değil.

41.6 Soru 3: Markov Kararlı Durum

Kolonlar bağımlı (singüler), trace 0.8 $\rightarrow \lambda = 0, 1, -0.2$.

$u_0 = (0, 10, 0)$. $k \rightarrow \infty$ sadece $\lambda = 1$ kalır.

$\lambda = 1$ özvektör $(3, 3, 4)$. Toplam korunur: $3 + 3 + 4 = 10 \rightarrow c_2 = 1$:

$$u_\infty = (3, 3, 4)$$

41.7 Soru 4: 2x2 Hızlı

İstek	Cevap
$\mathbf{a} = (4, -3)$ projeksiyon	$P = \mathbf{a}\mathbf{a}^T / \mathbf{a}^T \mathbf{a}$
Özdeğer 0, 3; özvektör $(1, 2), (2, 1)$	$A = S\Lambda S^{-1}$
$B^T B$ olarak yazılamaz	Simetrik-olmayan herhangi A
Ortogonal özvektör + simetrik değil	Anti-simetrik veya ortogonal Q

41.8 Soru 5: Least Squares

$$(0, 3), (1, 4), (2, 1) \rightarrow y = C + Dt, \hat{C} = 11/3, \hat{D} = -1.$$

- **Projeksiyon p:** $\frac{11}{3}\mathbf{c}_1 - \mathbf{c}_2$.
- **Çözüm 0** olsun: $\mathbf{b} \perp C(A)$. Örnek: $\mathbf{b} = (1, -2, 1)$ kolonlara dik \rightarrow en iyi $(0, 0)$.

41.9 Final Stratejisi

- **Çözüm sayısından rank oku.**
- $A^T A$ tersinir $\iff r = n$.
- **Özel matrisi özdeğerinden tanı.**
- **Markov:** $\lambda = 1 +$ korunum.
- **LS = projeksiyon;** $\mathbf{b} \perp C \rightarrow$ çözüm 0.
- $\det(AB) = \det(BA)$ yalnız kare.

41.10 Bu Dersin Özeti

1. **Çözülebilirlik** \leftrightarrow rank.
2. $A^T A, AA^T$ rank yapısına bağlı.
3. **Null uzayı** \leftrightarrow teklik.
4. **Ortonormal vektörler** \rightarrow dik projeksiyon.
5. **Markov:** $\lambda = 1 +$ toplam korunumu.
6. **LS = kolon uzayı projeksiyonu.**
7. **Özel matris = parmak izi.**

! Tek bir cümle

Tüm kurs **tek bağlı resim:** rank \rightarrow 4 alt-uzay \rightarrow özdeğer \rightarrow SVD \rightarrow LS \rightarrow pseudoinverse. Bir problemi (regresyon, boyut indirme, kararlılık) doğru lineer cebir aracına eşlemek bütünsel bakıştan gelir — ML model tasarımının sezgisel temeli.

41.11 Kontrol Soruları

i Soru 1: Çözumsuz + tek çözüm \rightarrow rank?

$$r < m + r = n \rightarrow r = n < m.$$

i Soru 2: $\det(A^T A) = \det(AA^T)$ ne zaman?

Yalnız A kare. Dikdörtgende biri 0, diğeri $\neq 0$ olabilir.

i Soru 3: Kolon komb. = 0 \rightarrow teklik?

$(1, -1, 1) \in N(A) \rightarrow$ çözüm tek değil.

i Soru 4: Markov kararlı durum?

$\lambda = 1$ özvektörü + toplam korunum.

i Soru 5: LS çözüm 0 için b ?

$b \perp C(A).$

41.12 Egzersizler

Egzersiz 1. $\lambda = 2, 5$, özvektör $(1, 0), (1, 1) \rightarrow A = SAS^{-1}$.

Egzersiz 2. $A 4 \times 6$, rank 4 \rightarrow çözüm var mı, tek mi?

Egzersiz 3. $b = (1, 1, 1) \rightarrow$ sabit $y = C$ fit \rightarrow ortalama.

Egzersiz 4. (Python) Tüm kavramları tek örnekte bir araya getir.

Egzersiz 5. İspat: Tek satırla, rank-1 + $C(A)$ + LS.

41.13 Kurs Tamamlandı 🎓

MIT 18.06 Lineer Cebir'in 34 dersi tamamlandı.

Bu Türkçe uyarlamada, Strang'ın pedagojik sırası korunarak her dersin yanına **ML/Builder köprüleri** eklendi: PCA, SVD, LoRA, normalizing flows, attention, RNN kararlılığı, Markov/PageRank, Cholesky, Fourier features...

“Above all, thanks for taking the course.” — Strang

Sonrası: öğrendiklerini koda dök (NumPy/PyTorch), gerçek veriyle PCA/SVD/regresyon dene, Strang'ın videolarını referans olarak sakla.

“Linear algebra is about the nice part of calculus, where everything’s flat and the formulas come out right.” — Strang, 21:52

41.14 Cheat Sheet (Kurs Geneli)

Soru	Anahtar
Çözüm var mı?	$\mathbf{b} \in C(A)$?
Tek mi?	$N(A) = \{\mathbf{0}\}$? ($r = n$)
Her \mathbf{b} çözülür?	Tam satır rank ($r = m$)
$A^T A$ tersinir	$r = n$
$\det(A^T A) = \det(AA^T)$	Yalnız A kare
Markov kararlı	$\lambda = 1$ özvektör + toplam
LS = projeksiyon	\mathbf{b} 'nin $C(A)$ üzerine
Çözüm = $\mathbf{0}$	$\mathbf{b} \perp C(A)$
$A = S\Lambda S^{-1}$	Özvektör/özdeğerden
Özel matris	Özdeğer parmak izi
SVD	$A = U\Sigma V^T$ her matrise
Pseudoinverse	$A^+ = V\Sigma^+U^T$

41.15 ML Bağlantıları (Kapanış)

💡 4 büyük tema

1. **Rank-çözülebilirlik = model belirlenebilirliği** → over/under-determined; regularization.
2. $A^T A$, **özdeğer**, **SVD = tek araç kutusu** → PCA, regresyon, kararlılık, sıkıştırma aynı kavramlar.
3. **Markov kararlı durum** → PageRank, RNN, MCMC.
4. **Lineer cebir = ML'in dili** → forward pass, embedding, attention, LoRA hepsi buradan.

! 🎓 Tek bir şey alıp gideceksen

Lineer cebir **tek bir bağlı resim** — rank, dört alt-uzay, özdeğer, SVD, least squares. Modern ML'in matematik dili. **NumPy ile pekiştir, gerçek veride dene, Strang'i referans olarak sakla.**

34/34 — Lineer cebir yolculuğu burada bitiyor; gerçek öğrenme, koddan başlıyor.

