~~Boring~~

Review of Essential Mathematics for Machine Learning (Pt. 1)

Linear Algebra

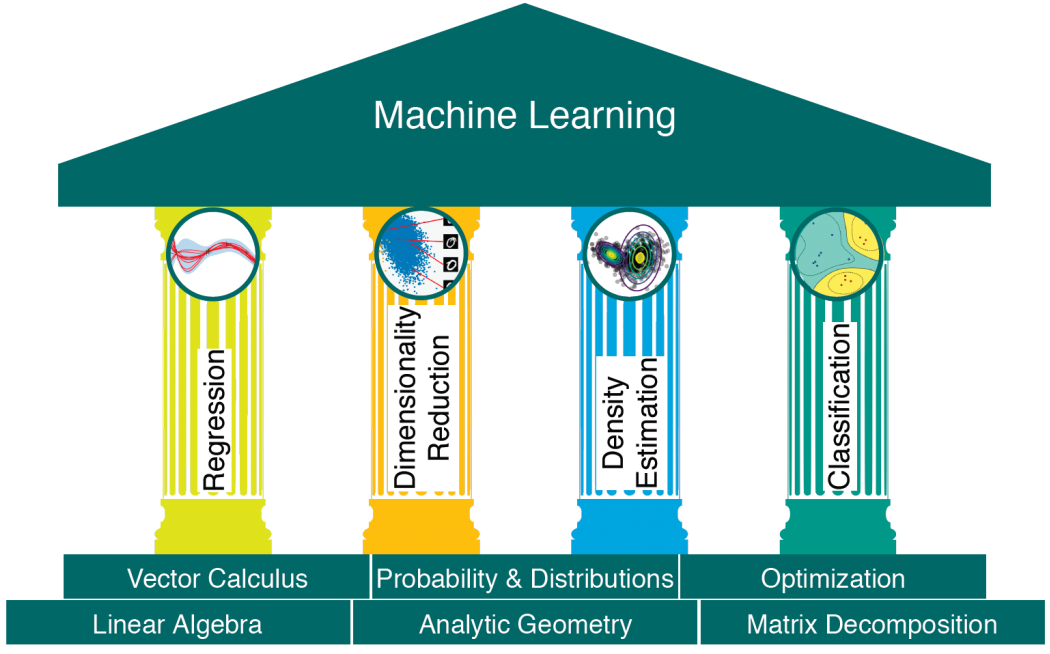
Image sources: **MML:** *Mathematics for Machine Learning* by Deisenroth, Faisal, and Ong @ mml-book.com

Mathematics in Machine Learning

Machine learning involves interrelated tasks of

- *representing data,*
- *modeling data, and*
- *learning from data*

(with little to no human intervention)



This requires understanding mathematical

- *expression (e.g., vectors, tensors, geometry)*
- *manipulation (e.g., functional mapping, probability), and*
- *optimization (e.g., vector calculus, matrix decomposition)*

PVT data for methane

Thermophysical properties of coexisting gaseous and liquid methane

T K	Pres. MPa	Density kg/m³	Density mol/dm³	E J/mol	H J/mol	S J/(mol·K)	C_v J/(mol·K)	C_p J/(mol·K)	Sound m/s	Visc. μPas	Therm. W/(m·K)	Diel. Const.
90.680[a]	0.01174	451.6	28.15	-5728.0	-5728.0	67.94	34.02	53.46	1534.0	205.0	0.224	1.67701
90.680[b]	0.01174	0.2333	0.01579	2235.0	2978.0	164.0	25.19	33.79	249.1	3.64	0.00923	1.00031
92.0	0.01388	449.8	28.04	-5658.0	-5658.0	68.70	33.89	53.50	1523.0	197.0	0.222	1.67398
92.0	0.01388	0.2944	0.01835	2266.0	3022.0	163.1	25.21	33.84	250.8	3.69	0.00939	1.00036
94.0	0.01770	447.2	27.88	-5553.0	-5552.0	69.84	33.74	53.60	1506.0	185.0	0.218	1.66936
94.0	0.01770	0.3667	0.02286	2313.0	3087.0	161.7	25.25	33.94	253.2	3.76	0.00963	1.00045
96.0	0.02234	444.6	27.71	-5446.0	-5446.0	70.96	33.61	53.74	1489.0	175.0	0.215	1.66469
96.0	0.02234	0.4523	0.02819	2360.0	3152.0	160.5	25.29	34.05	255.6	3.84	0.00988	1.00055
98.0	0.02790	441.9	27.55	-5340.0	-5339.0	72.06	33.50	53.91	1470.0	165.0	0.212	1.65998
98.0	0.02790	0.5530	0.03447	2406.0	3215.0	159.3	25.33	34.17	257.9	3.92	0.0101	1.00067
100.0	0.03451	439.2	27.38	-5232.0	-5231.0	73.14	33.41	54.11	1452.0	157.0	0.208	1.65523
100.0	0.03451	0.6705	0.04180	2451.0	3277.0	158.2	25.38	34.30	260.2	4.00	0.0104	1.00082
102.0	0.04232	436.5	27.21	-5125.0	-5123.0	74.21	33.33	54.32	1433.0	149.0	0.205	1.65042
102.0	0.04232	0.8067	0.05029	2496.0	3338.0	157.2	25.43	34.45	263.3	4.08	0.0106	1.00098
104.0	0.05146	433.7	27.03	-5016.0	-5014.0	75.26	33.25	54.54	1413.0	141.0	0.201	1.64557
104.0	0.05146	0.9634	0.06005	2541.0	3397.0	156.1	25.49	34.61	264.4	4.16	0.0109	1.00118
106.0	0.06208	430.9	26.86	-4907.0	-4905.0	76.30	33.17	54.78	1394.0	134.0	0.198	1.64066
106.0	0.06208	1.143	0.07123	2584.0	3456.0	155.2	25.55	34.78	266.4	4.23	0.0112	1.00139
108.0	0.07434	428.1	26.68	-4798.0	-4795.0	77.32	33.09	55.03	1374.0	128.0	0.195	1.63571
108.0	0.07434	1.347	0.08394	2627.0	3513.0	154.3	25.62	34.98	268.3	4.31	0.0114	1.00164
110.0	0.08840	425.2	26.50	-4688.0	-4684.0	78.34	33.01	55.29	1353.0	122.0	0.192	1.63070
110.0	0.08840	1.577	0.09833	2670.0	3569.0	153.4	25.69	35.19	270.2	4.40	0.0117	1.00192
112.0	0.1044	422.3	26.32	-4577.0	-4573.0	79.33	32.92	55.56	1333.0	117.0	0.188	1.62563
112.0	0.1044	1.837	0.1145	2711.0	3623.0	152.5	25.76	35.41	271.9	4.48	0.0120	1.00224
114.0	0.1226	419.3	26.14	-4466.0	-4461.0	80.32	32.84	55.84	1312.0	111.0	0.185	1.62051
114.0	0.1226	2.129	0.1327	2752.0	3676.0	151.7	25.84	35.66	273.6	4.56	0.0123	1.00260
116.0	0.1431	416.4	25.95	-4354.0	-4348.0	81.30	32.75	56.14	1292.0	107.0	0.182	1.61532
116.0	0.1431	2.454	0.1529	2792.0	3728.0	150.9	25.93	35.93	275.2	4.64	0.0126	1.00300

Modified Benedict-Rubin-Webb EOS:

$$P = \rho RT + \rho^2[G(1)T + G(2)T^{1/2} + G(3) + G(4)/T + G(5)/T^2] + \rho^3[G(6)T + G(7) + G(8)/T + G(9)/T^2]$$
$$+ \rho^4[G(10)T + G(11) + G(12)/T] + \rho^5[G(13)] + \rho^6[G(14)/T + G(15)/T^2] + \rho^7[G(16)/T]$$
$$+ \rho^8[G(17)/T + G(18)/T^2] + \rho^9[G(19)/T^2] + \rho^3[G(20)/T^2 + G(21)/T^3]\exp(\gamma\rho^2)$$
$$+ \rho^5[G(22)/T^2 + G(23)/T^4]\exp(\gamma\rho^2) + \rho^7[G(24)/T^2 + G(25)/T^3]\exp(\gamma\rho^2)$$
$$+ \rho^9[G(26)/T^2 + G(27)/T^4]\exp(\gamma\rho^2) + \rho^{11}[G(28)/T^2 + G(29)/T^3]\exp(\gamma\rho^2)$$
$$+ \rho^{13}[G(30)/T^2 + G(31)/T^3 + G(32)/T^4]\exp(\gamma\rho^2) .$$

Younglove and Ely, *J. Phys. Chem. Ref. Data* (1987)

$G(1) = 0.9898937956 \times 10^{-5}$
$G(2) = 0.2199608275 \times 10^{-1}$
$G(3) = -0.5322788000$
$G(4) = 0.2021657962 \times 10^{2}$
$G(5) = -0.2234398926 \times 10^{4}$
$G(6) = 0.1067940280 \times 10^{-4}$
$G(7) = 0.1457922469 \times 10^{-3}$
$G(8) = -0.9265816666$
$G(9) = 0.2915364732 \times 10^{3}$
$G(10) = 0.2313546209 \times 10^{-6}$
$G(11) = 0.1387214274 \times 10^{-3}$
$G(12) = 0.4780467451 \times 10^{-2}$
$G(13) = 0.1176103833 \times 10^{-4}$
$G(14) = -0.1982096730 \times 10^{-3}$
$G(15) = -0.2512887756 \times 10^{-1}$
$G(16) = 0.9748899826 \times 10^{-5}$
$G(17) = -0.1202192137 \times 10^{-6}$
$G(18) = 0.4128353939 \times 10^{-4}$
$G(19) = -0.7215842918 \times 10^{-6}$
$G(20) = 0.5081738255 \times 10^{3}$
$G(21) = -0.9198903192 \times 10^{5}$
$G(22) = -0.2732264677 \times 10^{1}$
$G(23) = 0.7499024351 \times 10^{5}$
$G(24) = 0.1114060908 \times 10^{-2}$
$G(25) = 0.1083955159 \times 10^{1}$
$G(26) = -0.4490960312 \times 10^{-4}$
$G(27) = -0.1380337847 \times 10^{1}$
$G(28) = -0.2371902232 \times 10^{-7}$
$G(29) = 0.3761652197 \times 10^{-4}$
$G(30) = -0.2375166954 \times 10^{-9}$
$G(31) = -0.1237640790 \times 10^{-7}$
$G(32) = 0.6766926453 \times 10^{-6}$

Essential Definitions: Vectors, Matrices, Tensors

$\boldsymbol{x}$ Vector: (denoted in ***bold lowercase***) elements of a **Vector Space** V

V Vector Space: a set of elements $\mathcal{V}$ with the operations of addition and scalar multiplication, such that

$\mathcal{V} + \mathcal{V} \in \mathcal{V}$ a vector in the space plus another in the space yields another in the space

$\mathbb{R} \times \mathcal{V} \in \mathcal{V}$ a vector times a real number in the space yields a vector in the space

You are very familiar with Euclidean vectors in $\mathbb{R}^3$

$$\boldsymbol{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

or perhaps even $\mathbb{R}^n$
(tuples of n real numbers)

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

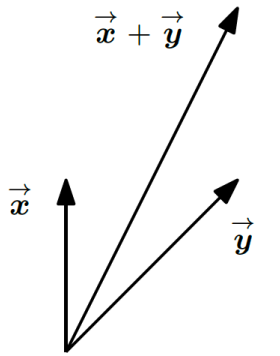
Essential Definitions: Vectors, Matrices, Tensors

$\boldsymbol{x}$ Vector: (denoted in ***bold lowercase***) elements of a **Vector Space** V

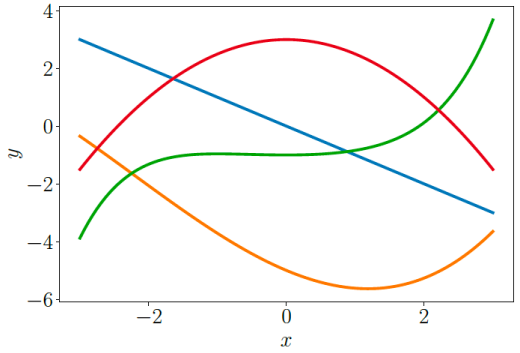
V Vector Space: a set of elements $\mathcal{V}$ with the operations of addition and scalar multiplication, such that

$\mathcal{V} + \mathcal{V} \in \mathcal{V}$ a vector in the space plus another in the space yields another in the space

$\mathbb{R} \times \mathcal{V} \in \mathcal{V}$ a vector times a real number in the space yields a vector in the space

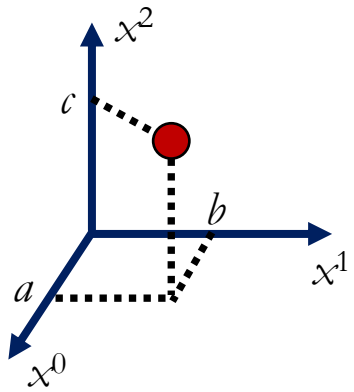


Euclidean vectors in $\mathbb{R}^2$



polynomials are also vectors but not in the Euclidean sense

Often it can be useful to make analogies to Euclidean vectors, since we have mathematical familiarity and intuition



$$f(x) = a + bx + cx^2$$

$$\boldsymbol{f} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

this relates to the concepts of *basis & coordinates* (to be revisited)

Essential Definitions: Vectors, Matrices, Tensors

Matrix:

$\boldsymbol{A}$

(denoted in ***BOLD UPPERCASE***)

For $m, n \in \mathbb{N}$

an (*m,n*) matrix is composed of an *mn*-tuple of elements arranged into *m* rows and *n* columns

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}, \quad a_{ij} \in \mathbb{R}$$

We can say that $\boldsymbol{A} \in \mathbb{R}^{m \times n}$

*(so can we say **A** lives in a vector space and is thus a vector? Sure!)*

Essential Definitions: Vectors, Matrices, Tensors

Matrix:

$\boldsymbol{A}$

(denoted in ***BOLD UPPERCASE***)

For $m, n \in \mathbb{N}$

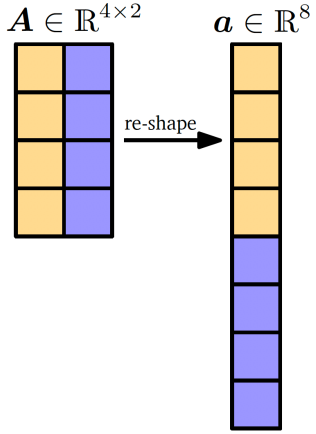
an (*m*,*n*) matrix is composed of an *mn*-tuple of elements arranged into *m* rows and *n* columns

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}, \quad a_{ij} \in \mathbb{R}$$

We can say that $\boldsymbol{A} \in \mathbb{R}^{m \times n}$

Matrices can be represented as vectors by stacking columns vertically or stacking rows horizontally (and this is really how they would be represented on machine memory)

$$\boldsymbol{A} \in \mathbb{R}^{m \times n} \rightarrow \boldsymbol{a} \in \mathbb{R}^{mn}$$



(*Aside:* Python stores elements of arrays in ROW-MAJOR order)

```
>>> import numpy as np
>>> a = np.array([[1,3,5],[2,4,6]])
>>> print(a)
[[1 3 5]
 [2 4 6]]
>>> print(a.flatten())
[1 3 5 2 4 6]
```

Essential Definitions: Vectors, Matrices, Tensors

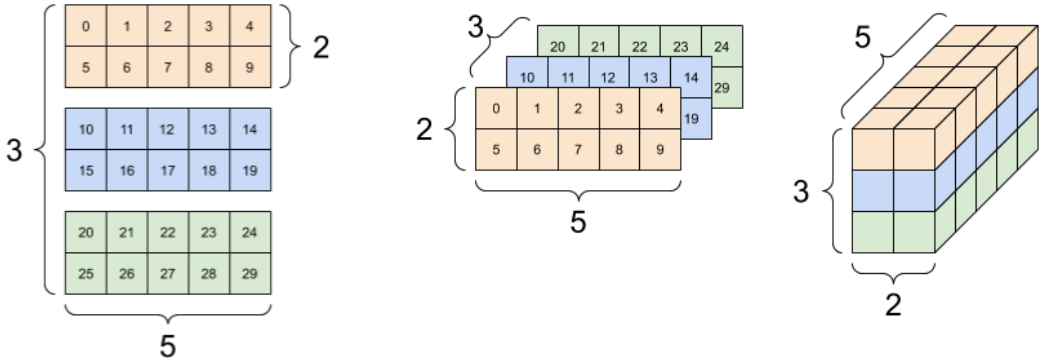
Tensor:

$$T, T_{ijk\ldots}$$

(denoted in ***BOLD UPPERCASE*** or explicit indices indicating rank)

- Mathematically, tensors are elements within the tensor product over a set of vector spaces
- Practically, tensors are objects that allow generalization of scalars, vectors, matrices, and higher-dimensional objects
- Tensors are defined by their ***rank*** (distinct from matrix rank)

rank-0 tensor → scalar
rank-1 tensor → vector
rank-2 tensor → matrix
rank-d tensor → array with d axes



rank-3 tensor with shape 3x2x5

In python, NumPy arrays are effectively Tensor-like containers

```
>>> rank0 = np.array(42).reshape(())
>>> rank1 = np.array(42).reshape((1))
>>> rank2 = np.array(42).reshape((1,1))
>>> rank3 = np.array(42).reshape((1,1,1))
>>> rank4 = np.array(42).reshape((1,1,1,1))
>>> for i in range(5):
...   print(locals()['rank{}'.format(i)])
...
42
[42]
[[42]]
[[[42]]]
[[[[42]]]]
>>> rank4[0,0,0,0]
42
```

```
>>> T = np.array([
  [[0, 1, 2, 3, 4],
   [5, 6, 7, 8, 9]],
  [[10, 11, 12, 13, 14],
   [15, 16, 17, 18, 19]],
  [[20, 21, 22, 23, 24],
   [25, 26, 27, 28, 29]]])
>>> print(T)
[[[ 0  1  2  3  4]
  [ 5  6  7  8  9]]

 [[10 11 12 13 14]
  [15 16 17 18 19]]

 [[20 21 22 23 24]
  [25 26 27 28 29]]]
>>> T[0]
array([[0, 1, 2, 3, 4],
       [5, 6, 7, 8, 9]])
```

Addition/Multiplication with Matrices

Addition

$$\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C} \in \mathbb{R}^{m \times n}$$

$$\boldsymbol{C} = \boldsymbol{A} + \boldsymbol{B}$$

$$c_{ij} = a_{ij} + b_{ij}$$

```
>>> A = np.array([[1,2],[3,4]])
>>> B = np.array([[8,7],[6,5]])
>>> C = A+B
>>> print(C)
[[9 9]
 [9 9]]
```

Scalar Multiplication

$$\boldsymbol{B} = \lambda \boldsymbol{A}, \lambda \in \mathbb{R}$$

$$b_{ij} = \lambda a_{ij}$$

Multiplication

$$\boldsymbol{A} \in \mathbb{R}^{m \times n} \qquad \boldsymbol{C} = \boldsymbol{A}\boldsymbol{B}$$

$$\boldsymbol{B} \in \mathbb{R}^{n \times k}$$

$$\boldsymbol{C} \in \mathbb{R}^{m \times k} \qquad c_{ij} = \sum_{l=1}^{n} a_{il} b_{lj}$$

Shorthand Einstein notation: any index appearing twice in the same term implies summation over that index

$$c_{ij} = a_{il} b_{lk}$$

```
>>> print(np.matmul(A,B))
[[20 17]
 [48 41]]
>>> print(np.dot(A,B))
[[20 17]
 [48 41]]
>>> print(np.einsum('il,lj->ij',A,B))
[[20 17]
 [48 41]]
>>> print(A@B)
[[20 17]
 [48 41]]
```

- matrix multiplication is not elementwise!

$$c_{ij} \neq a_{ij} b_{ij}$$

This is what you will get though in many programming languages just using the multiply operator. (Hadamard product)

- Order matters! ***AB ≠ BA*** *in general*

```
>>> print(np.dot(B,A))
[[29 44]
 [21 32]]
```

Matrix Inverse and Transpose

Let $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ and $\boldsymbol{B} \in \mathbb{R}^{n \times n}$

Suppose $\boldsymbol{AB} = \boldsymbol{I}_n = \boldsymbol{BA}$

Then $\boldsymbol{B} = \boldsymbol{A}^{-1}$ is the inverse of $\boldsymbol{A}$

$$\boldsymbol{I}_n := \begin{bmatrix} 1 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix} \in \mathbb{R}^{n \times n}$$

- *Not every matrix is invertible*
- *If it is invertible, then its inverse is unique*
- *The matrix **determinant** can be used to check if a square matrix is invertible*

$$\boldsymbol{A} := \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \in \mathbb{R}^{2 \times 2} \quad \boldsymbol{A}' := \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix}$$

$$\boldsymbol{A}\boldsymbol{A}' = \begin{bmatrix} a_{11}a_{22} - a_{12}a_{21} & 0 \\ 0 & a_{11}a_{22} - a_{12}a_{21} \end{bmatrix} = (a_{11}a_{22} - a_{12}a_{21})\boldsymbol{I}$$

$$\boldsymbol{A}^{-1} = \frac{1}{a_{11}a_{22} - a_{12}a_{21}} \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix}$$

Matrix Inverse and Transpose

Let $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ and $\boldsymbol{B} \in \mathbb{R}^{n \times n}$

Suppose $\boldsymbol{AB} = \boldsymbol{I}_n = \boldsymbol{BA}$

Then $\boldsymbol{B} = \boldsymbol{A}^{-1}$ is the inverse of ***A***

Let $\boldsymbol{A} \in \mathbb{R}^{n \times m}$ and $\boldsymbol{B} \in \mathbb{R}^{m \times n}$

Suppose $b_{ij} = a_{ji}$

Then $\boldsymbol{B} = \boldsymbol{A}^{T}$ is the transpose of ***A***

```
>>> print(A)
[[1 2 1]
 [4 4 5]
 [6 7 7]]
>>> np.linalg.inv(A)
array([[-7., -7.,  6.],
       [ 2.,  1., -1.],
       [ 4.,  5., -4.]])
>>> B = np.linalg.inv(A)
>>> print(B)
[[-7. -7.  6.]
 [ 2.  1. -1.]
 [ 4.  5. -4.]]
>>> print(np.linalg.inv(B))
[[1. 2. 1.]
 [4. 4. 5.]
 [6. 7. 7.]]
```

```
>>> print(A.transpose())
[[1 4 6]
 [2 4 7]
 [1 5 7]]
```

```
>>> print(A.T)
[[1 4 6]
 [2 4 7]
 [1 5 7]]
```

Let's go practice in Python to convince ourselves of the following useful properties:

$$\boldsymbol{A}\boldsymbol{A}^{-1} = \boldsymbol{I} = \boldsymbol{A}^{-1}\boldsymbol{A}$$

$$(\boldsymbol{AB})^{-1} = \boldsymbol{B}^{-1}\boldsymbol{A}^{-1}$$

$$(\boldsymbol{A} + \boldsymbol{B})^{-1} \neq \boldsymbol{A}^{-1} + \boldsymbol{B}^{-1}$$

$$(\boldsymbol{A}^{\top})^{\top} = \boldsymbol{A}$$

$$(\boldsymbol{A} + \boldsymbol{B})^{\top} = \boldsymbol{A}^{\top} + \boldsymbol{B}^{\top}$$

$$(\boldsymbol{AB})^{\top} = \boldsymbol{B}^{\top}\boldsymbol{A}^{\top}$$

Systems of Linear Equations

Consider the following system of linear equations

$$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1$$

$$\vdots$$

$$a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m$$

every vector $\boldsymbol{x} = [x_1, \ldots, x_n]^T$ satisfying the system is a solution

Any given system of equations may have

- **no** solution
- **one** unique solution
- **infinite**ly many solutions

tangible example:

- A chemical manufacturing company has access to 4, 3, and 3 units of chemical ***A***, ***B***, *and* ***C***;
- These base chemicals can be used in the synthesis of x_j units *of* downstream products (j= 1,2,3)
- One unit of chemical **A** is required for **each product**
- A unit of chemical **B** is necessary in **product 1** and **2** but unneeded for 3
- One unit of chemical **C** is needed for **product 2** and two units are needed for **product 3**; none are needed for **product 1**

Discuss:

What is the optimal production plan to ensure most complete utilization of resources?

What modifications might we practically consider in a realistic planning scenario?

Matrix Representation of Linear Equations

Consider the following system of linear equations

$$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1$$

$$\vdots$$

$$a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m$$

$$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}$$

$$[m \times n]\,[n \times 1] = [m \times 1]$$

$$\boldsymbol{Ax = b}$$

A set of linear equations can be compactly expressed in Linear algebraic terms and exploited to find solutions (in lieu of the intuition or systematic solution used previously)

Solution set is invariant with respect to ***elementary transformations***:

- *Exchanging two equations (i.e., swapping rows)*
- *Multiplying an equation by a non-zero constant*
- *Adding two equations (rows)*

General solutions can be represented as

- *A particular solution*
- *… plus all solutions to **Ax = 0***

Row Echelon Form & Gaussian Elimination

Obtaining solutions is “straightforward” by rearranging matrices into (Reduced) Row-Echelon Form via Gaussian Elimination

$$\begin{array}{rcrcrcrcr}
-2x_1 & + & 4x_2 & - & 2x_3 & - & x_4 & + & 4x_5 & = & -3 \\
4x_1 & - & 8x_2 & + & 3x_3 & - & 3x_4 & + & x_5 & = & 2 \\
x_1 & - & 2x_2 & + & x_3 & - & x_4 & + & x_5 & = & 0 \\
x_1 & - & 2x_2 & & & - & 3x_4 & + & 4x_5 & = & a
\end{array}$$

convert to “augmented” matrix form

$$\left[\begin{array}{rrrrr|r}
-2 & 4 & -2 & -1 & 4 & -3 \\
4 & -8 & 3 & -3 & 1 & 2 \\
1 & -2 & 1 & -1 & 1 & 0 \\
1 & -2 & 0 & -3 & 4 & a
\end{array}\right]$$

Row Echelon Form & Gaussian Elimination

Obtaining solutions is “straightforward” by rearranging matrices into (Reduced) Row-Echelon Form via Gaussian Elimination

$$\begin{array}{rcrcrcrcr} -2x_1 & + & 4x_2 & - & 2x_3 & - & x_4 & + & 4x_5 & = & -3 \\ 4x_1 & - & 8x_2 & + & 3x_3 & - & 3x_4 & + & x_5 & = & 2 \\ x_1 & - & 2x_2 & + & x_3 & - & x_4 & + & x_5 & = & 0 \\ x_1 & - & 2x_2 & & & - & 3x_4 & + & 4x_5 & = & a \end{array}$$

convert to “augmented” matrix form

$$\left[\begin{array}{rrrrr|r} -2 & 4 & -2 & -1 & 4 & -3 \\ 4 & -8 & 3 & -3 & 1 & 2 \\ 1 & -2 & 1 & -1 & 1 & 0 \\ 1 & -2 & 0 & -3 & 4 & a \end{array}\right]$$

swap

Now we want to use some equations to systematically eliminate variables (i.e., achieve leading zeros) from other equations (i.e., rows)

$$\left[\begin{array}{rrrrr|r} 1 & -2 & 1 & -1 & 1 & 0 \\ 4 & -8 & 3 & -3 & 1 & 2 \\ -2 & 4 & -2 & -1 & 4 & -3 \\ 1 & -2 & 0 & -3 & 4 & a \end{array}\right]$$

subtract 4R1

add 2R1

subtract 1R1

Row Echelon Form & Gaussian Elimination

Obtaining solutions is “straightforward” by rearranging matrices into (Reduced) Row-Echelon Form via Gaussian Elimination

$$\begin{array}{rcrcrcrcr} -2x_1 & + & 4x_2 & - & 2x_3 & - & x_4 & + & 4x_5 & = & -3 \\ 4x_1 & - & 8x_2 & + & 3x_3 & - & 3x_4 & + & x_5 & = & 2 \\ x_1 & - & 2x_2 & + & x_3 & - & x_4 & + & x_5 & = & 0 \\ x_1 & - & 2x_2 & & & - & 3x_4 & + & 4x_5 & = & a \end{array}$$

convert to “augmented” matrix form

$$\left[\begin{array}{rrrrr|r} -2 & 4 & -2 & -1 & 4 & -3 \\ 4 & -8 & 3 & -3 & 1 & 2 \\ 1 & -2 & 1 & -1 & 1 & 0 \\ 1 & -2 & 0 & -3 & 4 & a \end{array}\right]$$

Now we want to use some equations to systematically eliminate variables (i.e., achieve leading zeros) from other equations (i.e., rows)

$$\left[\begin{array}{rrrrr|r} 1 & -2 & 1 & -1 & 1 & 0 \\ 4 & -8 & 3 & -3 & 1 & 2 \\ -2 & 4 & -2 & -1 & 4 & -3 \\ 1 & -2 & 0 & -3 & 4 & a \end{array}\right]$$

subtract 4R1

add 2R1

subtract 1R1

$$\left[\begin{array}{rrrrr|r} 1 & -2 & 1 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 & -3 & 2 \\ 0 & 0 & 0 & -3 & 6 & -3 \\ 0 & 0 & -1 & -2 & 3 & a \end{array}\right]$$

subtract (R2+R3) from R4

$$\left[\begin{array}{rrrrr|r} 1 & -2 & 1 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 & -3 & 2 \\ 0 & 0 & 0 & -3 & 6 & -3 \\ 0 & 0 & 0 & 0 & 0 & a+1 \end{array}\right]$$

divide R3 by -3

$$\left[\begin{array}{rrrrr|r} 1 & -2 & 1 & -1 & 1 & 0 \\ 0 & 0 & 1 & -1 & 3 & -2 \\ 0 & 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 0 & a+1 \end{array}\right]$$

Row Echelon Form & Gaussian Elimination

Obtaining solutions is “straightforward” by rearranging matrices into (Reduced) Row-Echelon Form via Gaussian Elimination

$$\lambda_1 \begin{bmatrix}1\\0\\0\\0\end{bmatrix} + \lambda_2 \begin{bmatrix}1\\1\\0\\0\end{bmatrix} + \lambda_3 \begin{bmatrix}-1\\-1\\1\\0\end{bmatrix} = \begin{bmatrix}0\\-2\\1\\0\end{bmatrix} \Rightarrow \boldsymbol{x}^* = \begin{bmatrix}2\\0\\-1\\1\\0\end{bmatrix}$$

Particular solutions can be obtained by working with basic variables (easiest to work with rightmost to left)

$$\left[\begin{array}{ccccc|c} 1 & -2 & 1 & -1 & 1 & 0 \\ 0 & 0 & 1 & -1 & 3 & -2 \\ 0 & 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 0 & a+1 \end{array}\right]$$

Row-echelon Form

- *Any rows with only zeros are at the bottom*
- *First nonzero number from the left == pivot*
- *all pivots are strictly to the right of pivots above*
  → “staircase structure”
- *basic variables == pivot columns*
- *free variables == non-pivot columns*

Row Echelon Form & Gaussian Elimination

Obtaining solutions is “straightforward” by rearranging matrices into (Reduced) Row-Echelon Form via Gaussian Elimination

For the **general solution,** we focus on non-pivot columns and how they can be expressed as linear combinations of pivot columns (on their left), to satisfy ***Ax*** = **0**

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

1. R2 + R3 → R2
2. R1+R3-R2→R1

$$\boldsymbol{x} = \begin{bmatrix} 2 \\ 0 \\ -1 \\ 1 \\ 0 \end{bmatrix} + \lambda_1 \begin{bmatrix} 2 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \lambda_2 \begin{bmatrix} 2 \\ 0 \\ -1 \\ 2 \\ 1 \end{bmatrix}$$

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

Row-echelon Form

- *Any rows with only zeros are at the bottom*
- *First nonzero number from the left == pivot*
  all pivots are strictly to the right of pivots above
  → “staircase structure”
- *basic variables == pivot columns*
- *free variables == non-pivot columns*

Reduced Row-echelon Form

- *Row-echelon form*
- *All pivots are 1*
- *Pivot is the only nonzero entry in its column*

Practical Trick for homogeneous solutions

A is *m*x*n*… if we extend it to *n*x*n* by adding *n-m* rows of [0 … 0 -1 0 … 0] such that 1 or -1 populates the diagonal…

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

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

$$\boldsymbol{x} = \begin{bmatrix} 2 \\ 0 \\ -1 \\ 1 \\ 0 \end{bmatrix} + \lambda_1 \begin{bmatrix} 2 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \lambda_2 \begin{bmatrix} 2 \\ 0 \\ -1 \\ 2 \\ 1 \end{bmatrix}$$

We can simply read the homogeneous solutions from the non-pivot columns

Calculating the inverse of a matrix

Computing the inverse of a matrix is equivalent to solving a system of linear equations

$$\boldsymbol{AX} = \boldsymbol{I}_n \rightarrow [\boldsymbol{A}|\boldsymbol{I}_n] \rightarrow \cdots \rightarrow [\boldsymbol{I}_n|\boldsymbol{A}^{-1}]$$

Example

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 2 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 2 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \rightarrow \left[\begin{array}{cccc|cccc} 1 & 0 & 2 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 2 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{array}\right]$$

$$\downarrow$$

$$\left[\begin{array}{cccc|cccc} 1 & 0 & 0 & 0 & -1 & 2 & -2 & 2 \\ 0 & 1 & 0 & 0 & 1 & -1 & 2 & -2 \\ 0 & 0 & 1 & 0 & 1 & -1 & 1 & -1 \\ 0 & 0 & 0 & 1 & -1 & 0 & -1 & 2 \end{array}\right] \rightarrow \boldsymbol{A}^{-1} = \begin{bmatrix} -1 & 2 & -2 & 2 \\ 1 & -1 & 2 & -2 \\ 1 & -1 & 1 & -1 \\ -1 & 0 & -1 & 2 \end{bmatrix}$$

Calculating the inverse of a matrix

Computing the inverse of a matrix is equivalent to solving a system of linear equations

$$\boldsymbol{AX} = \boldsymbol{I}_n \rightarrow \left[\boldsymbol{A}|\boldsymbol{I}_n\right] \rightarrow \cdots \rightarrow \left[\boldsymbol{I}_n|\boldsymbol{A}^{-1}\right]$$

Example

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 2 & \\ 1 & 1 & 0 & \\ 1 & 2 & 0 & \\ 1 & 1 & 1 & \end{bmatrix}$$

```
>>> A = np.array([[1,0,2,0],[1,1,0,0],[1,2,0,1],[1,1,1,1]])
>>> print(A)
[[1 0 2 0]
 [1 1 0 0]
 [1 2 0 1]
 [1 1 1 1]]
>>> np.linalg.inv(A)
array([[-1.,  2., -2.,  2.],
       [ 1., -1.,  2., -2.],
       [ 1., -1.,  1., -1.],
       [-1.,  0., -1.,  2.]])
>>> import scipy.linalg as spla
>>> spla.inv(A)
array([[-1.,  2., -2.,  2.],
       [ 1., -1.,  2., -2.],
       [ 1., -1.,  1., -1.],
       [-1.,  0., -1.,  2.]])
```

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

Matrix Method for Solving a Linear system

We can generally use matrices to solve linear systems expressed as $\boldsymbol{Ax} = \boldsymbol{b}$

*…if the columns in **A** are **linearly independent**, then*

$$\boldsymbol{Ax} = \boldsymbol{b} \Leftrightarrow \boldsymbol{A}^T\boldsymbol{Ax} = \boldsymbol{A}^T\boldsymbol{b} \Leftrightarrow \boldsymbol{x} = \underbrace{(\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T}\boldsymbol{b}$$

Moore-Penrose pseudo-inverse

*… if **A** is square and **invertible**, then simply*

$$(\boldsymbol{AB})^{-1} = \boldsymbol{B}^{-1}\boldsymbol{A}^{-1}$$

$$\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b}$$

Linear Independence and Basis Sets

Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k \in V$ If $\mathbf{0} = \sum_{i=1}^{k} \lambda_i \boldsymbol{x}_i$ is satisfied by only the trivial solution, then the ***x*** are linearly independent

Use Gaussian elimination to convert the system into row-echelon form:

- *Pivot columns represent vectors that are linearly independent from the vectors to the left*
- *Non-pivot columns are linear combinations of those to the left*

If $\forall\, \boldsymbol{v} \in V, \exists\, \lambda_i \text{ s.t. } \boldsymbol{v} = \sum_{i=1}^{k} \lambda_i \boldsymbol{x}_i$ **and** the $\boldsymbol{x}_i$ are linearly independent

then we can call $\mathcal{B} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k\}$ a basis for *V*

e.g.,

for $\mathbb{R}^3$ the canonical/standard basis is $\mathcal{B} = \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\}$

Rank and Invertibility

In a matrix ***A***, the number of linearly independent columns equals the number of linearly independent rows, and we call this the ***Rank***

Two important properties:

$$\forall\ \boldsymbol{A}\ \in\ \mathbb{R}^{m\times n}\ \text{and}\ \boldsymbol{b}\in\mathbb{R}^{m},$$
$$\exists \boldsymbol{x}\ \text{s.t}\ \boldsymbol{A}\boldsymbol{x}=\boldsymbol{b},\ \text{iff}\ \text{rank}(\boldsymbol{A})=\text{rank}(\boldsymbol{A}|\boldsymbol{b})$$

*"A solution exists only if **b** exists in the subspace spanned by the columns of **A**"*

$$\forall\ \boldsymbol{A}\ \in\ \mathbb{R}^{n\times n}, \exists \boldsymbol{A}^{-1}\ \text{iff}\ \text{rank}(\boldsymbol{A})=n$$

"Invertible square matrices (regular) possess linearly independent columns"

How can we tell what the rank is for a matrix?

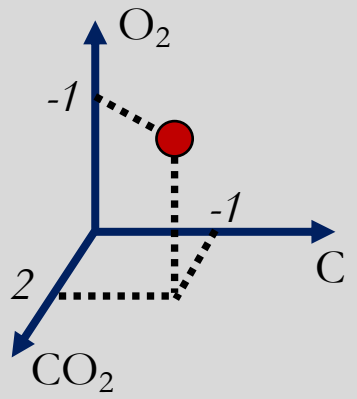
A Chemically motivated Example

Consider the following “goal gasification” reactions observed during steam reformation of coal for the eventual production of syngas

- $C + H_2O \rightleftharpoons CO + H_2$
- $C + 2H_2O \rightleftharpoons CO_2 + H2$
- $C + 2H_2 \rightleftharpoons CH_4$
- $C + CO_2 \rightleftharpoons 2CO$
- $CO + H_2O \rightleftharpoons CO_2 + H_2$

We can represent reactions as linear combinations of molecules, with molecules or fragments representing a basis in chemical space

$C+O_2 \rightarrow CO_2$



How many independent chemical reactions characterize the system?

$$\mathcal{B} = \{\boldsymbol{e}_{\mathrm{C}}, \boldsymbol{e}_{\mathrm{H_2O}}, \boldsymbol{e}_{\mathrm{CO}}, \boldsymbol{e}_{\mathrm{H_2}}, \boldsymbol{e}_{\mathrm{CO_2}}, \boldsymbol{e}_{\mathrm{CH_4}}\}$$

$$\boldsymbol{A} = \begin{bmatrix} -1 & -1 & 1 & 1 & 0 & 0 \\ -1 & -2 & 0 & 2 & 1 & 0 \\ -1 & 0 & 0 & -2 & 0 & 1 \\ -1 & 0 & 2 & -1 & 0 & 0 \\ 0 & -1 & -1 & 1 & 1 & 0 \end{bmatrix}$$

A Chemically motivated Example

Consider the following “goal gasification” reactions observed during steam reformation of coal for the eventual production of syngas

- $C + H_2O \rightleftharpoons CO + H_2$
- $C + 2H_2O \rightleftharpoons CO_2 + H2$
- $C + 2H_2 \rightleftharpoons CH_4$
- $C + CO_2 \rightleftharpoons 2CO$
- $CO + H_2O \rightleftharpoons CO_2 + H_2$

```
>>> print(A)
[[-1 -1  1  1  0  0]
 [-1 -2  0  2  1  0]
 [-1  0  0 -2  0  1]
 [-1  0  2  0 -1  0]
 [ 0 -1 -1  1  1  0]]
>>> np.linalg.matrix_rank(A)
3
```

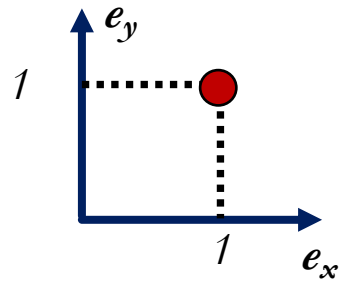
- $C + H_2O \rightleftharpoons CO + H_2$
- $CO + H_2O \rightleftharpoons CO_2 + H2$
- $2CO + H_2 \rightleftharpoons CO_2 + CH_4$

$$\mathcal{B} = \{\boldsymbol{e}_{\mathrm{C}}, \boldsymbol{e}_{\mathrm{H_2O}}, \boldsymbol{e}_{\mathrm{CO}}, \boldsymbol{e}_{\mathrm{H_2}}, \boldsymbol{e}_{\mathrm{CO_2}}, \boldsymbol{e}_{\mathrm{CH_4}}\}$$

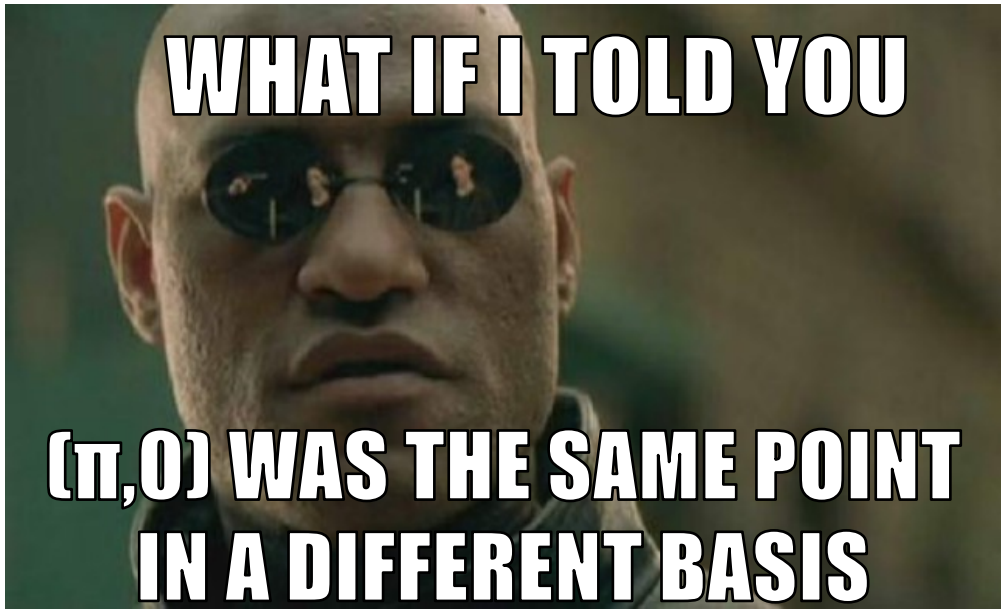
$$\boldsymbol{A} = \begin{bmatrix} -1 & -1 & 1 & 1 & 0 & 0 \\ -1 & -2 & 0 & 2 & 1 & 0 \\ -1 & 0 & 0 & -2 & 0 & 1 \\ -1 & 0 & 2 & -1 & 0 & 0 \\ 0 & -1 & -1 & 1 & 1 & 0 \end{bmatrix} \Rightarrow \boldsymbol{A} = \begin{bmatrix} -1 & -1 & 1 & 1 & 0 & 0 \\ 0 & -1 & -1 & 1 & 1 & 0 \\ 0 & 0 & -2 & -2 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

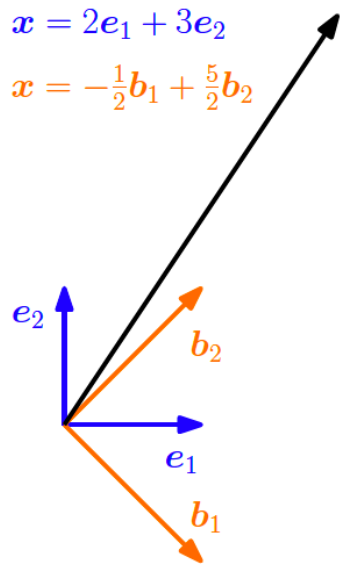
Coordinate Transformations/Linear Mappings

The concept of a basis is also fundamentally connected to our notion of a coordinate; a coordinate identifies our position within a vector space spanned by a particular basis



the point is defined by the tuple (1,1)





Changing coordinates within a given vector space can be achieved via matrix multiplication

$$\boldsymbol{T} = \begin{bmatrix} \pi & 0 \\ -1 & 1 \end{bmatrix} \qquad \boldsymbol{v} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \qquad \boldsymbol{Tv} = \begin{bmatrix} \pi \\ 0 \end{bmatrix}$$

transformation matrix — *coordinate in "old" basis* — *coordinate in "new" basis*

Coordinate Transformations/Linear Mappings

This operation is formally defined through a "linear mapping," Φ which can transform between vector spaces

- *A linear mapping preserves the structure of the origin vector space if*

$$\forall \boldsymbol{x}, \boldsymbol{y} \in V \, \forall \lambda, \psi \in \mathbb{R} : \Phi(\lambda \boldsymbol{x} + \psi \boldsymbol{y}) = \lambda \Phi(\boldsymbol{x}) + \psi \Phi(\boldsymbol{y})$$

Given V W

$$\mathcal{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n) \quad \mathcal{C} = (\boldsymbol{c}_1, \ldots, \boldsymbol{c}_m)$$

A mapping from V to W given by

$$\Phi(\boldsymbol{b}_j) = \sum_{i=1}^{m} \alpha_{ij} \boldsymbol{c}_i$$

$$\boldsymbol{B} = [\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n]$$

$$\boldsymbol{v} = \sum_{i=1}^{n} \alpha_i \boldsymbol{b}_i \qquad \boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} \in \mathbb{R}^n$$

*The coordinates of the transformed $\boldsymbol{b}_j$ with respect to the basis C are in the j^{th} column of **T***

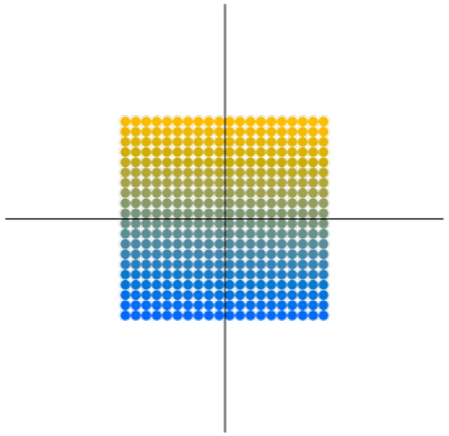
$$\boldsymbol{T}_\Phi(i, j) = \alpha_{ij}$$

$$\boldsymbol{v}_W = \boldsymbol{T}_\Phi \boldsymbol{v}_V$$

Coordinate Transformations/Linear Mappings

Which matrices lead to which coordinate mappings?

original data



A

$$\begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$$

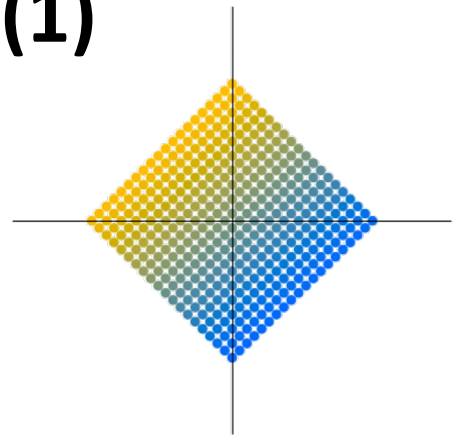
B

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

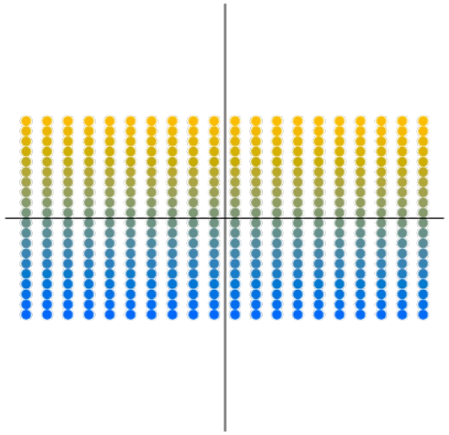
C

$$\begin{bmatrix} \cos\left(\frac{\pi}{4}\right) & -\sin\left(\frac{\pi}{4}\right) \\ \sin\left(\frac{\pi}{4}\right) & \cos\left(\frac{\pi}{4}\right) \end{bmatrix}$$

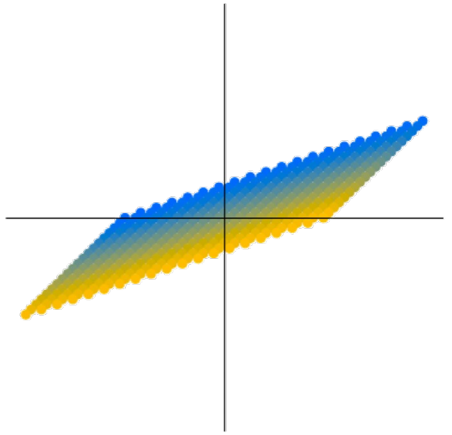
(1)



(2)



(3)



Changing Bases

Often representation and manipulation of data can be easier in a particular basis motivating a change from one to another

Bases in *V*

$$B = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n), \quad \tilde{B} = (\tilde{\boldsymbol{b}}_1, \ldots, \tilde{\boldsymbol{b}}_n)$$

Bases in *W*

$$C = (\boldsymbol{c}_1, \ldots, \boldsymbol{c}_m), \quad \tilde{C} = (\tilde{\boldsymbol{c}}_1, \ldots, \tilde{\boldsymbol{c}}_m)$$

Transformation matrices

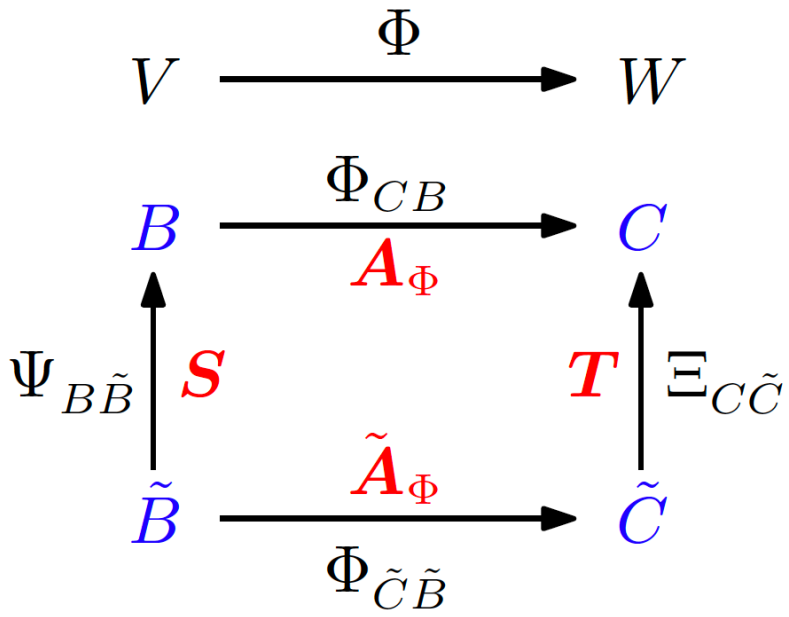
$$\boldsymbol{S} \in \mathbb{R}^{n \times n} \qquad \tilde{B} \to B$$

$$\boldsymbol{T} \in \mathbb{R}^{m \times m} \qquad \tilde{C} \to C$$

$$\boldsymbol{A}_\Phi \in \mathbb{R}^{m \times n} \qquad B \to C$$

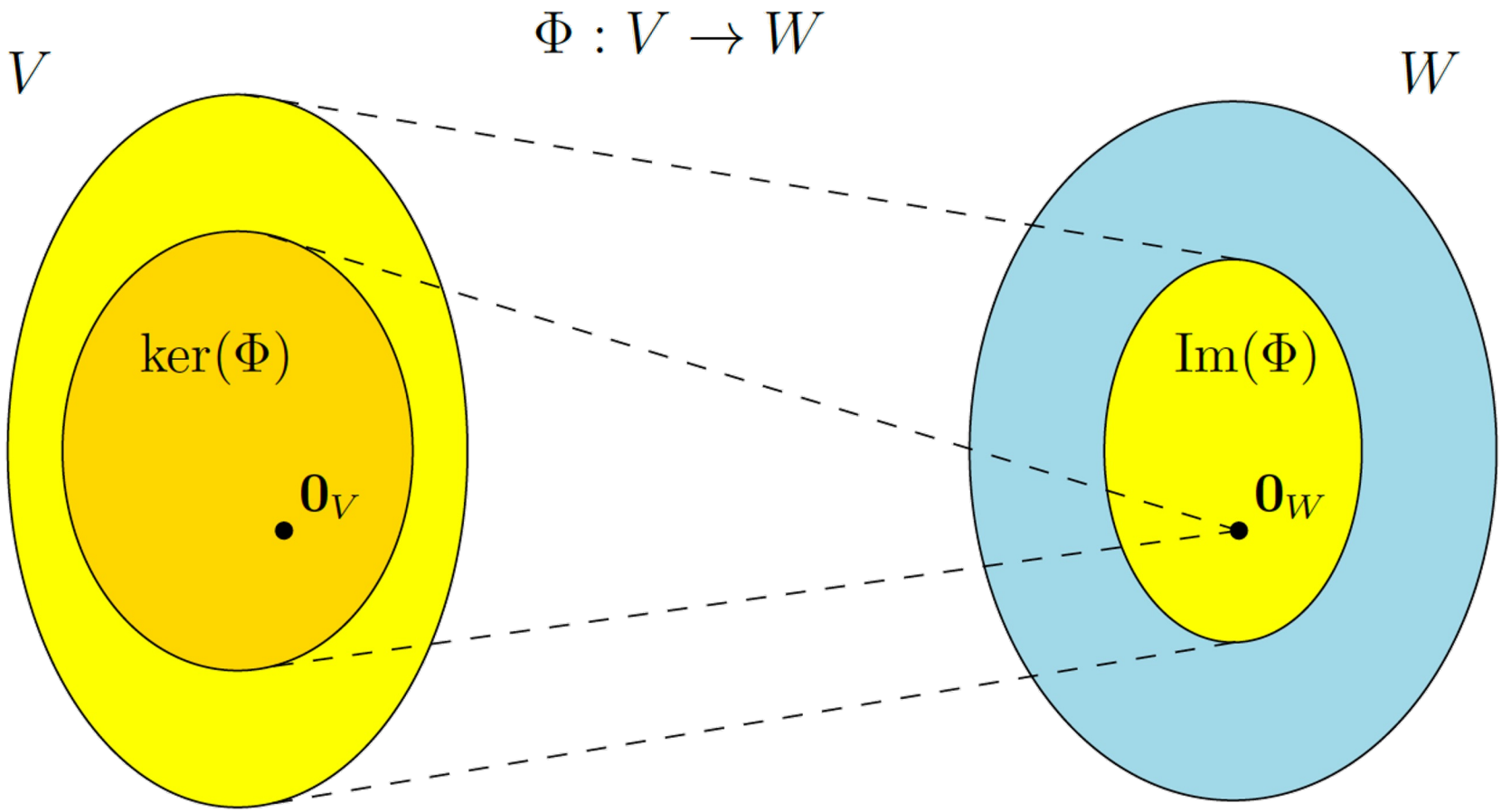
$$\tilde{\boldsymbol{A}}_\Phi \in \mathbb{R}^{m \times n} \qquad \tilde{B} \to \tilde{C}$$

$$\tilde{\boldsymbol{A}}_\Phi = \boldsymbol{T}^{-1} \boldsymbol{A}_\Phi \boldsymbol{S}$$



e.g., Useful for data compression while minimizing loss

Image and Kernels in Mappings



- ***Kernel*** (also “null space”) is the set of vectors in *V* that project to the “neutral” element in *W*
- ***Image*** is the set of vectors in *W* that can be reached from *V* through the mapping
- A one-to-one mapping is achievable only if the kernel is the zero vector

Eigenvectors and Eigenvalues

For $\boldsymbol{A} \in \mathbb{R}^{n\times n}, \lambda \in \mathbb{R}, \boldsymbol{x} \in \mathbb{R}^n \backslash \{\mathbf{0}\}$

$$\boldsymbol{A}\boldsymbol{x} = \lambda \boldsymbol{x}$$

eigenvalue *corresponding* ***eigenvector***

Equivalently, $(\boldsymbol{A} - \lambda \boldsymbol{I}_n)\,\boldsymbol{x} = 0$ can be solved non-trivially

which also implies $\det(\boldsymbol{A} - \lambda \boldsymbol{I}_n) = 0$

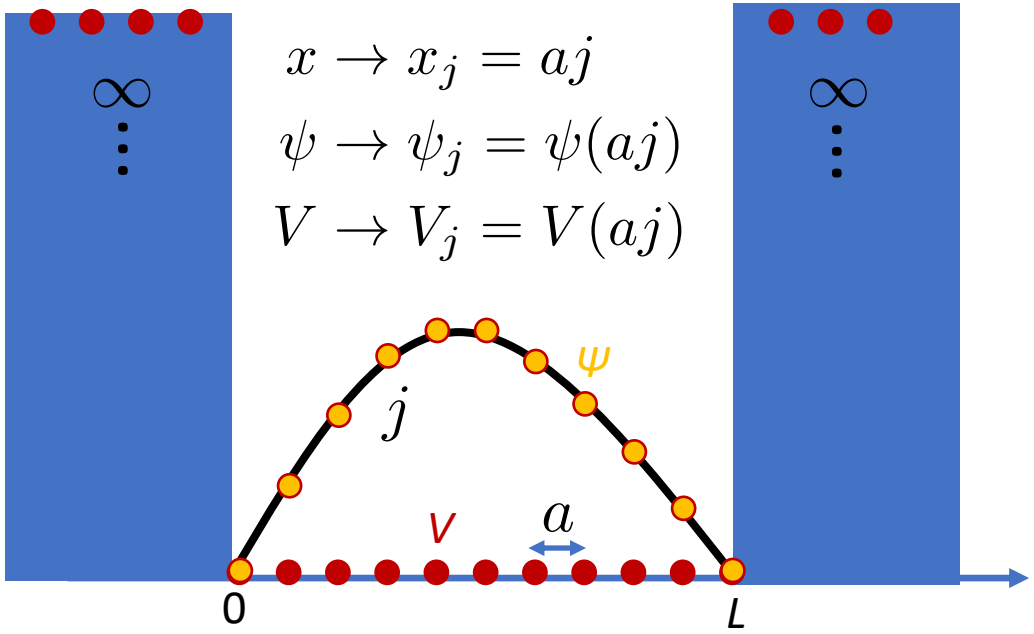
(characteristic polynomial)

$$p_{\mathbf{A}}(\lambda) := \det(\boldsymbol{A} - \lambda \boldsymbol{I}_n) = (-1)^n \lambda^n + \sum_{i=0}^{n-1} c_i \lambda^i$$

$$c_0 = \det(\boldsymbol{A})\,,$$
$$c_{n-1} = (-1)^{n-1}\mathrm{tr}(\boldsymbol{A})$$

This implies that λ is an eigenvalue iff it is a root of the characteristic polynomial

Eigenvectors and Eigenvalues



$$\frac{d^2\psi}{dx^2} + \frac{2m\varepsilon}{\hbar^2}\psi = 0$$

Using

$$\frac{d^2\psi(x_j)}{dx^2} \approx \frac{\psi_{j+1} - 2\psi_j + \psi_{j-1}}{a^2} \quad \text{and} \quad t \equiv \frac{\hbar^2}{2ma^2}$$

At each point j,

$$\hat{H}\psi_j = (V_j + 2t)\psi_j - t\psi_{j-1} - t\psi_{j+1} = \varepsilon\psi_j$$

This defines *a system of linear equations*, and the eigenvalues/eigenvectors == energies/wavefunctions!

$$\begin{pmatrix} 2t+V_0 & -t & 0 & 0 & 0 \cdots \\ -t & 2t+V_1 & -t & 0 & 0 \cdots \\ 0 & -t & 2t+V_2 & -t & 0 \cdots \\ 0 & 0 & -t & 2t+V_3 & -t \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix} \begin{pmatrix} \psi_0 \\ \psi_1 \\ \psi_2 \\ \psi_3 \\ \vdots \end{pmatrix} = \varepsilon \begin{pmatrix} \psi_0 \\ \psi_1 \\ \psi_2 \\ \psi_3 \\ \vdots \end{pmatrix}$$

$H \longrightarrow$ a matrix w/ elements H_{ij}

The elements of this matrix are given as:

$$H_{ij} = (V_i + 2t)\delta_{ij} - t\delta_{i,j-1} - t\delta_{i,j+1}$$

δ_{ij} is the Kronecker delta function

Eigenvectors and Eigenvalues

Useful properties:

- ***A*** *and* $\boldsymbol{A}^T$ *have the same eigenvalues (but not necessarily the same eigenvectors)*
- *The eigenspace of* ***A*** *is the null space or kernel of* $\boldsymbol{A}-\lambda\boldsymbol{I}$
- *Eigenvalues are independent of choice of basis*
- *Symmetric, positive definite matrices always have positive, real eigenvalues*
- *The determinant of a matrix is the product of its eignevalues*
- *The trace of a matrix is the sum of its eigenvalues*

n distinct eigenvalues implies ***n*** linearly independent eigenvectors → basis

Algebraic multiplicity of λ_i: the degeneracy/number of times λ_i is a root to the characteristic polynomial
Geometric multiplicity of λ_i : how many linearly independent eigenvectors are associated with λ_i

Eigenvectors and Eigenvalues

Useful properties:

- ***A*** *and* $\boldsymbol{A}^T$ *have the same eigenvalues (but not necessarily the same eigenvectors)*
- *The eigenspace of* ***A*** *is the null space or kernel of* $\boldsymbol{A}-\lambda\boldsymbol{I}$
- *Eigenvalues are independent of choice of basis*
- *Symmetric, positive definite matrices always have positive, real eigenvalues*
- *The determinant of a matrix is the product of its eigenvalues*
- *The trace of a matrix is the sum of its eigenvalues*

n distinct eigenvalues implies ***n*** linearly independent eigenvectors → basis

Algebraic multiplicity of λ_i: the degeneracy/number of times λ_i is a root to the characteristic polynomial
Geometric multiplicity of λ_i : how many linearly independent eigenvectors are associated with λ_i

Exercise: *determine the algebraic and geometric multiplicity associated with these matrices*

$$\underset{\text{(Jan.–Jun.)}}{\boldsymbol{A}_1} = \begin{bmatrix} 4 & 2 \\ 1 & 3 \end{bmatrix} \qquad \underset{\text{Jul.–Dec.)}}{\boldsymbol{A}_2} = \begin{bmatrix} 2 & 1 \\ 0 & 2 \end{bmatrix}$$

Matrix Decompositions

Matrix decompositions can enable faster/facile computations and are thus central to many numerical implementations underlying ML algorithms

- *If **A** is symmetric, positive definite*

Cholesky Decomposition $\boldsymbol{A = LL^T}$

$$\begin{bmatrix} l_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ l_{n1} & \cdots & l_{nn} \end{bmatrix}$$

Cholesky factor, a lower-triangular matrix with positive diagonal elements

A is symmetric positive definite if

$$\forall \boldsymbol{x} \in V \backslash \{\mathbf{0}\} : \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} > 0$$

- *If **A** is square*

Eigendecomposition $\boldsymbol{A = PDP^{-1}}$

- ***D** is a diagonal matrix with eigenvalue entries*
- ***P** has matching eigenvectors as its columns*

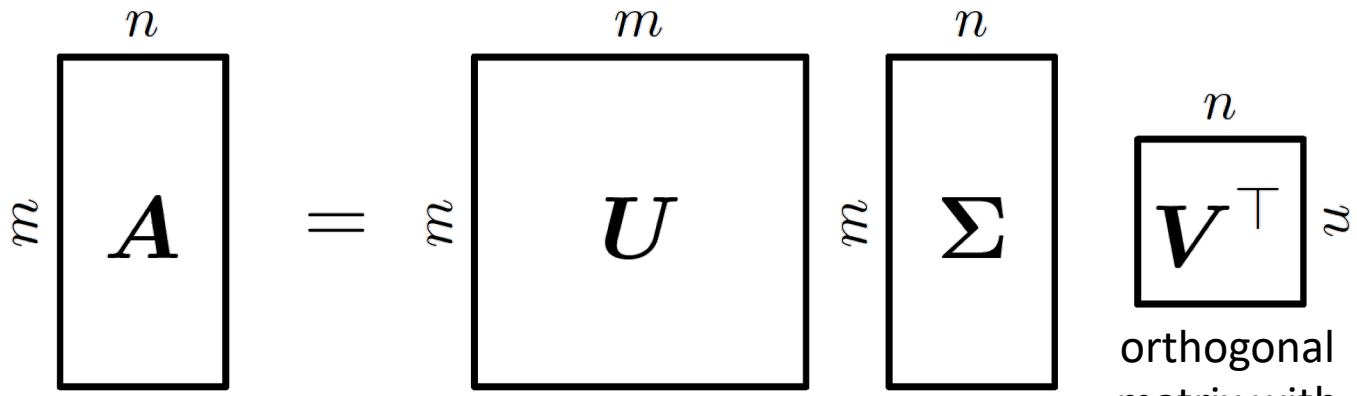
Spectral Theorem: If ***A*** is symmetric, there exists an orthonormal basis in the vector space of its eigenvectors, which have all real eigenvalues

$$\implies \boldsymbol{A = PDP^T}$$

Matrix Decompositions

Matrix decompositions can enable faster/facile computations and are thus central to many numerical implementations underlying ML algorithms

Singular value decomposition
(***SVD***, the never-fail decomposition)**:**



orthogonal matrix with vectors $\boldsymbol{u}_i$

left-singular vectors

diagonal matrix with entries $\Sigma_{ii} = \boldsymbol{\sigma}_i \geq 0$

singular values

orthogonal matrix with vectors $\boldsymbol{v}_i$

right-singular vectors

if *m*>*n*

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma_n \\ 0 & \dots & 0 \\ \vdots & & \vdots \\ 0 & \dots & 0 \end{bmatrix}$$

if *n*>*m*

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1 & 0 & 0 & 0 & \dots & 0 \\ 0 & \ddots & 0 & \vdots & & \vdots \\ 0 & 0 & \sigma_m & 0 & \dots & 0 \end{bmatrix}$$

Matrix Decompositions

Matrix decompositions can enable faster/facile computations and are thus central to many numerical implementations underlying ML algorithms

Singular value decomposition
(***SVD***, the never-fail decomposition)**:**

$$\underset{m \times n}{\boldsymbol{A}} = \underset{m \times m}{\boldsymbol{U}} \; \underset{m \times n}{\boldsymbol{\Sigma}} \; \underset{n \times n}{\boldsymbol{V}^\top}$$

$\boldsymbol{U}$: orthogonal matrix with vectors $\boldsymbol{u}_i$

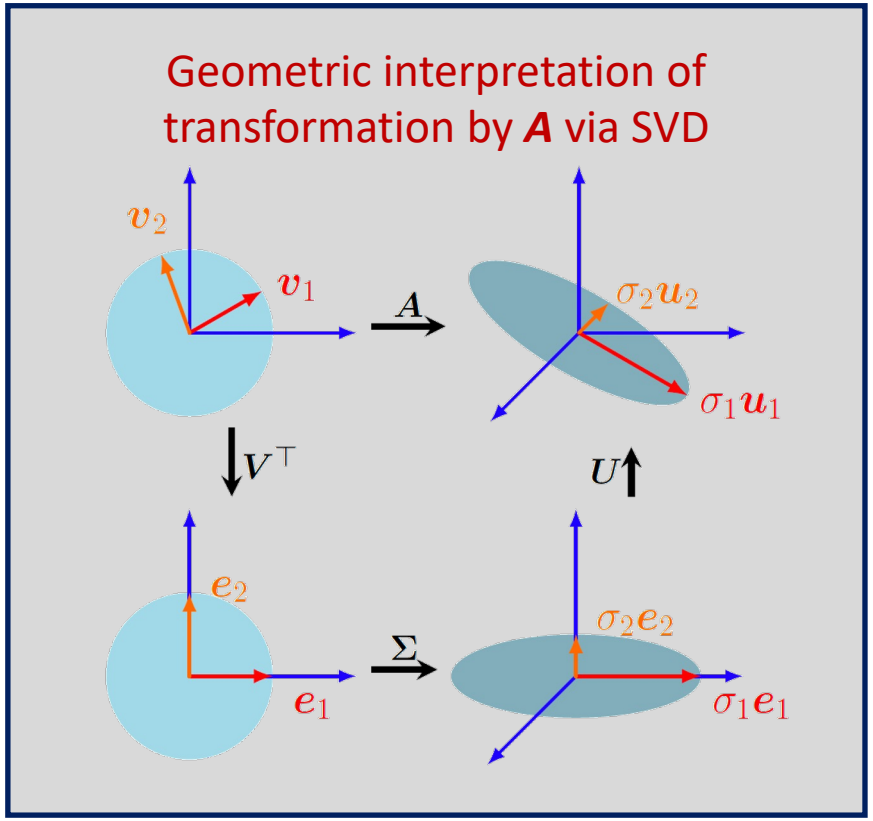
left-singular vectors

$\boldsymbol{\Sigma}$: diagonal matrix with entries $\Sigma_{ii} = \boldsymbol{\sigma}_i \geq 0$

singular values

$\boldsymbol{V}^\top$: orthogonal matrix with vectors $\boldsymbol{v}_i$

right-singular vectors



Matrix Decompositions

Matrix decompositions can enable faster/facile computations and are thus central to many numerical implementations underlying ML algorithms

CLASS EXERCISE

Review of Essential Mathematics for Machine Learning (Pt. 2)

Analytic Geometry

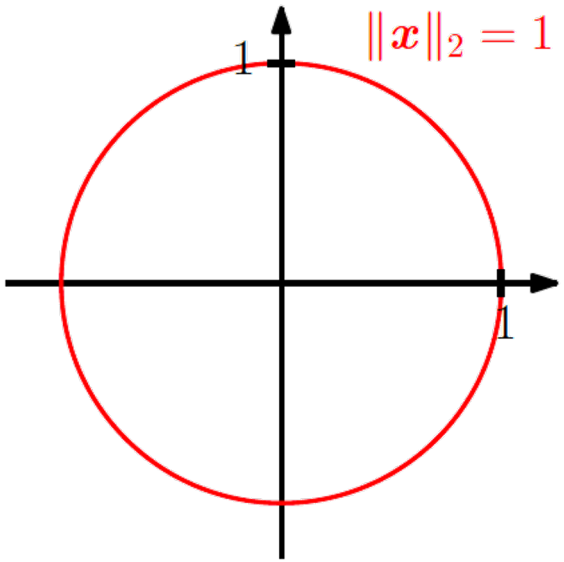
Image sources: **MML:** *Mathematics for Machine Learning* by Deisenroth, Faisal, and Ong @ mml-book.com

Norms, Inner Products, and Distances

We will find it useful to define "lengths" between vectors to measure similarity → ***Norm***

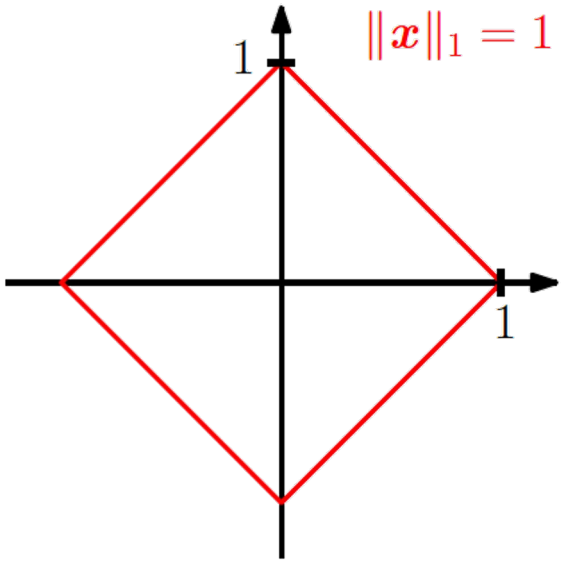
Euclidean norm (or l_2-norm)

$$\|\boldsymbol{x}\|_2 = \sum_i \sqrt{x_i^2} = \sqrt{\boldsymbol{x}^T \boldsymbol{x}}$$



Manhattan norm (or l_1-norm)

$$\|\boldsymbol{x}\|_1 = \sum_i |x_i|$$



Norms, Inner Products, and Distances

We may also like to characterize the relative orientation of two vectors → ***inner product***

fancy definition: "*positive definite, symmetric bilinear mapping*"

$$\Omega : V \times V \to \mathbb{R}$$

If $\Omega(\lambda \boldsymbol{x} + \psi \boldsymbol{y}, \boldsymbol{z}) = \lambda \Omega(\boldsymbol{x}, \boldsymbol{z}) + \psi \Omega(\boldsymbol{y}, \boldsymbol{z})$

$\Omega(\boldsymbol{x}, \lambda \boldsymbol{y} + \psi \boldsymbol{z}) = \lambda \Omega(\boldsymbol{x}, \boldsymbol{y}) + \psi \Omega(\boldsymbol{x}, \boldsymbol{z})$ (bilinear mapping)

and $\Omega(\boldsymbol{x}, \boldsymbol{y}) = \Omega(\boldsymbol{y}, \boldsymbol{x})$ (symmetric)

and $\forall \boldsymbol{x} \in V \backslash \{\boldsymbol{0}\} : \Omega(\boldsymbol{x}, \boldsymbol{x}) > 0\,, \quad \Omega(\boldsymbol{0}, \boldsymbol{0}) = 0$ (positive definite)

Then $\langle \boldsymbol{x}, \boldsymbol{y} \rangle \equiv \Omega(\boldsymbol{x}, \boldsymbol{y})$ is an inner product

Norms, Inner Products, and Distances

An inner product that you know:

$$\boldsymbol{x}^T \boldsymbol{y} = \sum_i x_i y_i$$

An example of one you probably don't:

$$\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^2$$

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle := x_1 y_1 - (x_1 y_2 + x_2 y_1) + 2 x_2 y_2$$

Inner products as matrices:

Let $\hat{\boldsymbol{x}}$ $\hat{\boldsymbol{y}}$ be coordinates of $\boldsymbol{x}$ $\boldsymbol{y}$ in the basis $\mathcal{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$

If $\boldsymbol{A}$, s.t. $A_{ij} := \langle \boldsymbol{b}_i, \boldsymbol{b}_j \rangle$ is symmetric, positive definite

Then $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \hat{\boldsymbol{x}}^\top \boldsymbol{A} \hat{\boldsymbol{y}}$

Norms, Inner Products, and Distances

Any ***inner product*** also defines a ***norm***:

$$\|\boldsymbol{x}\| := \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle}$$

The ***distance (metric)*** between two vectors is

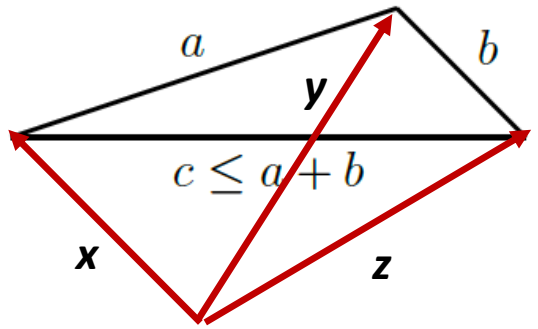
$$d(\boldsymbol{x}, \boldsymbol{y}) := \|\boldsymbol{x} - \boldsymbol{y}\| = \sqrt{\langle \boldsymbol{x} - \boldsymbol{y}, \boldsymbol{x} - \boldsymbol{y} \rangle}$$

Cauchy-Schwarz Inequality

$$|\langle \boldsymbol{x}, \boldsymbol{y} \rangle| \leqslant \|\boldsymbol{x}\| \|\boldsymbol{y}\|$$

Triangle Inequality

$$d(\boldsymbol{x}, \boldsymbol{z}) \leqslant d(\boldsymbol{x}, \boldsymbol{y}) + d(\boldsymbol{y}, \boldsymbol{z})$$



Norms, Inner Products, and Distances

Exercise:

Compute the distance metric between the two vectors

$$\boldsymbol{v}_1 = [3, 2]^T$$

$$\boldsymbol{v}_2 = [2, 1]^T$$

using ***dot product*** *vs.*

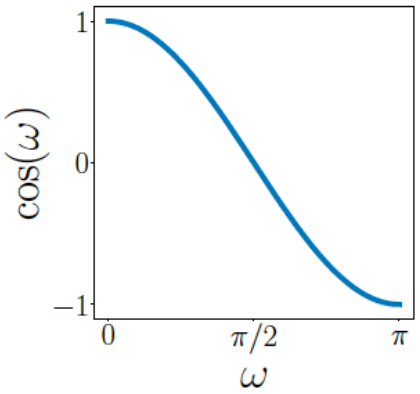
$$\boldsymbol{A} = \begin{bmatrix} 1 & -\frac{1}{2} \\ -\frac{1}{2} & 1 \end{bmatrix}$$

Norms, Inner Products, and Distances

Inner products also enable definition of an angle between vectors

Following Cauchy-Schwarz inequality,

$$-1 \leqslant \frac{\langle \boldsymbol{x}, \boldsymbol{y} \rangle}{\|\boldsymbol{x}\| \, \|\boldsymbol{y}\|} \leqslant 1$$



there is a unique mapping to an angle ω on the interval [0,π]

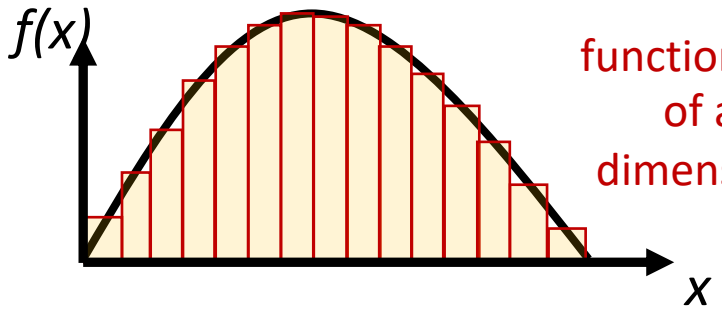
$$\cos\omega = \frac{\langle \boldsymbol{x}, \boldsymbol{y} \rangle}{\|\boldsymbol{x}\| \, \|\boldsymbol{y}\|}$$

orthogonality

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = 0$$

We can also define inner products over functions

$$\langle u, v \rangle := \int_a^b u(x)v(x)dx$$



functions can be analogously thought of as vectors with an infinite-dimensional basis of delta functions

Norms, Inner Products, and Distances

Orthogonal Bases/Complements

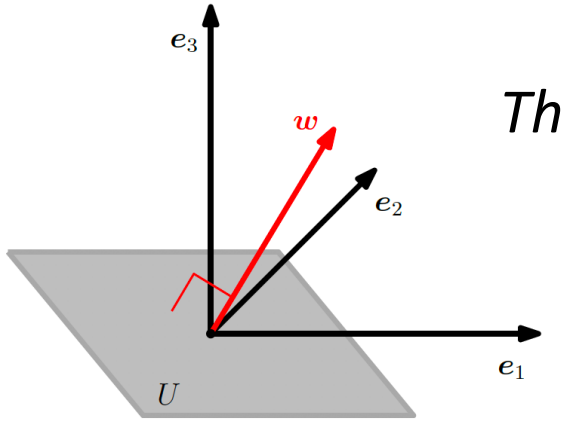
$\mathcal{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$ is orthonormal if $\langle \boldsymbol{b}_i, \boldsymbol{b}_j \rangle = \delta_{ij} \Leftrightarrow \boldsymbol{B}^T \boldsymbol{B} = \boldsymbol{I}_n$

Suppose $\tilde{\mathcal{B}} = (\tilde{\boldsymbol{b}}_1, \ldots, \tilde{\boldsymbol{b}}_n)$ is not necessarily orthogonal and/or not normalized

Gram-Schmidt Process enables orthogonalization by Gaussian elimination on $\left[\tilde{\mathbf{B}}\tilde{\mathbf{B}}^T | \tilde{\mathbf{B}}\right]$

(examine QR decomposition, np.linalg.qr)

For an *N*-dimensional vector space *V* with an *M*-dimensional subspace *U,* the orthogonal complement of $U^\perp$ is a (*N*-*M*)-dimensional subspace of *V* containing all vectors in *V* that are orthogonal to every vector in *U* ($U^\perp$ describes a hyperplane)



This enables a unique decomposition:

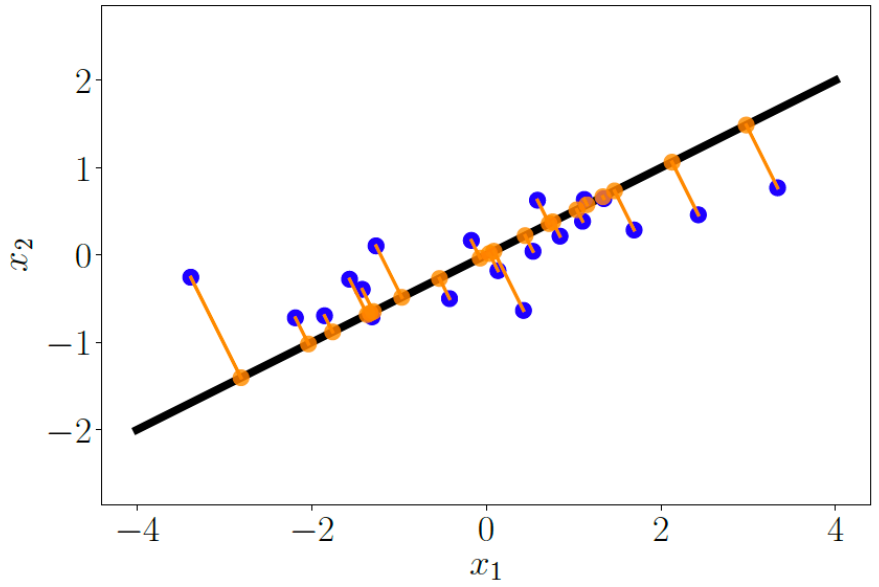
$$\boldsymbol{v} = \sum_{i=1}^{M} \lambda_i \boldsymbol{b}_i + \sum_{j=1}^{N-M} \psi_j \boldsymbol{b}_j^\perp$$

Projections

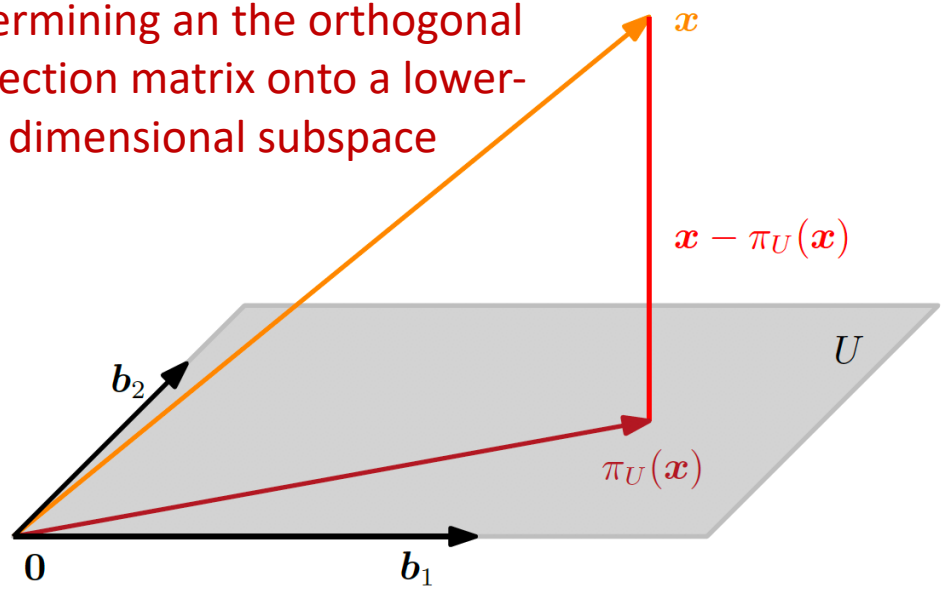
A ***projection*** is a linear mapping between vector spaces (V and U, a subspace of V) with the property that

$$\boldsymbol{P}_\pi^2 = \boldsymbol{P}_\pi \quad \leftarrow \text{projection matrix}$$

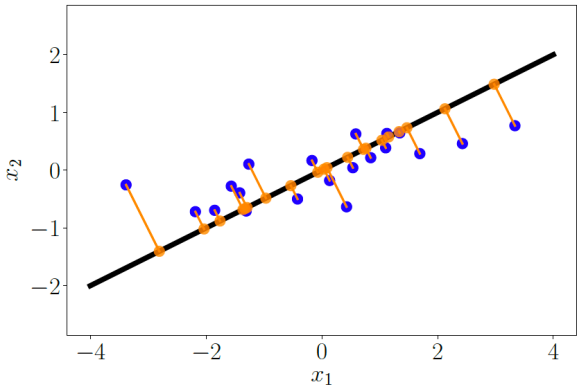
projection of a 2d-dataset onto a 1d-subspace

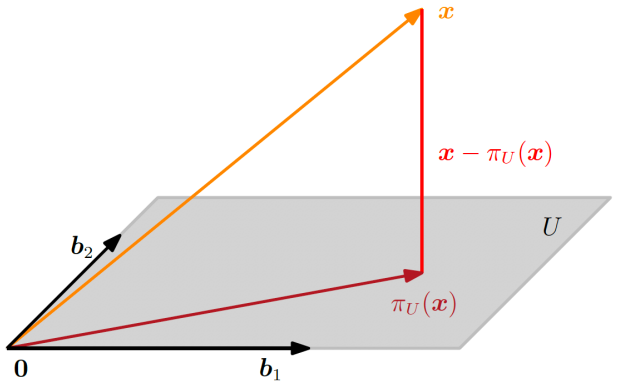


An oft-encountered task may be determining an the orthogonal projection matrix onto a lower-dimensional subspace



Projections





We want $\pi_U(\boldsymbol{x}) = \sum_{i=1}^{M} \lambda_i \boldsymbol{b}_i = \boldsymbol{B}\boldsymbol{\lambda}$ given $\mathcal{B}_U = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_M)$

such that $\pi_U(\boldsymbol{x}) - \boldsymbol{x}$ is ***orthogonal*** to *U* and its ***distance*** in ***minimized***

Assuming the dot product as the inner product…

$$\implies \boldsymbol{b}_i^T(\boldsymbol{x} - \boldsymbol{B}\boldsymbol{\lambda}) = 0, \; i = 1, \ldots, M$$

$$\Updownarrow$$

$$\boldsymbol{B}^T(\boldsymbol{x} - \boldsymbol{B}\boldsymbol{\lambda}) = \boldsymbol{0} \iff \boldsymbol{B}^T\boldsymbol{B}\boldsymbol{\lambda} = \boldsymbol{B}^T\boldsymbol{x}$$

normal equation

$$\boldsymbol{\lambda} = (\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{B}^T\boldsymbol{x}$$

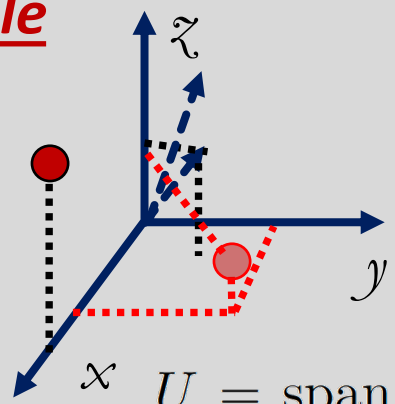
if ***B*** describes an orthonormal basis??

$$\implies \boldsymbol{P}_\pi = \boldsymbol{B}(\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{B}^T$$

Projections

Example

$$x = \begin{bmatrix} 6 \\ 0 \\ 0 \end{bmatrix}$$



$$U = \text{span}[\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix}] \subseteq \mathbb{R}^3$$

```
>>> x = np.array([[6,0,0]]).transpose()
>>> P@x
array([[ 5.],
       [ 2.],
       [-1.]])
```

Projection Error

$$\|x - \pi_U(x)\| = \left\| \begin{bmatrix} 1 & -2 & 1 \end{bmatrix}^\top \right\| = \sqrt{6}$$

```
>>> dx = x - P@x
>>> print(dx.transpose()@B)
[[-2.22044605e-16 -4.44089210e-16]]
```

$$B = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{bmatrix} \qquad P_\pi = B(B^T B)^{-1} B^T$$

```
>>> b1 = np.array([[1,1,1]]).transpose()
>>> b2 = np.array([[0,1,2]]).transpose()
>>> B  = np.hstack([b1,b2])
>>> print(B)
[[1 0]
 [1 1]
 [1 2]]
>>> BT = B.transpose()
>>> print(BT)
[[1 1 1]
 [0 1 2]]
>>> P = B@np.linalg.inv(BT@B)@BT
>>> print(P)
[[ 0.83333333  0.33333333 -0.16666667]
 [ 0.33333333  0.33333333  0.33333333]
 [-0.16666667  0.33333333  0.83333333]]
```

```
>>> print(P@P)
[[ 0.83333333  0.33333333 -0.16666667]
 [ 0.33333333  0.33333333  0.33333333]
 [-0.16666667  0.33333333  0.83333333]]
```

Other remarks on Projections

Recall that for any given system of linear equations, $Ax = b$ we may have

no solution **one** unique solution **infinite**ly many solutions

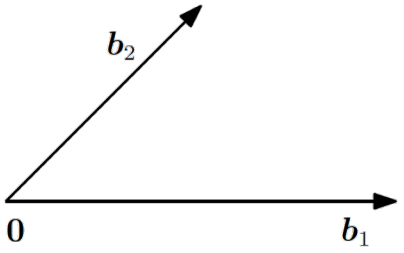
It is rare that any realistic situation is exactly solvable

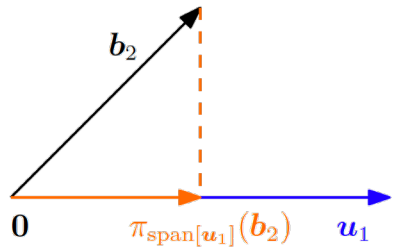
Interpretation: ***b*** does not lie in the span of ***A***

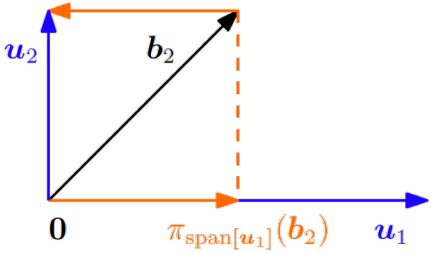
Solution: find the closest projection of ***b*** onto the subspace spanned by ***A*** ***"Least-squares"***

Gram-Schmidt orthogonalization constructs an orthogonal basis ($u_1,\ldots,u_n$) from any basis ($b_1,\ldots,b_n$) by iteratively determining projections on incrementally larger subspaces

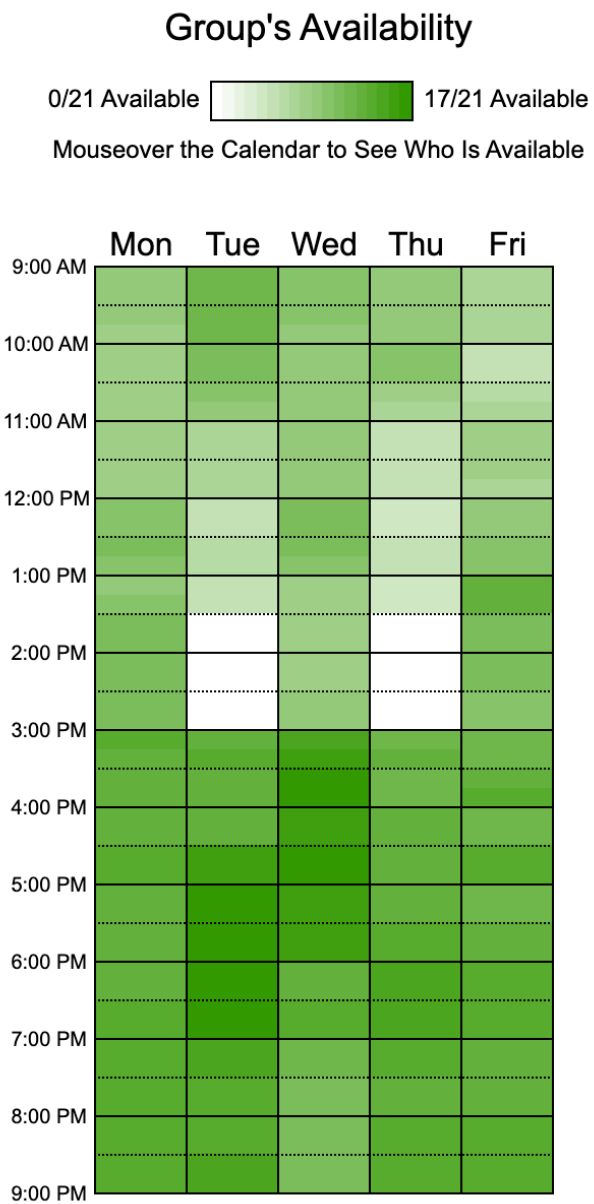
$$u_1 := b_1$$
$$u_k := b_k - \pi_{\mathrm{span}[u_1,\ldots,u_{k-1}]}(b_k)\,, \quad k = 2,\ldots,n$$







Office Hours



- **Slot 1:** Tuesday, 5:00-6:30pm

Proposed Slot 2

- Wednesday, 5:00-6:00pm
- Friday, 4:00-5:00pm
- Friday, 5:00-6:00pm

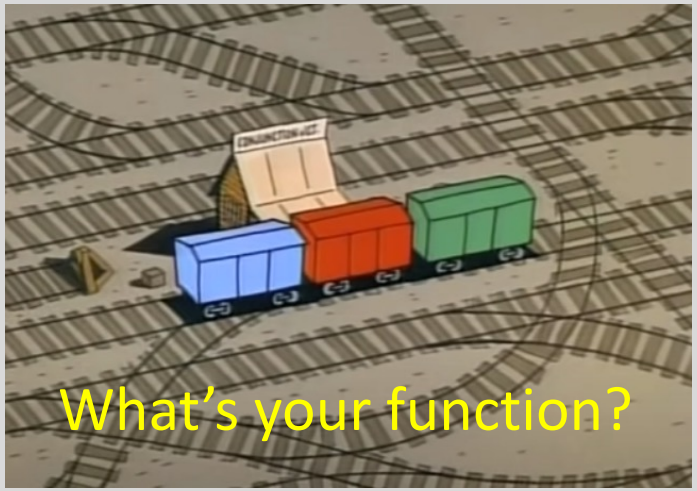
Review of Essential Mathematics for Machine Learning (Pt. 3)

Vector Calculus & Probability

Image sources: **MML:** *Mathematics for Machine Learning* by Deisenroth, Faisal, and Ong @ mml-book.com

Optimization and Gradients in Machine Learning

Many ML algorithms depend on optimization algorithms; such algorithms typically exploit gradient information with respect to some prescribed objective function



$$f : \mathbb{R}^n \to \mathbb{R}, \boldsymbol{x} \mapsto f(\boldsymbol{x})$$

a ***function*** *f* is a mapping that explicitly assigns every input *x* exactly one value ***f(x)***

Partial derivatives are defined

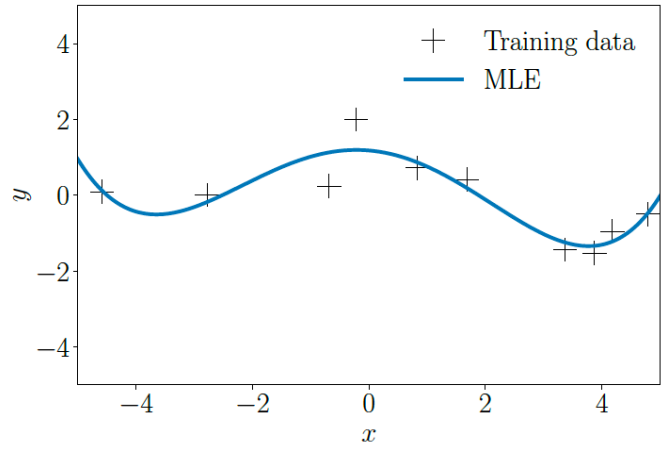
$$\frac{\partial f}{\partial x_1} = \lim_{h \to 0} \frac{f(x_1 + h, x_2, \ldots, x_n) - f(\boldsymbol{x})}{h}$$

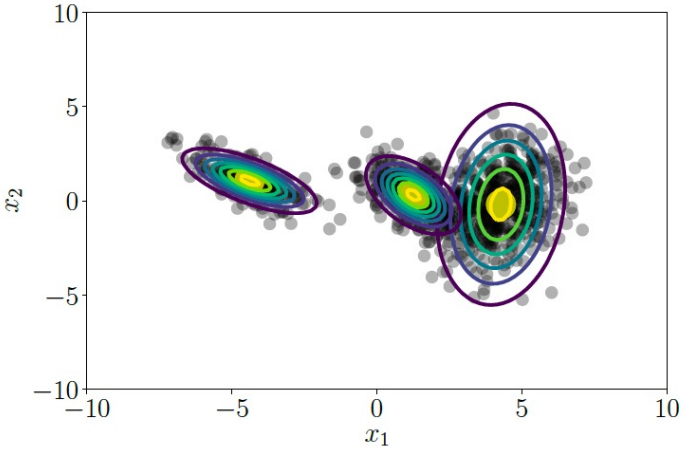
$$\vdots$$

$$\frac{\partial f}{\partial x_n} = \lim_{h \to 0} \frac{f(x_1, \ldots, x_{n-1}, x_n + h) - f(\boldsymbol{x})}{h}$$

with a **gradient** expressed as

$$\nabla_{\boldsymbol{x}} f = \operatorname{grad} f = \frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{x}} = \left[\frac{\partial f(\boldsymbol{x})}{\partial x_1} \quad \frac{\partial f(\boldsymbol{x})}{\partial x_2} \quad \cdots \quad \frac{\partial f(\boldsymbol{x})}{\partial x_n} \right] \in \mathbb{R}^{1 \times n}$$





Partial Differentiation of functions

Basic differentiation rules still apply with some care

product rule

$$\frac{\partial}{\partial \boldsymbol{x}}\big(f(\boldsymbol{x})g(\boldsymbol{x})\big) = \frac{\partial f}{\partial \boldsymbol{x}}g(\boldsymbol{x}) + f(\boldsymbol{x})\frac{\partial g}{\partial \boldsymbol{x}}$$

sum rule

$$\frac{\partial}{\partial \boldsymbol{x}}\big(f(\boldsymbol{x}) + g(\boldsymbol{x})\big) = \frac{\partial f}{\partial \boldsymbol{x}} + \frac{\partial g}{\partial \boldsymbol{x}}$$

chain rule

$$\frac{\partial}{\partial \boldsymbol{x}}\big(g(f(\boldsymbol{x}))\big) = \frac{\partial g}{\partial f}\frac{\partial f}{\partial \boldsymbol{x}}$$

Application of chain rule

Suppose $f(\boldsymbol{x}), \boldsymbol{x}(t)$

$$\frac{df(\boldsymbol{x}(t))}{dt} = \frac{\partial f}{\partial \boldsymbol{x}}\frac{\partial \boldsymbol{x}}{\partial t} = \begin{bmatrix}\frac{\partial f}{\partial x_1} & \cdots & \frac{\partial f}{\partial x_n}\end{bmatrix}\begin{bmatrix}\frac{\partial x_1}{\partial t} \\ \vdots \\ \frac{\partial x_n}{\partial t}\end{bmatrix}$$

Extension of chain rule for gradient

Suppose now $f(\boldsymbol{x}), \boldsymbol{x}(s,t)$

$$\frac{df}{d(s,t)} = \frac{\partial f}{\partial \boldsymbol{x}}\frac{\partial \boldsymbol{x}}{\partial (s,t)} = \underbrace{\begin{bmatrix}\frac{\partial f}{\partial x_1} & \cdots & \frac{\partial f}{\partial x_n}\end{bmatrix}}_{1\text{x}n}\underbrace{\begin{bmatrix}\frac{\partial x_1}{\partial s} & \frac{\partial x_1}{\partial t} \\ \vdots & \vdots \\ \frac{\partial x_n}{\partial t} & \frac{\partial x_n}{\partial t}\end{bmatrix}}_{n\text{x}2}$$

Note that we could compute via matrix multiplication

Gradients of Vector-valued functions (vector fields)

We will also encounter vector-valued functions $\boldsymbol{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} f_1(\boldsymbol{x}) \\ \vdots \\ f_m(\boldsymbol{x}) \end{bmatrix}$$

simply a vector of functions

Rules for operating on each of the functions are precisely the same as just shown

The gradient of a vector field yields a matrix

$$\frac{\mathrm{d}\boldsymbol{f}(\boldsymbol{x})}{\mathrm{d}\boldsymbol{x}} = \begin{bmatrix} \boxed{\frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial x_1}} \cdots \boxed{\frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial x_n}} \end{bmatrix}$$

$$= \begin{bmatrix} \boxed{\begin{matrix} \frac{\partial f_1(\boldsymbol{x})}{\partial x_1} \\ \vdots \\ \frac{\partial f_m(\boldsymbol{x})}{\partial x_1} \end{matrix}} \cdots \boxed{\begin{matrix} \frac{\partial f_1(\boldsymbol{x})}{\partial x_n} \\ \vdots \\ \frac{\partial f_m(\boldsymbol{x})}{\partial x_n} \end{matrix}} \end{bmatrix}$$

This *m* x *n* matrix of first-order partial derivatives is often referred to as the ***Jacobian*, *J***

$$J_{ij} = \frac{\partial f_i}{\partial x_j}$$

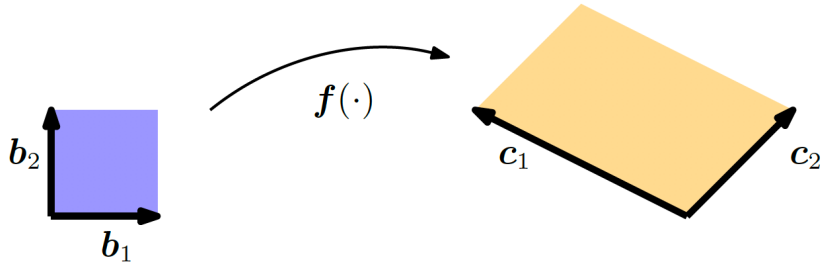
*Note: the **Jacobian** can represent a coordinate transformation (specifically from a coordinate representation in **x** to a coordinate representation in **f**)*

Gradients of Vector-valued functions (vector fields)

Example

$$\boldsymbol{b}_1 = [1,0]^T ; \boldsymbol{b}_2 = [0,1]^T$$

$$\boldsymbol{c}_1 = [-2,1]^T ; \boldsymbol{c}_2 = [1,1]^T$$



*What's the transformation matrix for changing basis from B to C (**x** to **f**)?*

$$\boldsymbol{T}_f = \begin{bmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{122} \end{bmatrix}$$

T has columns of the coordinate mapping from one basis to the other

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

$$f_1(\boldsymbol{x}) = \alpha_{11}x_1 + \alpha_{12}x_2$$

$$f_2(\boldsymbol{x}) = \alpha_{21}x_1 + \alpha_{22}x_2$$

$$\boldsymbol{J} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix}$$

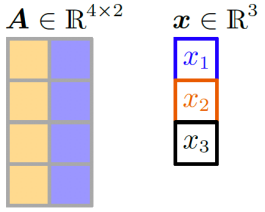
So, the Jacobian is identical to the linear transformation matrix in this case; advantage of Jacobian is for non-linear transformations!

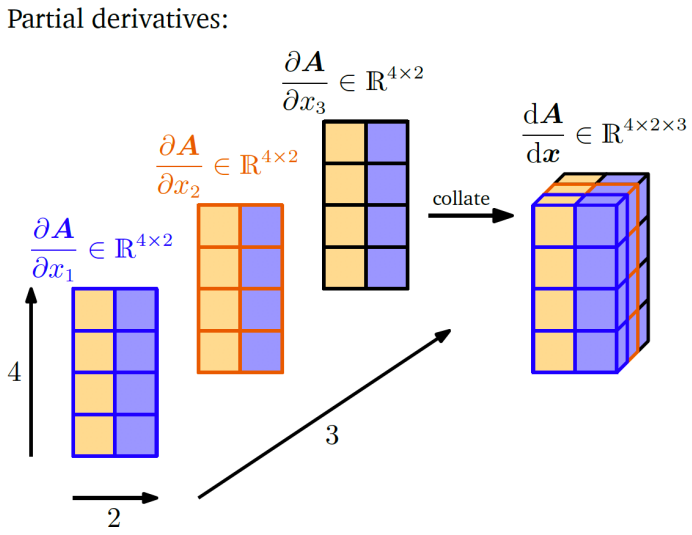
Gradients of Matrices

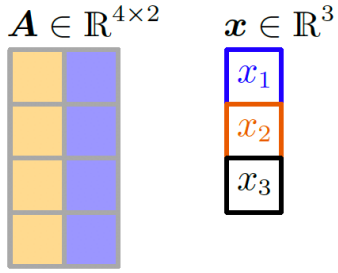
Gradients of matrices with respect to vectors (or other matrices) → **tensors**

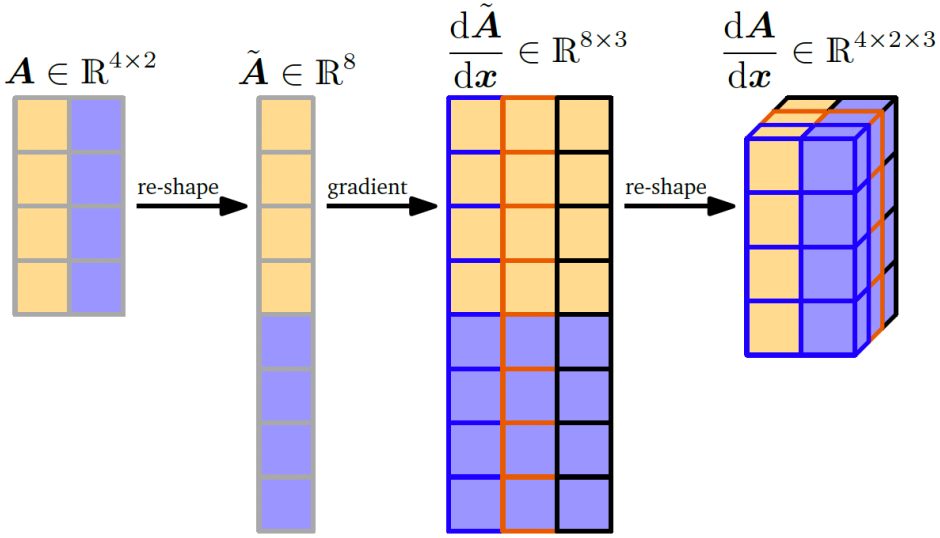
$$\boldsymbol{A} \in \mathbb{R}^{m \times n}; \boldsymbol{B} \in \mathbb{R}^{p \times q} \implies \boldsymbol{J} \in \mathbb{R}^{(m \times n) \times (p \times q)} \quad J_{ijkl} = \frac{\partial A_{ij}}{\partial B_{kl}}$$

Two ways to think about (and compute) it:









Higher-order Derivatives and the Hessian

2nd derivatives of *f*

$$\frac{\partial^2 f}{\partial x^2} = \frac{\partial}{\partial x}\left(\frac{\partial f}{\partial x}\right)$$

$$\frac{\partial^2 f}{\partial x \partial y} = \frac{\partial}{\partial x}\left(\frac{\partial f}{\partial y}\right)$$

$$\frac{\partial^2 f}{\partial y \partial x} = \frac{\partial}{\partial y}\left(\frac{\partial f}{\partial x}\right)$$

$$\frac{\partial^2 f}{\partial y \partial x} = \frac{\partial^2 f}{\partial x \partial y}$$

if *f(x,y) is continuous and twice differentiable*

nth derivative of *f* with respect to *x*

$$\frac{\partial^n f}{\partial x^n} = \frac{\partial}{\partial x}\left(\frac{\partial^{n-1} f}{\partial x^{n-1}}\right) \quad D^n_{\boldsymbol{x}} f$$

The ***Hessian*** is the second derivative analogue of the Jacobian (it contains all the second derivatives)

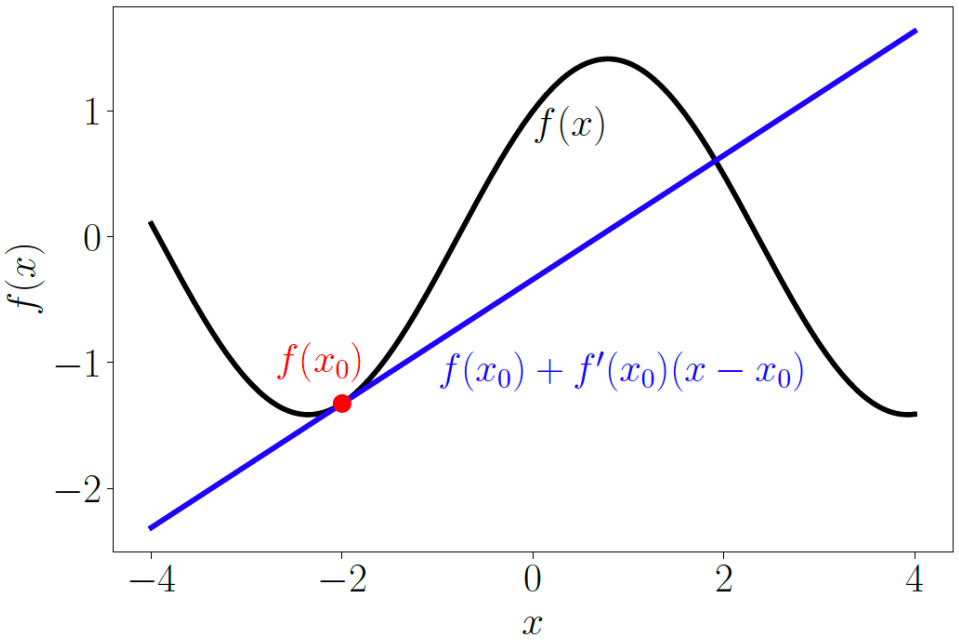
$$\boldsymbol{H} = \nabla^2_{\boldsymbol{x}} f(\boldsymbol{x}) \quad H_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}$$

Here, ***H*** is an *n*x*n*, matrix

Question: What would be the corresponding form for the Hessian if applied to a vector field?

$$\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^m$$

Linearization and Taylor Expansions



Gradients enable local approximations to functions

Linear: $f(\boldsymbol{x}) \approx f(\boldsymbol{x}_0) + \dfrac{df(\boldsymbol{x}_0)}{d\boldsymbol{x}}(\boldsymbol{x} - \boldsymbol{x}_0)$

General: $f(\boldsymbol{x}) = \sum_{k=0}^{\infty} \dfrac{D_{\boldsymbol{x}}^k f(\boldsymbol{x}_0)}{k!}(\boldsymbol{x} - \boldsymbol{x}_0)^k$

kth-order tensor

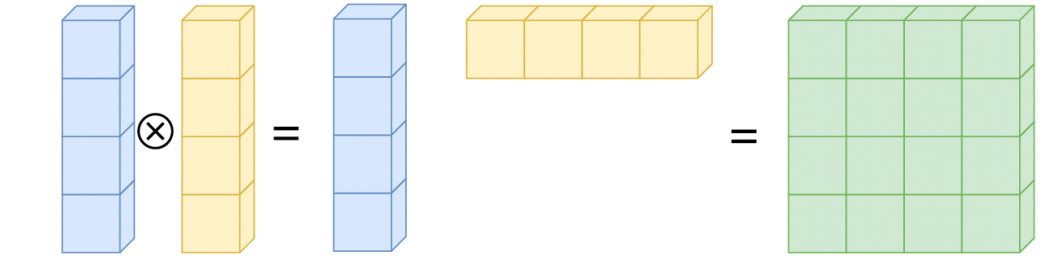
easiest to understand visually (at least up to 3D):

$$\boldsymbol{\delta} \equiv \boldsymbol{x} - \boldsymbol{x}_0$$

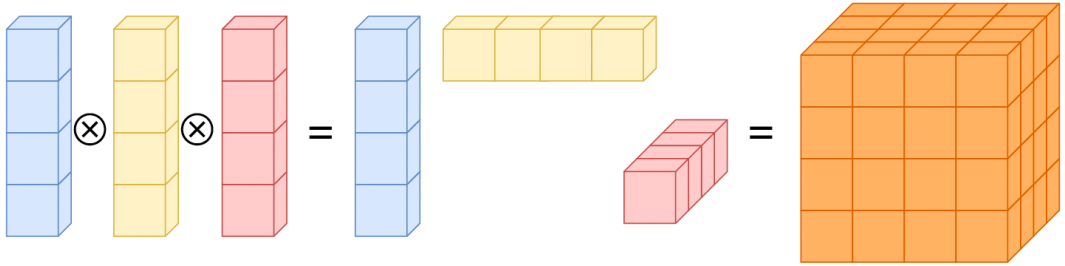
$$\boldsymbol{\delta}^2 = \boldsymbol{\delta} \otimes \boldsymbol{\delta}$$

$$\boldsymbol{\delta}^3 = \boldsymbol{\delta} \otimes \boldsymbol{\delta} \otimes \boldsymbol{\delta}$$

$\otimes$ is the ***outer product***



$$\boldsymbol{\delta}^2 = \boldsymbol{\delta} \otimes \boldsymbol{\delta} = \boldsymbol{\delta}\boldsymbol{\delta}^T$$



$$\boldsymbol{\delta}^3 = \boldsymbol{\delta} \otimes \boldsymbol{\delta} \otimes \boldsymbol{\delta}$$

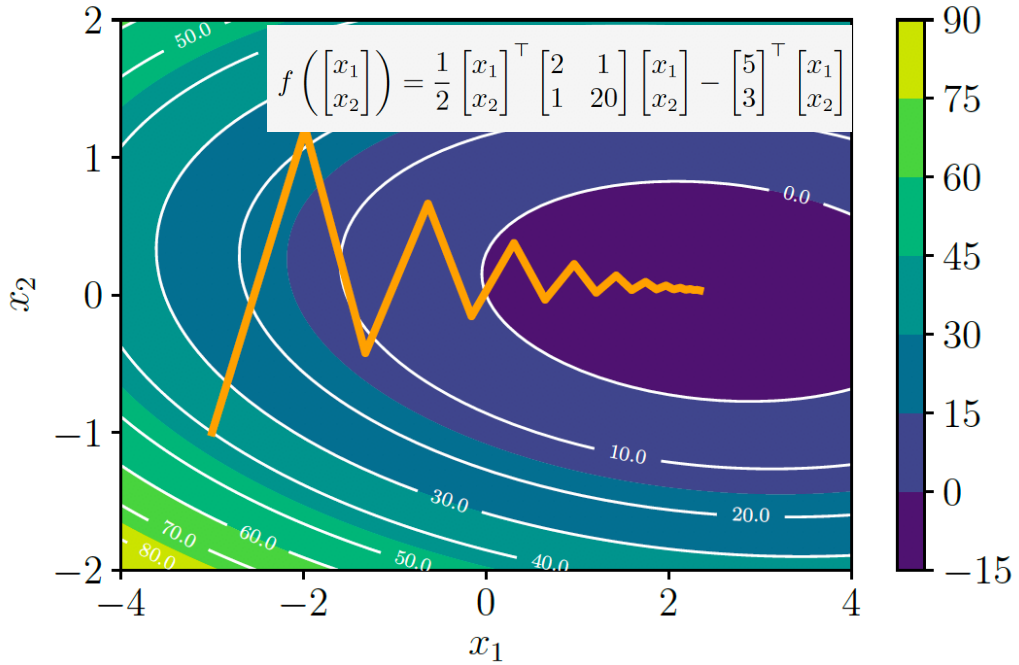
$$\boldsymbol{\delta}^3_{ijk} = \delta_i \delta_j \delta_k$$

Optimization

Essential task: $f : \mathbb{R}^n \to \mathbb{R}, \boldsymbol{x} \mapsto f(\boldsymbol{x})$ $\min_{\boldsymbol{x}} f(\boldsymbol{x})$

Gradient Descent

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i - \gamma_i \left[\nabla f(\boldsymbol{x}_i)\right]^T$$



- *very simple method*
- *slowly convergent to closest minima*
- *depends on (adaptive) stepsize*

with momentum

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i - \gamma_i \left[\nabla f(\boldsymbol{x}_i)\right]^T + \alpha \Delta \boldsymbol{x}_i$$

$$\Delta \boldsymbol{x}_i = \alpha \Delta \boldsymbol{x}_{i-1} - \gamma_{i-1} \left[\nabla f(\boldsymbol{x}_{i-1})\right]^T$$

- *uses "memory" to reduce jitter*

stochastic

$$f(\boldsymbol{x}) = \sum_{k=1}^{N} f_k(\boldsymbol{x})$$

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i - \gamma_i \sum_{k \subset \mathbb{N}: k \leq N} \left[\nabla f_k(\boldsymbol{x}_i)\right]^T$$

- *useful for large N, which may not be atypical in machine learning applications*

Constrained Optimization

Task: $f : \mathbb{R}^n \to \mathbb{R}, \boldsymbol{x} \mapsto f(\boldsymbol{x}) \quad \min_{\boldsymbol{x}} f(\boldsymbol{x}) \quad g_k(\boldsymbol{x}) \leq 0 \text{ for } k = 1, \ldots, m$

Naïve approach: Add some penalty function that induces large function value when constraint(s) not satisfied

Lagrange multipliers: $\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \boldsymbol{\lambda}^T \boldsymbol{g}(\boldsymbol{x})$

Duality: exchange optimization in one set of variables (primal variables, **x**) for a different optimization problem in another set of variables (dual variables, **λ**)

→ $\max_{\boldsymbol{\lambda} \geq \mathbf{0}} \underbrace{\min_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda})}$

unconstrained optimization for fixed ***λ***

concave

Probability: Basic Definitions and Rules

Probability equates to a quantified notion of uncertainty → some element of randomness

A ***random variable*** *X* is a *function* that maps outcomes of random experiments to a result

These constitute a so-called ***probability space***

- Ω ***sample (state) space***: set of all possible outcomes from an "experiment"
- $\mathcal{A}$ ***event space***: the space of potential results from an experiment
- P ***probability***: for an event *A* in the event space, *P(A) is* a numerical measure of the belief/frequency that *A* will occur

T Typically, we are not so interested in the probability space but rather ***a target space*** where outcomes are mapped to quantity of interest via the random variable

e.g.,	sample space	target space
	ABBAABABABABA	*the number of B monomers*

Probability: Basic Definitions and Rules

Probability equates to a quantified notion of uncertainty → some element of randomness

The complement of A being anything but A

$$P(A^c) = 1 - P(A)$$

Probability of the union of two events

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

A, B are mutually exclusive if

$$P(A \cap B) = 0$$

probability of A occurring given that B has occurred (conditional probability)

If A and B are independent, then

$$P(A \cap B) = P(A)P(B)$$

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

Probability: Basic Definitions and Rules

Contrived example: *grabbing two colored marbles out of a bag with replacement*

Q: How many red marbles do you pull?

$$\Omega \rightarrow \{(\mathrm{R},\mathrm{R}),(\mathrm{Y},\mathrm{Y}),(\mathrm{R},\mathrm{Y}),(\mathrm{Y},\mathrm{R})\}$$

state space

$$X((\mathrm{R},\mathrm{R})) = 2,\ X((\mathrm{R},\mathrm{Y})) = 1,\ X((\mathrm{Y},\mathrm{R})) = 1,\ X((\mathrm{Y},\mathrm{Y})) = 0$$

random variable

$$T \rightarrow \{0, 1, 2\}$$

target space

probability of pulling two reds

$$P(X = 2) = P((\mathrm{R},\mathrm{R})) = P(\mathrm{R} \cap \mathrm{R}) = P(\mathrm{R})P(\mathrm{R}) = 0.7^2$$

probability of pulling one red

$$P(X = 1) = P((\mathrm{R},\mathrm{Y}) \cup (\mathrm{Y},\mathrm{R})) = P((\mathrm{R},\mathrm{Y})) + P((\mathrm{Y},\mathrm{R})) = 2P(R)P(Y) = 2P(R)(1 - P(R)) = 2(0.7)(0.3)$$

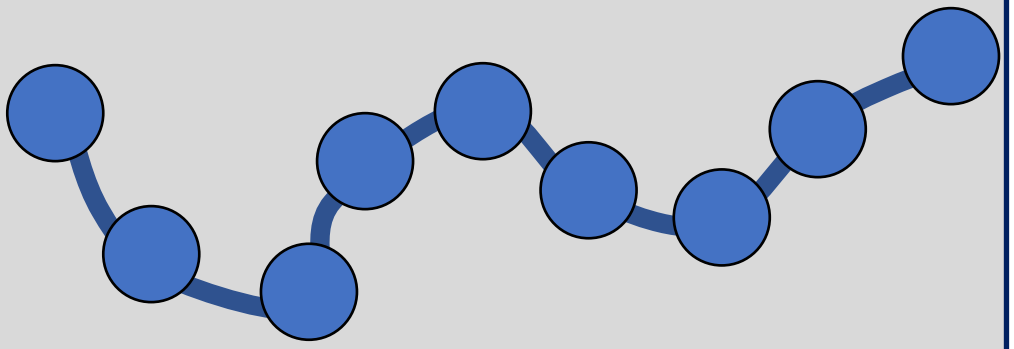
probability of pulling no reds

$$P(X = 0) = P((\mathrm{Y},\mathrm{Y})) = (1 - P(R))(1 - P(R)) = 0.3^2$$

Although we expressed probabilities in terms of the probability space, notice that the random variable X exhibits its own distribution function, which we can denote P_X

Probability: Basic Definitions and Rules

Exercise: Polymer of length N with probability p of bond scission at each bond over interval of time t



Q: How many chain segments there are after time t?

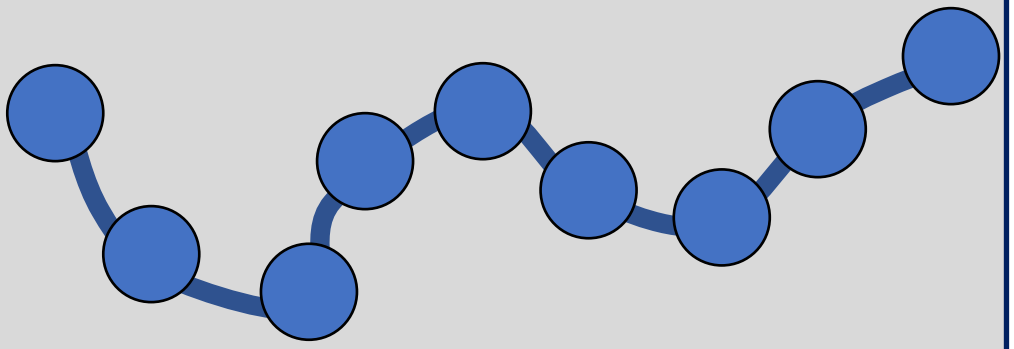
What is the sample/state space?

What is the target space and random variable?

What is the probability of observing 2 segments?

Probability: Basic Definitions and Rules

Exercise: Polymer of length *N* with probability *p* of bond scission at each bond over interval of time *t*



Q: How many chain segments there are after time *t*?

What is the sample/state space?

$$\nu_i = (b_1^{(i)}, \cdots, b_{n-1}^{(i)}) \qquad b_{n-1}^{(i)} = \begin{cases} 1 \text{ if bond exists} \\ 0 \text{ else} \end{cases}$$

$$\Omega = \{\nu_i\}$$

What is the target space and random variable?

$$X(\nu_i) = n - \sum_{j=1}^{n-1} b_j^{(i)} \in [1, n]$$

Here, our target space is finite or countably infinite, which makes X a discrete random variable; X can also be continuous

What is the probability of observing 2 segments?

Need precisely 1 of the n-1 bonds to break, and this can happen for any n-1 of the bonds…

$$(n-1)p(1-p)^{n-2}$$

Discrete and Continuous Probabilities

If target space is ***discrete***, probability is defined for any given random variable:

$$P(X = a) = p_a \quad \textit{probability mass function}$$

$$\sum_X P(X) = 1 \quad \textit{normalization condition}$$

If the target space is ***continuous***, it makes more sense to consider intervals:

$$P(a \leq X \leq b) = \int_a^b f(x)dx \quad \textit{probability density function (PDF)}$$

$$\int_{\mathbb{R}} f(x)dx = 1 \quad \textit{normalization condition}$$

$$F_X(x) = P(X < x) = \int_{-\infty}^{x} f(x')dx' \quad \textit{cumulative distribution function (CDF)}$$

Revisiting Rules

Suppose we have a random variables ***x, y*** with joint probability distribution $p(\boldsymbol{x},\boldsymbol{y})$

Sum rule:

$$p(\boldsymbol{x}) = \sum_{\boldsymbol{y}\in\mathcal{Y}} p(\boldsymbol{x},\boldsymbol{y}) \qquad p(\boldsymbol{x}) = \int_{\mathcal{Y}} p(\boldsymbol{x},\boldsymbol{y})d\boldsymbol{y}$$

marginal distribution

Product rule:

$$p(\boldsymbol{x},\boldsymbol{y}) = p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})$$

Bayes' theorem/rule:

Idea is to make some inference regarding an unobserved random variable based on observed values of another

$$p(\boldsymbol{x}|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})}{p(\boldsymbol{y})}$$

likelihood ***prior*** ***evidence*** ***posterior***

posterior– *what we want (to know something about x having observed y)*

prior – *subjective presumption about the distribution of* ***x*** *(must be nonzero for all plausible* ***x****)*

likelihood– probability of observing **y *were we to know x***

evidence *or marginal likelihood*

$$p(\boldsymbol{y}) := \int p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x}$$

Revisiting Rules

COVID-19 RAPID REPORT Clinical Medicine 2021 Vol 21, No 1: e54–6

The impact of false positive COVID-19 results in an area of low prevalence

Authors: Brendan Healy,[A] Azizah Khan,[B] Huria Metezai,[B] Ian Blyth[C] and Hibo Asad[D]

During the same period, there were 5,079 negative tests; thus a total of 5,110 tests were carried out. The false positive rate (single gene low level positive) in this period was 0.5% (26/5110), giving a specificity of 99.5% (5,079/5,105). Over the same period, the true

PLOS ONE

Real-life clinical sensitivity of SARS-CoV-2 RT-PCR test in symptomatic patients

Elisa Kortela[1☯], Vesa Kirjavainen[2☯], Maarit J. Ahava[2☯], Suvi T. Jokiranta[3☯], Anna But[4], Anna Lindahl[5], Anu E. Jääskeläinen[2], Annemarjut J. Jääskeläinen[2], Asko Järvinen[1], Pia Jokela[2], Hannimari Kallio-Kokko[2], Raisa Loginov[2], Laura Mannonen[2], Eeva Ruotsalainen[1], Tarja Sironen[6,7], Olli Vapalahti[2,6,7], Maija Lappalainen[2], Hanna-Riikka Kreivi[5], Hanna Jarva[2,3], Satu Kurkela[2‡*], Eliisa Kekäläinen[2,3‡]

All 1,194 inpatients (mean [SD] age, 63.2 [18.3] years; 45.2% women) admitted to COVID-19 cohort wards during the study period were included. The outpatient cohort of 1,814 individuals (mean [SD] age, 45.4 [17.2] years; 69.1% women) was sampled from epidemiological line lists by systematic quasi-random sampling. The sensitivity (95% CI) for laboratory confirmed cases (repeat-tested patients) was 85.7% (81.5–89.1%) inpatients; 95.5% (92.2–97.5%) outpatients, 89.9% (88.2–92.1%) all. When also patients that were graded as high

Testing disease specificity/sensitivity over large populations

Suppose that a COVID test is 99.5% specific , such that it reports negative 99.5% of the time someone is not infected and 95.5% sensitive (95.5% of the time it reports positive for someone who is infected)…

If someone tests positive, what's the probability that they have COVID?

The answer: *It depends!*

What is the ***prior*** ? What is the incidence rate if we did not know about the result?

Scenario 1:

0.1% of the population in the area is estimated to have COVID

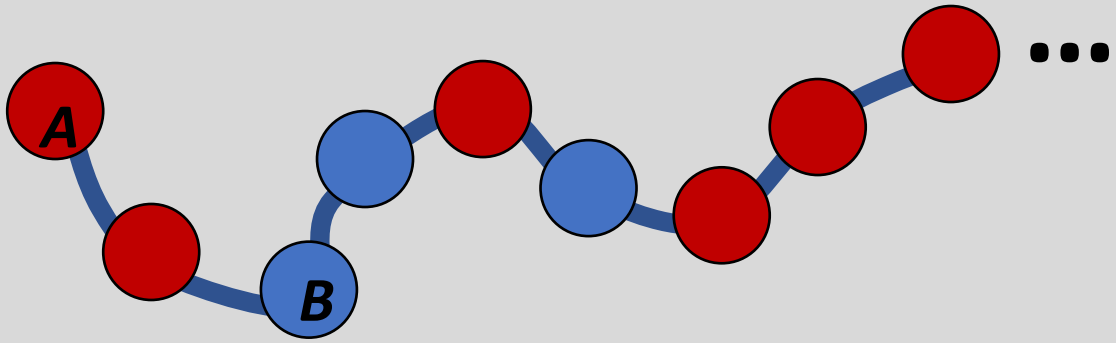
Scenario 2:

1% of the population in the area is estimated to have COVID

Revisiting Rules

Unconventional example

Suppose I have a copolymer comprised of 50% A and 50% B units, and I observe the following segment in the chain. What's the probability that the next unit is type B?



Discuss!

Some possible hypotheses/conjectures….

- *My knowledge of the current sequence doesn't matter → 1/2*
- *I use my observations of the chain to build P(B|A) → 2/5*
- *We use Bayes' Rule with an estimate of P(A|B) from the segment*

$$\rightarrow P(B|A) = \frac{P(A|B)P(B)}{P(A)} = \frac{\frac{2}{3}\frac{1}{2}}{\frac{1}{2}} = \frac{2}{3}$$

(that we have observed more A following B in a population we know to have 50/50 A and B informs our prediction)

Summary Statistics: Definitions

A ***summary statistic*** of a random variable is a deterministic description of how it behaves

For a univariate random variable *X*, ***the expected value*** of a function *g*(*x*) is:

$$\mathbb{E}_X\left[g(x)\right] = \int_{\mathcal{X}} g(x)p(x)dx \text{ or } \sum_{x\in\mathcal{X}} g(x)p(x)$$

(Extension to multivariate random variables requires *X* to be a vector of univariate random variables)

For a random variable *X*, with states ***x*** [*m*]

Mean

$$\boldsymbol{\mu} := \mathbb{E}_X\left[\boldsymbol{x}\right] = \begin{bmatrix} \mathbb{E}_{X_1}\left[x_1\right] \\ \vdots \\ \mathbb{E}_{X_m}\left[x_m\right] \end{bmatrix} \in \mathbb{R}^m$$

For a random variable *X*, with states ***x*** [*m*], and a variably Y, with states ***y*** [*n*]

Covariance

$$\boldsymbol{\Sigma}_{xy} := \operatorname{Cov}\left[\boldsymbol{x},\boldsymbol{y}\right] = \mathbb{E}\left[\boldsymbol{x}\boldsymbol{y}^T\right] - \mathbb{E}\left[\boldsymbol{x}\right]\mathbb{E}\left[\boldsymbol{y}\right]^T \in \mathbb{R}^{m\times n}$$

Summary Statistics: Definitions

A ***summary statistic*** of a random variable is a deterministic description of how it behaves

Variance

$$\begin{aligned}\mathbb{V}[\boldsymbol{x}] &= \mathrm{Cov}_X[\boldsymbol{x}, \boldsymbol{x}] \\ &= \mathbb{E}\left[(\boldsymbol{x}-\mu)(\boldsymbol{x}-\mu)^T\right] \\ &= \mathbb{E}\left[\boldsymbol{x}\boldsymbol{x}^T\right] - \mathbb{E}[\boldsymbol{x}]\,\mathbb{E}[\boldsymbol{x}]^T\end{aligned} = \begin{bmatrix} \mathrm{Cov}[x_1, x_1] & \mathrm{Cov}[x_1, x_2] & \dots & \mathrm{Cov}[x_1, x_m] \\ \mathrm{Cov}[x_2, x_1] & \mathrm{Cov}[x_2, x_2] & \dots & \mathrm{Cov}[x_2, x_m] \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}[x_m, x_1] & \dots & \dots & \mathrm{Cov}[x_m, x_m] \end{bmatrix}$$

standard deviation

$$\boldsymbol{\sigma} := \begin{bmatrix} \sqrt{\mathrm{Cov}[x_1, x_1]} \\ \vdots \\ \sqrt{\mathrm{Cov}[x_m, x_m]} \end{bmatrix}$$

- diagonals → variances of marginal distributions
- off-diagonals are the cross-covariances
- matrix is symmetric, positive semidefinite

Correlation

$$\mathrm{corr}[x, y] = \frac{\mathrm{Cov}[x, y]}{\sqrt{\mathbb{V}[x]\mathbb{V}[y]}} \in [-1, 1]$$

Correlation Matrix

$$\begin{bmatrix} \mathrm{Cov}[\tilde{x}_1, \tilde{x}_1] & \mathrm{Cov}[\tilde{x}_1, \tilde{x}_2] & \dots & \mathrm{Cov}[\tilde{x}_1, \tilde{x}_m] \\ \mathrm{Cov}[\tilde{x}_2, \tilde{x}_1] & \mathrm{Cov}[\tilde{x}_2, \tilde{x}_2] & \dots & \mathrm{Cov}[\tilde{x}_2, \tilde{x}_m] \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}[\tilde{x}_m, \tilde{x}_1] & \dots & \dots & \mathrm{Cov}[\tilde{x}_m, \tilde{x}_m] \end{bmatrix} \quad \tilde{x}_i := \frac{x_i}{\sigma_i}$$

Summary Statistics: Definitions

A ***summary statistic*** of a random variable is a deterministic description of how it behaves

$$\mathbb{E}[\boldsymbol{x}+\boldsymbol{y}] = \mathbb{E}[\boldsymbol{x}] + \mathbb{E}[\boldsymbol{y}]$$
$$\mathbb{E}[\boldsymbol{x}-\boldsymbol{y}] = \mathbb{E}[\boldsymbol{x}] - \mathbb{E}[\boldsymbol{y}]$$
$$\mathbb{V}[\boldsymbol{x}+\boldsymbol{y}] = \mathbb{V}[\boldsymbol{x}] + \mathbb{V}[\boldsymbol{y}] + \mathrm{Cov}[\boldsymbol{x},\boldsymbol{y}] + \mathrm{Cov}[\boldsymbol{y},\boldsymbol{x}]$$
$$\mathbb{V}[\boldsymbol{x}-\boldsymbol{y}] = \mathbb{V}[\boldsymbol{x}] + \mathbb{V}[\boldsymbol{y}] - \mathrm{Cov}[\boldsymbol{x},\boldsymbol{y}] - \mathrm{Cov}[\boldsymbol{y},\boldsymbol{x}]$$

useful relationships

Often summary statistics are estimated empirically from a set of *N* observations

$$\bar{\boldsymbol{x}} := \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}_n \qquad \boldsymbol{\Sigma} := \frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{x}-\bar{\boldsymbol{x}})(\boldsymbol{x}-\bar{\boldsymbol{x}})^T$$

Gaussian Distribution

Gaussian distribution is the most well-studied distribution in science, engineering, and ML

- *ubiquity (often stemming from **central limit theorem**)*
- *computationally convenient*

univariate

$$p(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

multivariate

$$p(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{m/2}|\boldsymbol{\Sigma}|^{-1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$$

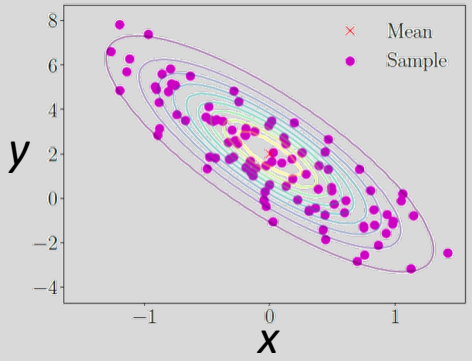
$$p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma})$$ for short

standard normal distribution → ***0*** mean and $\boldsymbol{I}_n$ as the covariance matrix

Basic tenant of Gaussian distributions: *doing things with Gaussians results in Gaussians*

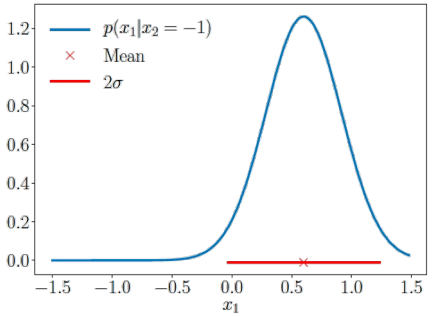
Gaussian Distribution

Consider the joint distribution over *X* and *Y*:



$$p(\boldsymbol{x}, \boldsymbol{y}) = \mathcal{N}\left(\begin{bmatrix}\boldsymbol{\mu}_x \\ \boldsymbol{\mu}_y\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Sigma}_{xx} & \boldsymbol{\Sigma}_{xy} \\ \boldsymbol{\Sigma}_{yx} & \boldsymbol{\Sigma}_{yy}\end{bmatrix}\right)$$

- *Conditionals of Gaussians are Gaussians*



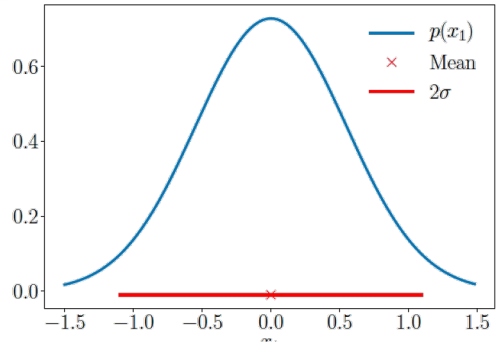
$$p(\boldsymbol{x} \mid \boldsymbol{y}) = \mathcal{N}(\boldsymbol{\mu}_{x \mid y}, \boldsymbol{\Sigma}_{x \mid y})$$

$$\boldsymbol{\mu}_{x \mid y} = \boldsymbol{\mu}_x + \boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1}(\boldsymbol{y} - \boldsymbol{\mu}_y)$$

$$\boldsymbol{\Sigma}_{x \mid y} = \boldsymbol{\Sigma}_{xx} - \boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1}\boldsymbol{\Sigma}_{yx}$$

- *Marginals over Gaussians are Gaussians*

$$p(\boldsymbol{x}) = \int p(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}\boldsymbol{y} = \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_x, \boldsymbol{\Sigma}_{xx})$$



- *The product of Gaussians is Gaussian*

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{a}, \boldsymbol{A})\mathcal{N}(\boldsymbol{x}|\boldsymbol{b}, \boldsymbol{B}) \rightarrow c\mathcal{N}(\boldsymbol{x}|\boldsymbol{c}, \boldsymbol{C})$$

$$\boldsymbol{C} = (\boldsymbol{A}^{-1} + \boldsymbol{B}^{-1})^{-1}$$

$$\boldsymbol{c} = \boldsymbol{C}(\boldsymbol{A}^{-1}\boldsymbol{a} + \boldsymbol{B}^{-1}\boldsymbol{b}) \quad c = \mathcal{N}(\boldsymbol{a}|\boldsymbol{b}, \boldsymbol{A} + \boldsymbol{B})$$

- *The sum over independent Gaussian variables is… Gaussian*

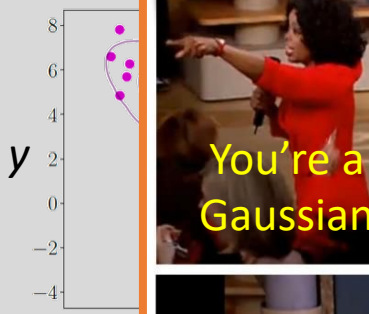
$$p(\boldsymbol{x}, \boldsymbol{y}) = p(\boldsymbol{x})p(\boldsymbol{y})$$

$$p(\boldsymbol{x} + \boldsymbol{y}) = \mathcal{N}(\boldsymbol{\mu}_x + \boldsymbol{\mu}_y, \boldsymbol{\Sigma}_x + \boldsymbol{\Sigma}_y)$$

- *Any linear transformation of Gaussians is Gaussian* $p(a\boldsymbol{x} + b\boldsymbol{y}) = \mathcal{N}(a\boldsymbol{\mu}_x + b\boldsymbol{\mu}_y, a^2\boldsymbol{\Sigma}_x + b^2\boldsymbol{\Sigma}_y)$

Gaussian Distribution

Consider the joint distribution over *X* and *Y*:

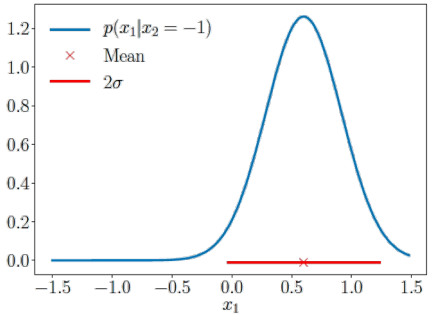


$$\mathcal{N}\left(\begin{bmatrix}\boldsymbol{\mu}_x\\ \boldsymbol{\mu}_y\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Sigma}_{xx} & \boldsymbol{\Sigma}_{xy}\\ \boldsymbol{\Sigma}_{yx} & \boldsymbol{\Sigma}_{yy}\end{bmatrix}\right)$$

- *Ma* *Gaussians*

$$p(\boldsymbol{x} \mid \boldsymbol{\mu}_x, \boldsymbol{\Sigma}_{xx})$$

- *Conditionals of Gaussians are Gaussians*



$$p(\boldsymbol{x} \mid \boldsymbol{y}) = \mathcal{N}(\boldsymbol{\mu}_{x \mid y}, \boldsymbol{\Sigma}_{x \mid y})$$

$$\boldsymbol{\mu}_{x \mid y} = \boldsymbol{\mu}_x + \boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1}(\boldsymbol{y} - \boldsymbol{\mu}_y)$$

$$\boldsymbol{\Sigma}_{x \mid y} = \boldsymbol{\Sigma}_{xx} - \boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1}\boldsymbol{\Sigma}_{yx}$$

- *The product of Gaussians is Gaussian*

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{a}, \boldsymbol{A})\mathcal{N}(\boldsymbol{x}|\boldsymbol{b}, \boldsymbol{B}) \rightarrow c\mathcal{N}(\boldsymbol{x}|\boldsymbol{c}, \boldsymbol{C})$$

$$\boldsymbol{C} = (\boldsymbol{A}^{-1} + \boldsymbol{B}^{-1})^{-1}$$

$$\boldsymbol{c} = \boldsymbol{C}(\boldsymbol{A}^{-1}\boldsymbol{a} + \boldsymbol{B}^{-1}\boldsymbol{b}) \quad c = \mathcal{N}(\boldsymbol{a}|\boldsymbol{b}, \boldsymbol{A} + \boldsymbol{B})$$

- *The sum over independent Gaussian variables is… Gaussian*

$$p(\boldsymbol{x}, \boldsymbol{y}) = p(\boldsymbol{x})p(\boldsymbol{y})$$

$$p(\boldsymbol{x} + \boldsymbol{y}) = \mathcal{N}(\boldsymbol{\mu}_x + \boldsymbol{\mu}_y, \boldsymbol{\Sigma}_x + \boldsymbol{\Sigma}_y)$$

- *Any linear transformation of Gaussians is Gaussian* $p(a\boldsymbol{x} + b\boldsymbol{y}) = \mathcal{N}(a\boldsymbol{\mu}_x + b\boldsymbol{\mu}_y, a^2\boldsymbol{\Sigma}_x + b^2\boldsymbol{\Sigma}_y)$

Examples of Other "Named" Distributions

Most widely used model distributions are members of a so-called ***exponential family***

$$p(\boldsymbol{x} \mid \boldsymbol{\theta}) = h(\boldsymbol{x}) \exp\left(\langle \boldsymbol{\theta}, \boldsymbol{\phi}(\boldsymbol{x}) \rangle - A(\boldsymbol{\theta})\right) \propto \exp(\langle \boldsymbol{\theta}, \phi(\boldsymbol{x}) \rangle)$$

vector of summary statistics — *an inner product (e.g., dot)*

The importance of these (and other) distributions is largely in the realm of probabilistic modeling

Gaussian distribution:

$$\begin{aligned}\mathcal{N}(x|\mu,\sigma^2) &\propto \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right) \\ &\propto \exp\left(-\frac{1}{2\sigma^2}(x^2-2x\mu+\mu^2)\right) \\ &\propto \exp\left(\frac{x\mu}{\sigma^2}-\frac{x^2}{2\sigma^2}\right)\end{aligned}$$

$$\boldsymbol{\theta} = \left[\frac{\mu}{\sigma^2}, -\frac{1}{2\sigma^2}\right]^T$$

$$\phi(\boldsymbol{x}) = \left[x, x^2\right]^T$$

Bernoulli distribution: *distribution for binary outcomes (0,1)*

$$\mathrm{Ber}(\mu) = p(x|\mu) = \mu^x(1-\mu)^{1-x}$$

$$\mathbb{E}[x] = \mu;\ \mathbb{V}[x] = \mu(1-\mu)$$

Binomial distribution: *distribution arising from N Bernoulli events*

$$\mathrm{Bin}(N,\mu) = p(m|N,\mu) = \binom{N}{m}\mu^m(1-\mu)^{N-m}$$

$$\mathbb{E}[m] = N\mu;\ \mathbb{V}[m] = N\mu(1-\mu)$$

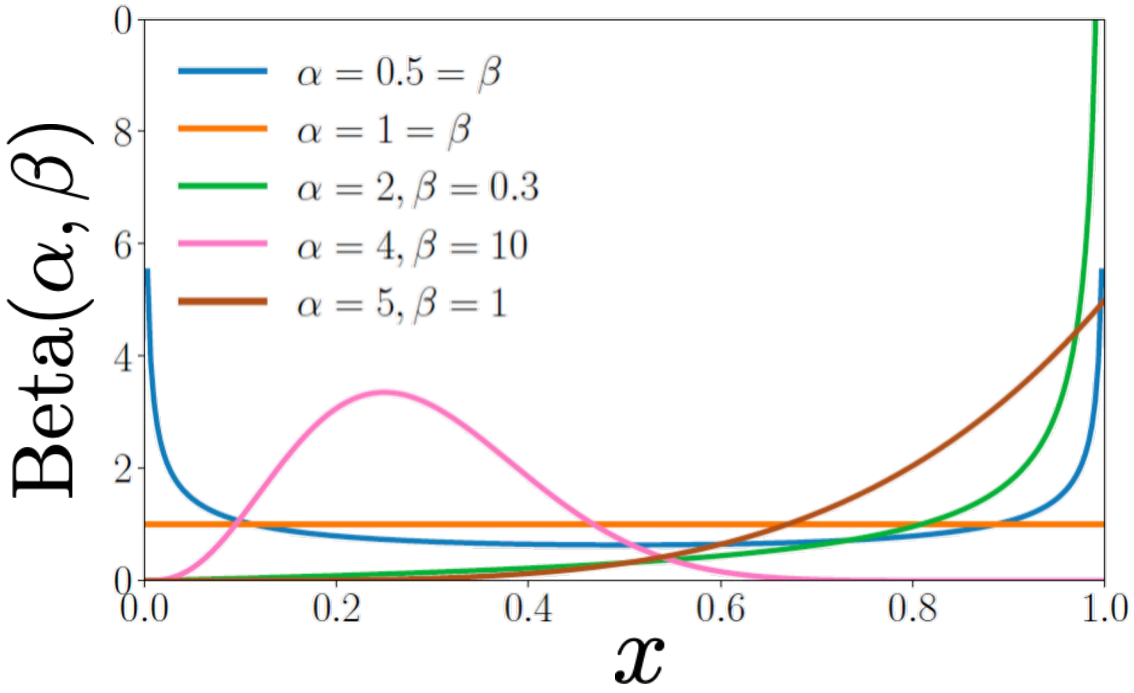
Examples of Other "Named" Distributions

Beta distribution: *distribution of a continuous variable over a finite interval [0,1]*

$$\text{Beta}(\alpha,\beta) = p(x|\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}$$

$$\Gamma(t) := \int_0^\infty x^{t-1}\exp(-x)dx$$

$$\Gamma(t+1) = t\Gamma(t)$$



- $\boldsymbol{\alpha} = \beta = 1$, uniform distribution
- $\boldsymbol{\alpha}, \beta < 1$, bimodal with spikes at 0 and 1
- $\boldsymbol{\alpha}, \beta > 1$, unimodal
- $\boldsymbol{\alpha}, \beta > 1$ and $\boldsymbol{\alpha} = \beta$, unimodal, centered, & symmetric