

Foundations of data science

summer term 2020

Michael Nüsken

Bonn-Aachen International Center for Information Technology

20 April 2020 – 16 July 2020

Section overview

Organizational

Webpage & mailing list

Digital teaching

Time & place

Hand-in & exam

Literature

Eigenvalues and eigenvectors

Best-fit subspaces and SVD

Power method for SVD

Applications of SVD

Introduction

Machine learning

High-dimensional space

*Clustering

Gaussians in high dimensions

Summary / Outro

Organizational:

Webpage & mailing list

Course page

<https://crypto.bit.uni-bonn.de/teaching/20ss/fds/>

Here, you will find notes, exercises and more.

Mailing list for discussions

20ss-fds@lists.bit.uni-bonn.de

Subscribe today!

- ▶ Moodle (cs account).
- ▶ Lectures.
- ▶ Exercise cycle.
- ▶ Tutorial.
- ▶ Mailing list.
- ▶ Discussion forum.
- ▶ Free BigBlueButton room.
- ▶ Sciebo.
 - ▶ Sciebo folder.
 - ▶ Your own sciebo space (Uni ID).
 - ▶ Sciebo app.

For more details and links see course page.

Organizational:

Time & place

We will start all meetings **c.t.** — cum tempore (latin: with time):

The first 15 minutes are free for questions and technical setup.

Lectures

- ▶ Monday, 12^{15} c.t.– 14^{00} , BigBlueButton/moodle lecture room.
- ▶ Thursday, 12^{00} c.t.– 14^{00} , BigBlueButton/moodle lecture room.

Tutorial

- ▶ Monday, 14^{00} c.t.– 16^{00} , BigBlueButton/moodle tutorial room.

Exercise cycle

- ▶ Out: Typically, Thursday, 18⁰⁰.
 - ▶ Via sciebo.
- ▶ Hand in: Thursday, 12⁰⁰ (noon).
 - ▶ Handin in the moodle.
- ▶ Corrected: in time for the tutorial.
- ▶ ≥ 50% of all points ⇒ Admitted to the exam.

Final exam (tentative)

- ▶ Monday, 17 August 2020.
- ▶ Second try: Thursday, 24 September 2020.

Major source

The course is based on the book

*Avrim Blum, John Hopcroft & Ravindran Kannan (2020).
Foundations of Data Science.*

The latest draft can be found on Blum's or Hopcroft's homepage.

Other sources

... are mentioned where appropriate.

Some supplementary texts are on the course' page.

Section overview

Organizational

Introduction

High-dimensional space

Gaussians in high dimensions

Eigenvalues and eigenvectors

Best-fit subspaces and SVD

Power method for SVD

Applications of SVD

Machine learning

*Clustering

Summary / Outro

Introduction

Organizational

Webpage & mailing list

Digital teaching

Time & place

Hand-in & exam

Literature

Introduction

High-dimensional space

Probabilities

The law of large numbers

Tail bounds

Geometry of high dimensions

Properties of the unit ball

Volume near the equator

Gaussians in high dimensions

Generating points uniformly at random
from a ball

Interludium: Inverse transform sampling

Gaussians

Fitting a spherical Gaussian to data

Separating Gaussians

Eigenvalues and eigenvectors

Basics

Symmetric matrices

Extremal properties of eigenvalues

Eigenvalues of the sum of two symmetric
matrices

Norms

Additional linear algebra

Best-fit subspaces and SVD

Introduction

Singular vectors

Singular value decomposition (SVD)

Best rank- k approximation

Left singular vectors

Power method for SVD

Applications of SVD

Centering data

Unmixing a mixture of spherical
Gaussians

Principal component analysis

Ranking documents and web pages

Machine learning

Introduction

The perceptron algorithm

Kernel functions and non-linearly separa-
ble data

Generalizing to new data

VC-dimension

VC-dimension and generalizing

*Deep learning

*Online learning

*Further current directions

*Clustering

Introduction

k -means clustering

k -center clustering

Spectral clustering

Approximation stability

High-density clusters

*Kernel methods

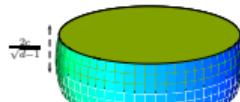
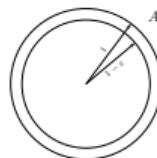
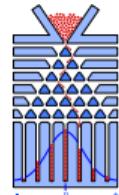
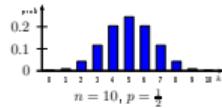
*Sparse cuts & recursive clustering

*Spectral clustering applied to social
networks

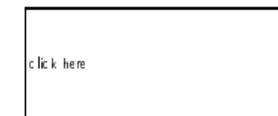
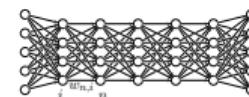
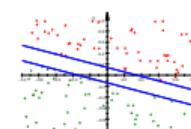
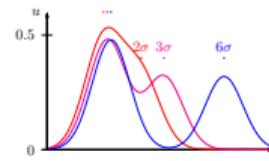
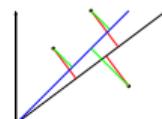
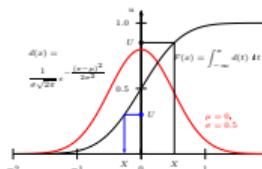
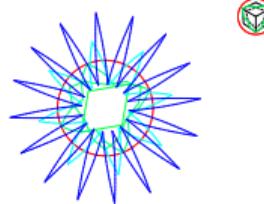
Summary / Outro

Introduction

$\{\square, \blacksquare, \blacksquare, \blacksquare, \blacksquare, \blacksquare, \blacksquare\}$



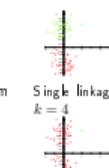
$$A_{n \times d} = \begin{matrix} U_{n \times r} \\ D_{r \times r} \\ V^T_{r \times d} \end{matrix}$$



Single linkage
 $k=4$



Lloyd's algorithm
 $k=3$



Single linkage
 $k=3$

Section overview

Organizational

Introduction

High-dimensional space

Probabilities

Random variables

Expectation

Variance

Mean or median or ...

The law of large numbers

Tail bounds

Geometry of high dimensions

Properties of the unit ball

Volume near the equator

Gaussians in high dimensions

Eigenvalues and eigenvectors

Best-fit subspaces and SVD

Power method for SVD

Applications of SVD

Machine learning

*Clustering

Summary / Outro

Long data vectors...

Given $x \in \mathbb{R}^d$, a measurement of certain features of some person, object or experiment.

► data of an epidemic

x_0 timestamp,
 x_1 susceptible persons,
 x_2 infected persons,
 x_3 recovered persons,
 x_4 deceased persons,
...

► data of a city

x_0 longitude,
 x_1 latitude,
 x_2 elevation,
 x_3 area,
 x_4 total population,
 x_5 average temperature,
 x_6 average rainfall,
 x_7 taxes,
 x_8 traffic expenses,
 x_9 school expenses,
...

► data of a weather station

x_0 timestamp,
 x_1 longitude,
 x_2 latitude,
 x_3 elevation,
 x_4 temperature,
 x_5 atmospheric pressure,
 x_6 humidity,
 x_7 wind speed,
 x_8 wind direction,
 x_9 solar radiation,
 x_{10} liquid precipitation,
 x_{11} drop size,
 x_{12} visibility,
...

► sensitive medical data

x_0 height,
 x_1 weight,

x_2 body temperature,
 x_3 blood pressure,
 x_4 heart rate,
 x_5 lung volume,
 x_6 respiratory rate,
 x_7 blood sugar level,
 x_8 respiratory rate,
 x_9 sodium (Na) concentration,
 x_{10} potassium (K) concentration,
 x_{11} cholesterol level,
 x_{12} plasma viscosity,
 x_{13} thyroid-stimulating hormone level,
 x_{14} parathyroid hormone level,
 x_{15} erythrocyte sedimentation rate,
 x_{16} urea concentration,
 x_{17} white blood cell count,
 x_{18} red blood cell count,
 x_{19} hemoglobin level,
...

Random variables

- ▶ A *random variable* X is any function on the probability space. Often all possible output values are in \mathbb{R} .
- ▶ In the discrete setting, we know its *distribution* P such that

$$\text{prob}(X = x) = P(x)$$

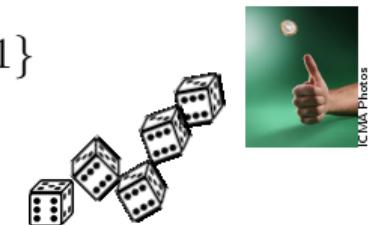
or in the continuous setting its *probability density* p

$$\text{prob}(a \leq X < b) = \int_a^b p(x) \, dx.$$

Informally: a random variable X is a blackbox that outputs values x from a known set according to some known distribution $P(x)$.

Random variables — examples

- ▶ Flip (or toss) a fair coin: $X \leftarrow \{0, 1\}$, $P(x) = \frac{1}{2}$ for $x \in \{0, 1\}$ and 0 elsewhere.
- ▶ Roll a die: $X \leftarrow \mathcal{D} := \{\square, \square, \square, \square, \square, \square\}$, $P(x) = \frac{1}{6}$.
- ▶ Uniform choice in $[0, 1]$: $X \leftarrow [0, 1]$, $p(x) = 1$ for $x \in [0, 1]$.
- ▶ Indicator variables: $X \leftarrow \{0, 1\}$, $P(1)$ is the probability of the indicated event $\mathcal{A} = \{X = 1\}$.
- ▶ Bernoulli variable: n independent, p -biased coin flips: $X = \sum_{i=1}^n X_i$, $X_i \leftarrow \{0, 1\}$, $P(1) = p$, $P(0) = 1 - p$.
- ▶ Gaussian choice in \mathbb{R} with mean μ and variance σ^2 , for short $X \sim \mathcal{N}(\mu, \sigma^2)$: $X \leftarrow \mathbb{R}$ with density $p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$.



Independence

Definition

Several random variables X_i are (*mutually*) *independent* iff

$$\begin{aligned} & \text{prob} (X_0 \in \mathcal{X}_0 \wedge \cdots \wedge X_{n-1} \in \mathcal{X}_{n-1}) \\ &= \text{prob} (X_0 \in \mathcal{X}_0) \cdot \dots \cdot \text{prob} (X_{n-1} \in \mathcal{X}_{n-1}) \end{aligned}$$

for all (nice) sets \mathcal{X}_i .

Warning: Mutual independence is much stronger than pairwise independence.

Independence — example

Take two independent fair coins $X_0, X_1 \xleftarrow{\text{IID}} \{0, 1\}$ and put $X_2 = X_0 \oplus X_1$.

That is, we assume that the distribution of the joint variable (X_0, X_1, X_2) is given by

$$\begin{aligned} \text{prob}(X_0 = x_0 \wedge X_1 = x_1) &= \underbrace{\text{prob}(X_0 = x_0)}_{=\frac{1}{2}} \cdot \underbrace{\text{prob}(X_1 = x_1)}_{=\frac{1}{2}} = \frac{1}{4} \end{aligned}$$

and

$$\text{prob}(X_0 = x_0 \wedge X_1 = x_1 \wedge X_2 = x_2) = \begin{cases} \text{prob}(X_0 = x_0 \wedge X_1 = x_1) & \text{if } x_2 = x_0 \oplus x_1 \\ 0 & \text{otherwise.} \end{cases}$$

Independence — example

Take two independent fair coins $X_0, X_1 \xleftarrow{\text{IID}} \{0, 1\}$ and put $X_2 = X_0 \oplus X_1$.

Then probabilities are

	$x = 0$	$x = 1$
$\text{prob } (X_0 = x)$	$\frac{1}{2}$	$\frac{1}{2}$
$\text{prob } (X_1 = x)$	$\frac{1}{2}$	$\frac{1}{2}$
$\text{prob } (X_2 = x)$	$\frac{1}{2}$	$\frac{1}{2}$

and ...

Independence — example

Take two independent fair coins $X_0, X_1 \xleftarrow{\text{IID}} \{0, 1\}$ and put $X_2 = X_0 \oplus X_1$.

... and joint probabilities are

prob	$X_0 = 0$	$X_0 = 1$
$X_1 = 0$	$\frac{1}{4}$	$\frac{1}{4}$
$X_1 = 1$	$\frac{1}{4}$	$\frac{1}{4}$

prob	$X_0 = 0$	$X_0 = 1$
$X_2 = 0$	$\frac{1}{4}$	$\frac{1}{4}$
$X_2 = 1$	$\frac{1}{4}$	$\frac{1}{4}$

prob	$X_1 = 0$	$X_1 = 1$
$X_2 = 0$	$\frac{1}{4}$	$\frac{1}{4}$
$X_2 = 1$	$\frac{1}{4}$	$\frac{1}{4}$

Independence — example

Take two independent fair coins $X_0, X_1 \xleftarrow{\text{IID}} \{0, 1\}$ and put $X_2 = X_0 \oplus X_1$.
... and all three together

	$X_2 = 0$		$X_2 = 1$	
X_1	$X_0 = 0$	$X_0 = 1$	$X_0 = 0$	$X_0 = 1$
$X_1 = 0$	$\frac{1}{4}$	0	0	$\frac{1}{4}$
$X_1 = 1$	0	$\frac{1}{4}$	$\frac{1}{4}$	0

Independence — example

Take two independent fair coins $X_0, X_1 \xleftarrow{\text{IID}} \{0, 1\}$ and put $X_2 = X_0 \oplus X_1$.

Bottom lines

- ▶ Each variable is uniformly distributed.
- ▶ Each pair (X_0, X_1) , (X_0, X_2) and (X_1, X_2) is independent.
- ▶ The triple (X_0, X_1, X_2) is **not** independent.

Side remark: We never needed to know which probability space is used, only that the random variables have the named joint distribution.

Expectation

For a real random variable X we define its *expected value*

$$\mathsf{E}(X) = \int_{\mathcal{X}} xp(x) \, dx,$$

or

$$\mathsf{E}(X) = \sum_{x \in \mathcal{X}} x P(x)$$

if \mathcal{X} is finite.

Theorem

Expectation is linear: $\mathsf{E}(X + Y) = \mathsf{E}(X) + \mathsf{E}(Y)$ and $\mathsf{E}(cX) = c \mathsf{E}(X)$.

... and monotone: if $X \leq Y$ then $\mathsf{E}(X) \leq \mathsf{E}(Y)$.

If X, Y are independent then $\mathsf{E}(XY) = \mathsf{E}(X) \cdot E(Y)$.

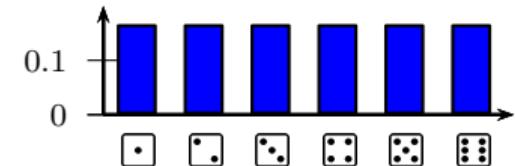
Expectation — example

Consider rolling a fair die:

$D \xleftarrow{\text{roll}} \mathcal{D} := \{\square, \square\!\cdot, \square\!\cdot\!, \square\!\cdot\!\cdot, \square\!\cdot\!\cdot\!, \square\!\cdot\!\cdot\!\cdot\}$ with uniform distribution, ie. prob $(D = x) = P_D(x) = \frac{1}{6}$ for all x .

If we identify the outcomes with the number of pips: $1 = \square$, ..., $6 = \square\!\cdot\!\cdot\!\cdot\!$ as usual in many games, we can ask for the average number of pips to see:

$$\begin{aligned} E(D) &= \sum_{x \in \mathcal{D}} x \operatorname{prob}(D = x) \\ &= \sum_{1 \leq x \leq 6} x \frac{1}{6} = \frac{21}{6} = 3.5. \end{aligned}$$



Expectation — example

Consider picking a uniformly random real number from $[0, 1]$:

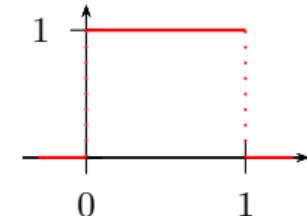
$U \xleftarrow{\text{IID}} [0, 1]$ with uniform density,

i.e. $p_U(x) = 1$ for $0 \leq x \leq 1$ and $\text{prob}(a \leq U < b) = b - a$ for $0 \leq a \leq b \leq 1$.

What is the expected value of U^2 ?

Well:

$$\begin{aligned} E(U^2) &= \int_0^1 x^2 p_U(x) \, dx \\ &= \frac{1}{3}. \end{aligned}$$



Variance

The *variance*

$$\text{var}(X) = \sigma^2(X) = E((X - E(X))^2)$$

measures the expected squared deviation from the mean value.

We have

$$\begin{aligned}\text{var}(X) &= E(X^2) - 2E(E(X)X) + E(X)^2 \\ &= E(X^2) - E(X)^2.\end{aligned}$$

Theorem

Variance is quadratic: $\text{var}(aX) = a^2 \text{var}(X)$.

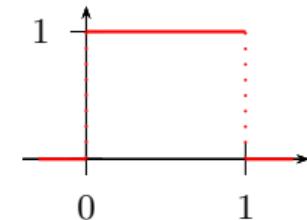
If X, Y are independent then $\text{var}(X + Y) = \text{var } X + \text{var } Y$.

Variance — example

Consider picking a uniformly random real number from $[0, 1]$:

$U \xleftarrow{\text{IID}} [0, 1]$ with uniform density,

i.e. $p_U(x) = 1$ for $0 \leq x \leq 1$ and $\text{prob}(a \leq U < b) = b - a$ for $0 \leq a \leq b \leq 1$.



What is the variance of U ?

Well, knowing $E(U) = \frac{1}{2}$, $E(U^2) = \frac{1}{3}$ we obtain:

$$\begin{aligned}\text{var}(U) &= E(U^2) - E(U)^2 \\ &= \frac{1}{3} - \frac{1}{2^2} = \frac{1}{12}.\end{aligned}$$

Random variables — example

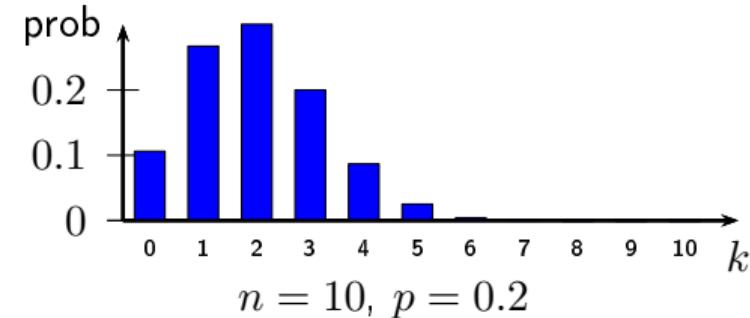
Consider a Bernoulli variable X .

Namely, start with n independent, p -biased coin flips $X_i \xleftarrow{\text{IID}} \{0, 1\}$, $\text{prob}(X_i = 1) = p$, $\text{prob}(X_i = 0) = 1 - p$ and put

$$X = \sum_{1 \leq i \leq n} X_i.$$

We find:

- ▶ $\text{prob}(X = k) = \binom{n}{k} p^k (1 - p)^{n-k}$.
- ▶ $E(X) = \sum_i E(X_i) = np$.
- ▶ $\text{var}(X) = \sum_i \text{var}(X_i) = np(1 - p)$.



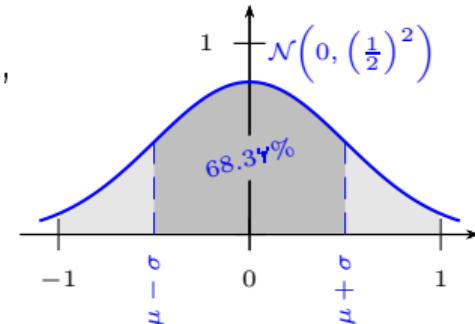
Random variables — example

Consider Gaussian choice in \mathbb{R} with mean μ and variance σ^2 ,
for short $X \sim \mathcal{N}(\mu, \sigma^2)$:

$X \leftarrow \mathbb{R}$ with density $p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$.

We find:

- ▶ $\text{prob}(a \leq X < b) = \int_a^b p(x) dx.$
- ▶ $E(X) = \mu.$
- ▶ $\text{var}(X) = \sigma^2.$
- ▶ Remark: $\text{prob}(|X - \mu| < \sigma) = 68.27\%$. (See 68-95-99.7 rule.)



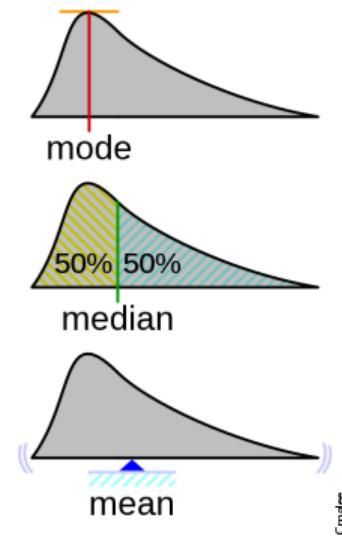
Mean or median or ...

- ▶ Sometimes the expected value, ie. the mean, does not tell what we expect.
- ▶ Outliers are problematic.

Possible alternative: median.

Definition

For a random variable $X \leftarrow \mathbb{R}$, the median is the value such that half of the density is below and half is above.



High-dimensional space:

The law of large numbers

High-dimensional space or: How to define probabilities?

Consider rolling a die $D \xleftarrow{\text{roll}} \{\square, \square\circ, \circ\square, \square\square, \square\circ\circ, \circ\square\circ\}$.

Before Kolmogorov people defined

$$\text{prob}(D = \square\square) := \lim_{n \rightarrow \infty} \frac{\#\{i < n \mid D_i = \square\square\}}{\#\{i < n\}}.$$

or —identifying $\square\square = 4 \in \mathbb{R}$ and so on—

$$\mathbb{E}(D) := \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i < n} D_i.$$

But that approach was cumbersome to use. :(

Modern approach

We postulate probabilities **axiomatically** and work with random variables. This results in a simple calculus.

And we gain the ‘limit definitions’ back as the **Law of Large Numbers**.

Theorem (Markov's inequality)

Let X be a non-negative random variable. Then for $a > 0$

$$\text{prob}(X \geq a) \leq \frac{E(X)}{a}.$$

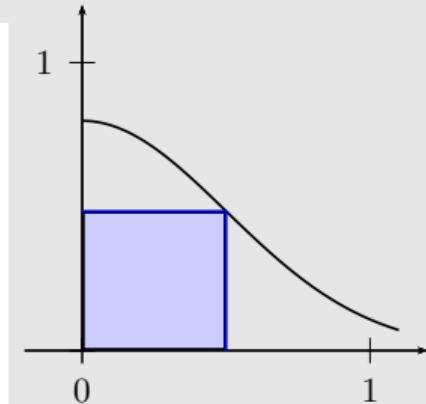
Proof. . . .



Proof.

Assume X is continuous r.v. with density p_X . Then

$$\begin{aligned} E(X) &= \int_0^\infty x p_X(x) \, dx \\ &= \int_0^a \underbrace{x}_{\geq 0} p_X(x) \, dx + \int_a^\infty \underbrace{x}_{\geq a} p_X(x) \, dx \\ &\geq a \int_a^\infty p_X(x) \, dx = a \operatorname{prob}(X \geq a) \end{aligned}$$



□

Theorem (Chebyshev's inequality)

Let X be a random variable. Then for $c > 0$ we have

$$\text{prob}(|X - E(X)| \geq c) \leq \frac{\text{var}(X)}{c^2}.$$

Proof. . . .

□

Proof.

Apply Markov's inequality to $Y = |X - E(X)|^2$. Note $E(Y) = \text{var}(X)$ and so

$$\text{prob}(|X - E(X)| \geq c) = \text{prob}\left(|X - E(X)|^2 \geq c^2\right) \leq \frac{E(Y)}{c^2} = \frac{\text{var}(X)}{c^2}.$$

□

Theorem ([Weak] law of large numbers)

Assume X_i are independent, identically distributed random variables with mean μ and variance σ^2 . Then

$$\frac{1}{n} \sum_{1 \leq i \leq n} X_i \xrightarrow{n \rightarrow \infty} \mu$$

in the sense that

$$\text{prob} \left(\left| \frac{1}{n} \sum_{1 \leq i \leq n} X_i - \mu \right| \geq \varepsilon \right) \leq \frac{\sigma^2}{n\varepsilon^2}.$$

Proof. . .

□

Proof.

Consider $X = \frac{1}{n} \sum_{1 \leq i \leq n} X_i$. Then $E(X) = \frac{1}{n} \sum_i E(X_i) = \mu$ and

$$\text{var}(X) = \frac{1}{n^2} \text{var}\left(\sum_i X_i\right) = \frac{1}{n^2} \sum_i \text{var}(X_i) = \frac{\sigma^2}{n}.$$

Applying Chebyshev's inequality yields

$$\text{prob}(|X - E(X)| \geq \varepsilon) \leq \frac{\sigma^2}{n\varepsilon^2}.$$



Remark

The sample size n for estimating μ does not depend on the size of the universe, but only on ε , σ and δ :

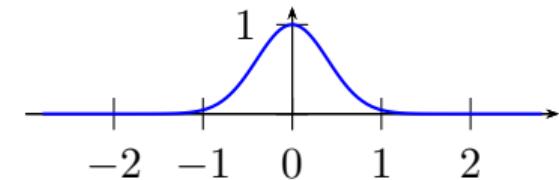
$$\text{prob} \left(\left| \frac{1}{n} \sum_{1 \leq i \leq n} X_i - \mu \right| \geq \varepsilon \right) \leq \frac{\sigma^2}{n\varepsilon^2} \leq \delta.$$

In other words: if the sample size n is at least $\frac{\sigma^2}{\delta\varepsilon^2}$ then

$$\text{prob} \left(\left| \frac{1}{n} \sum_{1 \leq i \leq n} X_i - \mu \right| \geq \varepsilon \right) \leq \delta.$$

Application

Consider a d -dimensional random point $z \in \mathbb{R}^d$ with independently drawn coordinates $z_i \xrightarrow{\text{ iid }} \mathcal{N}(0, \frac{1}{2\pi})$.



Thus each z_i is a copy of a random variable Z with distribution $\mathcal{N}(0, \frac{1}{2\pi})$.

By the law of large numbers $\frac{1}{d} \|z\|_2^2 = \frac{1}{d} \sum_i z_i^2 \xrightarrow{d \rightarrow \infty} E(Z^2)$ in probability. Notice that $E(Z^2) = \text{var } Z + E(Z)^2 = \frac{1}{2\pi}$. Thus with high probability

$$\|z\|_2 \approx \sqrt{\frac{d}{2\pi}}.$$

In particular, the probability for $z \in B^d$, namely $\|z\|_2 \leq 1$, is tiny. Quite contrary to the case $d = 1$.

Theorem (Master tail bounds theorem)

Let $X = \sum_{i < n} X_i$ with independent random variables X_i with zero mean and variance at most $2\sigma^2$.

Let $0 \leq a \leq \sqrt{2} \cdot n\sigma^2$.

Assume that $\frac{1}{s!} |\mathbb{E}(X_i^s)| \leq \sigma^2$ for $s = 2, 3, 4, \dots, \left\lfloor \frac{a^2}{4n\sigma^2} \right\rfloor$.

Then

$$\text{prob}(|X| \geq a) \leq 3e^{-\frac{a^2}{12n\sigma^2}}.$$

Remark: Given $X_i = Y_i - \mathbb{E}(Y_i)$ we have

- ▶ $\mathbb{E}(X_i) = 0$ and
- ▶ $\text{prob}\left(\left|\frac{1}{n} \sum_{i < n} Y_i - \mathbb{E}\left(\frac{1}{n} \sum_{i < n} Y_i\right)\right| \geq \frac{a}{n}\right) = \text{prob}(|X| \geq a)$.

Proof. . .



Remarks

As $X = \sum_i X_i$ is a sum of n random variables, each with standard deviation at most σ , we consider $a = \alpha \cdot n\sigma$.

- ▶ Then for $0 \leq \alpha \leq \sqrt{2}\sigma$ we obtain

$$\text{prob}(|X| \geq \alpha \cdot n\sigma) \leq 3e^{-\frac{\alpha^2}{12} \cdot n},$$

provided we have good bounds on higher moments, namely $\frac{1}{s!} |\mathbb{E}(X_i^s)| \leq \sigma^2$ for $2 \leq s \leq \frac{\alpha^2}{4}n$.

This bound decreases exponentially with n .

- ▶ Applying Chebyshev —as in the law of large numbers— only yields

$$\text{prob}(|X| \geq \alpha \cdot n\sigma) \leq \frac{1}{\alpha^2 \cdot n}.$$

This bound does decrease with n but much slower.

Of course, the needed assumptions are much weaker here.

Proof.

Consider $r \leq s \leq \frac{n\sigma^2}{2}$. We prove the slightly better bound

$$\text{prob}(|X| \geq a) \leq \left(\frac{2r n \sigma^2}{a^2} \right)^{\frac{r}{2}}.$$

Markov: $\text{prob}(X^r \geq a^r) \leq \frac{\mathbb{E}(X^r)}{a^r}$ for $r \in 2\mathbb{N}$.

Recall:

- ▶ $\mathbb{E}(X_i^0) = 1$.
- ▶ $\mathbb{E}(X_i^1) = 0$.
- ▶ $\mathbb{E}(X_i^2) \leq 2\sigma^2 + E(X_i)^2 = 2\sigma^2$.
- ▶ $\frac{1}{s!} \mathbb{E}(X_i^s) \leq \sigma^2$ for $3 \leq r \leq s$.

Using independence we expand

$$\mathbb{E}(X^r) = \sum_{\substack{R \in \mathbb{N}^n, \\ \sum R_i = r}} \binom{r}{R} \prod_i \mathbb{E}(X_i^{R_i}).$$

Split the sum into $1 + \frac{r}{2}$ pieces according to R :

- ▶ R contains a 1: drop term!
- ▶ R has t non-zero entries, each at least 2.

We arrive at a bound

$$\mathbb{E}(X^r) \leq r! \sum_{t \leq \frac{r}{2}} \sum_{\substack{R, \\ \sum R_i = r, \\ \text{wt}(R) = t}} \sigma^{2t}.$$

The inner sum has $\binom{n}{t} \cdot \binom{r-2t+t-1}{t-1}$ terms. (Pick the t places where R shall be non-zero: $\binom{n}{t}$ options. Put each of them 2 and distribute the remaining $r - 2t$ arbitrarily to the t places.)

The remainder is clever, but elementary analysis. This count is bounded by

$$\binom{n}{t} \cdot \binom{r-2t+t-1}{t-1} \sigma^{2t} \leq \frac{(n\sigma^2)^t}{t!} 2^{r-t-1} =: h(t).$$

For $t \leq \frac{r}{2} \leq \frac{n\sigma^2}{4}$ we have

$$\frac{h(t)}{h(t-1)} = \frac{n\sigma^2}{2t} \geq 2.$$

Thus $h(t) \leq h\left(\frac{r}{2}\right) 2^{t-\frac{r}{2}}$ and $\sum_{1 \leq t \leq \frac{r}{2}} h(t) \leq 2h\left(\frac{r}{2}\right) = \frac{(n\sigma^2)^{\frac{r}{2}}}{\left(\frac{r}{2}\right)!} 2^{\frac{r}{2}}$. Thus by the Markov inequality we obtain

$$\begin{aligned} \text{prob}(|X| > a) &= \text{prob}(X^r > a^r) \leq \frac{\mathbb{E}(X^r)}{a^r} \\ &\leq \frac{r!(n\sigma^2)^{\frac{r}{2}} 2^{\frac{r}{2}}}{\left(\frac{r}{2}\right)! a^r} =: g(r) \\ &\leq \left(\frac{2r n \sigma^2}{a^2} \right)^{\frac{r}{2}} \end{aligned}$$

For even r , $\frac{g(r)}{g(r-2)} = \frac{4(r-1)n\sigma^2}{a^2}$. This is at most 1, ie. $g(r)$ decreases, as long as $r - 1 \leq \frac{a^2}{4n\sigma^2}$. This motivates us, taking $r = 2 \left\lfloor \frac{a^2}{12n\sigma^2} \right\rfloor$ and we obtain

$$\left(\frac{2r n \sigma^2}{a^2} \right)^{\frac{r}{2}} \leq \left(\frac{a^2}{6n\sigma^2} \frac{2n\sigma^2}{a^2} \right)^{\frac{r}{2}} = 3^{-\frac{r}{2}} \leq e^{-\frac{r}{2}} \leq e^{1 - \frac{a^2}{12n\sigma^2}} \leq 3e^{-\frac{a^2}{12n\sigma^2}}. \quad \square$$

(Actually, estimating $3^{-\frac{r}{2}} \leq 3e^{-\frac{\ln 3 \cdot a^2}{12n\sigma^2}}$ with $\frac{12}{\ln 3} = 10.92$ would be slightly better. Or $r = 2 \left\lfloor \frac{a^2}{4en\sigma^2} \right\rfloor$ would give the bound $e^{1 - \frac{a^2}{4en\sigma^2}} \leq 3e^{-\frac{a^2}{4en\sigma^2}}$ with $4e = 10.87$.)

High-dimensional space:

Tail bounds

Summary: main ideas in the proof

Given $X = \sum_{i < n} X_i$.

- ▶ For simplicity, assume zero mean, ie. $E(X_i) = 0$.
- ▶ Goal: estimate $\text{prob}(|X| \geq a)$.
- ▶ Idea:
 - ▶ Use Markov on even powers of X :

$$\text{prob}(|X| \geq a) = \text{prob}(X^r \geq a^r) \leq \frac{E(X^r)}{a^r}$$

for $r \in 2\mathbb{N}$ so that we can drop the absolute value.

- ▶ Estimate $E(X^r)$ using higher moments $E(X_i^s)$ of the X_i .
- ▶ Use the best available r .

- ▶ For estimating $E(X^r)$ we use the following:

- ▶ The multinomial expansion:

$$\left(\sum_{i < n} X_i\right)^r = \sum_{\substack{R \in \mathbb{N}^n, \\ \sum R = r}} \binom{r}{R} \prod_i X_i^{R_i}$$

where $\binom{r}{R} = \frac{r!}{\prod_i (R_i!)}$.

- ▶ Bounds on $E(X_i^s)$ for all needed values s .
- ▶ The remainder is a lengthy, clever, but elementary analysis.
 - ▶ We do not find the best r , a good one is enough.
 - ▶ We weaken the bound slightly to get a manageable final expression.

High-dimensional space:

Tail bounds

	condition	tail bound
Markov	$X \geq 0$	$\text{prob}(X \geq a) \leq \frac{\mathbb{E}(X)}{a}$
Chebychev	X real	$\text{prob}(X - \mathbb{E}(X) \geq a) \leq \frac{\text{var}(X)}{a^2}$
higher moments	$r \in 2\mathbb{N}$	$\text{prob}(X ^r \geq a) \leq \frac{\mathbb{E}(X^r)}{a^r}$
Master tail bound	$X = \sum_i X_i$, X_i independent, $\mathbb{E}(X_i) = 0$, $\forall 2 \leq s \leq \left\lfloor \frac{a^2}{4n\sigma^2} \right\rfloor : \frac{1}{s!} \mathbb{E}(X_i^s) \leq \sigma^2$	$\text{prob}(X \geq a) \leq 3e^{-\frac{a^2}{12n\sigma^2}}$
Chernoff	$X = \sum_i X_i$, each $X_i \xrightarrow{\text{IID}} \{0, 1\}$ iid.	$\text{prob}(X - \mathbb{E}(X) \geq \varepsilon \mathbb{E}(X)) \leq 3e^{-c\varepsilon^2 E(X)}$
Gaussian annulus	$X = \sqrt{\sum_i X_i^2}$, $X_i \sim \mathcal{N}(0, 1)$ iid., $\beta \leq \sqrt{n}$	$\text{prob}(X - \sqrt{n} \geq \beta) \leq 3e^{-\frac{1}{48}\beta^2}$
power law for X_i , order $k \geq 4$	$X = \sum_i X_i$, X_i iid., $\varepsilon \leq \frac{1}{k^2}$	$\text{prob}(X - \mathbb{E}(X) \geq \varepsilon \mathbb{E}(X)) \leq \left(\frac{4}{\varepsilon^2 kn}\right)^{\frac{k-3}{2}}$

... many other 'large deviation' results.

High-dimensional space:

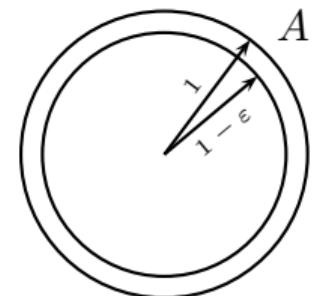
Geometry of high dimensions

What's the near surface volume of some object in \mathbb{R}^d ?

Well, $\text{vol}((1 - \varepsilon)A) = (1 - \varepsilon)^d \cdot \text{vol } A$.

Thus, assuming $(1 - \varepsilon)A \subset A$ as for a convex body containing the origin, we obtain:

$$\text{vol}(A \setminus (1 - \varepsilon)A) = \left(1 - (1 - \varepsilon)^d\right) \text{vol } A$$



and

$$1 - (1 - \varepsilon)^d \geq 1 - e^{-\varepsilon d} \xrightarrow{d \rightarrow \infty} 1.$$

In other words: most of the volume is concentrated near the surface!

Lemma

$$\frac{\text{vol}(\frac{1}{d}\text{-annulus})}{\text{vol}(B^d)} = 1 - \left(1 - \frac{1}{d}\right)^d \geq 1 - e^{-1} = 0.634 \quad \text{for the } \varepsilon\text{-annulus } B^d \setminus (1 - \varepsilon)B^d.$$

High-dimensional space:

Properties of the unit ball

Theorem (Volume and surface area)

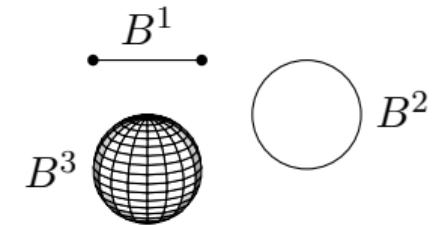
Let B^d be the d -dimensional unit ball. Then

- ▶ $\text{vol}(B^d) = \frac{2\pi^{\frac{d}{2}}}{d\Gamma(\frac{d}{2})}.$
- ▶ $\text{surface}(B^d) = \frac{2\pi^{\frac{d}{2}}}{\Gamma(\frac{d}{2})}.$

Proof. . . .

In particular,

- ▶ $\text{vol}(rB^1) = 2r$, $\text{surface}(rB^1) = 2.$
- ▶ $\text{vol}(rB^2) = \pi r^2$, $\text{surface}(rB^2) = 2\pi r.$
- ▶ $\text{vol}(rB^3) = \frac{4}{3}\pi r^3$, $\text{surface}(rB^3) = 4\pi r^2.$
- ▶ $\text{vol}(rB^4) = \frac{1}{2}\pi^2 r^4$, $\text{surface}(rB^4) = 2\pi^2 r^3.$



Visualize B^4 ?

Excursion to Γ

The Γ -function is an extension of the factorials.

It is *the* logarithmically convex, analytic function with $\Gamma(1) = 1$, $\Gamma(z+1) = z\Gamma(z)$.

For $\operatorname{Re}(z) > 0$ it can be defined as

$$\Gamma(z) = \int_0^\infty x^{z-1} e^{-x} \, dx .$$

One finds

- ▶ $\Gamma(n+1) = n!$.
- ▶ $\Gamma(\frac{1}{2}) = \sqrt{\pi}$.
- ▶ $\Gamma(n + \frac{1}{2}) = \frac{(2n-1)!!}{2^n} \sqrt{\pi}$.

▶ $\left(\frac{n}{2}\right)^{\frac{n}{2}} \leq \Gamma(n+1) \leq n^n$ for $n > 0$.

▶ Stirling's formula:

$$\Gamma(x+1) = \sqrt{2\pi} x^{x+\frac{1}{2}} e^{-x} e^{\mu(x)} \text{ with } \frac{1}{12x+1} < \mu(x) < \frac{1}{12x} .$$

Proof.

We could compute in Cartesian coordinates:

$$\begin{aligned}\text{vol}(B^d) &= \int_{x_0^2+x_1^2+\dots+x_{d-1}^2 \leq 1} dx_{d-1} \dots dx_1 dx_0 \\ &= \int_{x_0=-1}^{x_0=1} \int_{x_1=-\sqrt{1-x_0^2}}^{x_1=\sqrt{1-x_0^2}} \dots \\ &\quad \int_{x_{d-1}=-\sqrt{1-x_0^2-\dots-x_{d-2}^2}}^{x_{d-1}=\sqrt{1-x_0^2-\dots-x_{d-2}^2}} dx_{d-1} \\ &\quad \dots dx_1 dx_0 \\ &= \int_{x_0=-1}^{x_0=1} \text{vol} \left(\sqrt{1-x_0^2} B^{d-1} \right) dx_0.\end{aligned}$$

Easier in polar coordinates:

$$\begin{aligned}\text{vol}(B^d) &= \int_{S^d} \int_{r=0}^1 r^{d-1} dr d\Omega \\ &= \underbrace{\int_{r=0}^1 r^{d-1} dr}_{\frac{1}{d}} \cdot \int_{S^d} d\Omega = \frac{1}{d} \underbrace{\text{surface}(B^d)}_{=\text{vol}(\partial B^d)}\end{aligned}$$

Instead, consider

$$I(d) := \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} e^{-\sum_{i< d} x_i^2} dx_{d-1} \dots dx_1 dx_0.$$

In Cartesian coordinates we evaluate this to

$$I(d) = \left(\underbrace{\int_{-\infty}^{\infty} e^{-x^2} dx}_{=\sqrt{\pi}} \right)^d = \pi^{\frac{d}{2}}.$$

In polar coordinates, we obtain

$$I(d) = \underbrace{\int_{S^d} d\Omega}_{=\text{surface}(B^d)} \cdot \underbrace{\int_0^{\infty} e^{-r^2} r^{d-1} dr}_{=\frac{1}{2} \Gamma\left(\frac{d}{2}\right)}.$$

The integral over r can be transformed by using the substitution $t = r^2$, $dt = 2r dr$ and so

$$\int_0^{\infty} e^{-r^2} r^{d-2} \frac{1}{2} \cdot 2r dr = \frac{1}{2} \int_0^{\infty} e^{-t} t^{\frac{d}{2}-1} dt = \frac{1}{2} \Gamma\left(\frac{d}{2}\right).$$

Combining yields $\text{surface}(B^d) = \frac{\pi^{\frac{d}{2}}}{\frac{1}{2} \Gamma\left(\frac{d}{2}\right)}$.

□

High-dimensional space:

Properties of the unit ball

Unit ball volume tends to zero

Recall $\text{vol}(rB^d) = r^d \frac{2\pi^{\frac{d}{2}}}{d\Gamma(\frac{d}{2})}$.

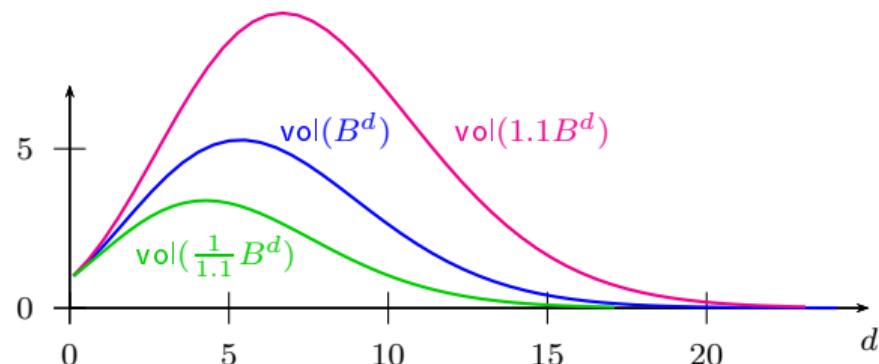
Using Stirling's formula $\text{vol}(rB^d) \sim \frac{1}{\sqrt{2\pi}} e^{-\frac{d}{2} \ln \frac{d}{2}} + (2 \ln r - \ln \pi + 1) \frac{d}{2} - \frac{1}{2} \ln \frac{d}{2}$.

Corollary

For each $r > 0$ we have

$$\lim_{d \rightarrow \infty} \text{vol}(rB^d) = 0.$$

□



Other perspective: How to choose the radius so that rB^d has volume one? How does this radius behave with $d \rightarrow \infty$?

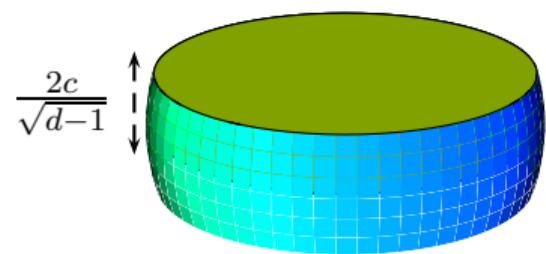
High-dimensional space:

Volume near the equator

Theorem

Let T be the *tropical slice*, or equator slice, of the unit ball B^d with $|x_0| \leq \frac{c}{\sqrt{d-1}}$. If $c \geq 1$ and $d \geq 3$ then

$$\text{vol}(T) \geq \left(1 - \frac{2}{c} e^{-\frac{c^2}{2}}\right) \text{vol}(B^d).$$



In other words, most of the volume of a high-dimensional ball is concentrated near the equator.

Proof. . . .

□

Proof.

Suffices to show: at most $\frac{2}{c}e^{-\frac{c^2}{2}}$ of the volume of upper hemisphere H of B^d , namely where $x_0 \geq 0$, has $x_0 > \frac{c}{\sqrt{d-1}} =: \alpha$.

Consider the cap $A := B^d \cap \{x_0 \geq \alpha\}$ and the upper hemisphere $H := B^d \cap \{x_0 \geq 0\}$. Let $r(s)$ denote the radius of the $(d-1)$ -ball $B^d \cap \{x_0 = s\}$. Its radius is $r(s) = (1 - s^2)^{\frac{1}{2}}$.

Then

$$\begin{aligned}\text{vol } A &= \int_{\alpha}^1 \text{vol}(r(x_0) \cdot B^{d-1}) \, dx_0 \\ &= \int_{\alpha}^1 (\underbrace{1 - x_0^2}_{\leq e^{-x_0^2}})^{\frac{d-1}{2}} \, dx_0 \cdot \text{vol } B^{d-1} \\ &\leq \int_{\alpha}^1 1 \cdot e^{-\frac{d-1}{2}x_0^2} \, dx_0 \cdot \text{vol } B^{d-1} \\ &\leq \frac{1}{(d-1)\alpha} \underbrace{\int_{\alpha}^1 (d-1)x_0 e^{-\frac{d-1}{2}x_0^2} \, dx_0}_{= \left[-e^{-\frac{d-1}{2}x_0^2} \right]_{x_0=\alpha}^1} \cdot \text{vol } B^{d-1} \\ &\leq \frac{1}{c\sqrt{d-1}} e^{-\frac{c^2}{2}} \cdot \text{vol } B^{d-1}\end{aligned}$$

and

$$\begin{aligned}\text{vol } H &\geq \text{vol } ([0, \beta] \times r(\beta)B^{d-1}) \\ &= \beta(1 - \beta^2)^{\frac{d-1}{2}} \cdot \text{vol } B^{d-1} \\ &\geq \beta \left(1 - \frac{d-1}{2}\beta^2 \right) \cdot \text{vol } B^{d-1}\end{aligned}$$

using $(1 - x)^a \geq 1 - ax$ for $a \geq 1$ and $x \in [0, 1]$, which is applicable since $d \geq 3$ and $\beta \in [0, 1]$. Choosing $\beta = \frac{1}{\sqrt{d-1}}$ we get

$$\text{vol } H \geq \frac{1}{2\sqrt{d-1}} \cdot \text{vol } B^{d-1}.$$

Together

$$\frac{\text{vol } A}{\text{vol } H} \leq \frac{2}{c}e^{-\frac{c^2}{2}}.$$

And that proves the theorem. □



Near orthogonality

Idea:

- ▶ Picking a first random vector from the unit ball is fixing the poles.
- ▶ Picking a second random vector is with high probability from the equator region.

Theorem

Consider drawing n points $x^{(0)}, x^{(1)}, \dots, x^{(n-1)} \in B^d$ randomly, $n \leq d$. Then with probability $1 - O\left(\frac{1}{n}\right)$ we have

1. *almost unit vectors*: for all i we have $\|x^{(i)}\|_2 \geq 1 - \frac{2\ln n}{d}$, and
2. *almost pairwise orthogonal*: for all $i \neq j$ we have $|\langle x^{(i)}, x^{(j)} \rangle| \leq \frac{\sqrt{6\ln n}}{\sqrt{d-1}}$.

Proof. . .



Proof.

Put together

- ▶ $\text{prob} \left(\|x^{(i)}\|_2 < 1 - \varepsilon \right) \leq e^{-\varepsilon d}$ with $\varepsilon = \frac{2 \ln n}{d}$.
- ▶ $\text{prob} \left(|\langle x^{(i)} | x^{(j)} \rangle| \geq \frac{c}{\sqrt{d-1}} \right) \leq 1 - \frac{\text{vol}(T)}{\text{vol}(B^d)} \leq \frac{2}{c} e^{-\frac{c^2}{2}}$ with $c = \sqrt{6 \ln n}$.

Actually, we use the first statement n times and the second statement $\binom{n}{2}$ times. The overall probability that one of the statements of the theorem does *not* hold is thus bound by

$$n \cdot \underbrace{e^{-\varepsilon d}}_{=n^{-2}} + \binom{n}{2} \cdot \frac{2}{c} \underbrace{e^{-\frac{c^2}{2}}}_{=n^{-3}} \in \mathcal{O}(n^{-1})$$

as claimed. □

High-dimensional space:

Volume near the equator

Again: Unit ball volume tends to zero

Consider the intersection of equator regions wrt. to each axis. The resulting cube with side length $\frac{2c}{\sqrt{d-1}}$ contains at least $1 - \frac{2d}{c}e^{-\frac{c^2}{2}}$ of the ball volume.

Taking $c = 2\sqrt{\ln d}$, $d \geq 2$, this is at least half the ball volume.

But the volume of that cube is

$$\left(\frac{4\sqrt{\ln d}}{\sqrt{d-1}} \right)^d$$

which obviously tends to zero.

We obtain again:

$$\text{vol}(B^d) \xrightarrow{d \rightarrow \infty} 0.$$

Summary

- ▶ Most points of a ball are near its surface.
- ▶ Most points of a ball are near the equator.
- ▶ Most point tuples are almost an orthonormal basis.

High-dimensional space:

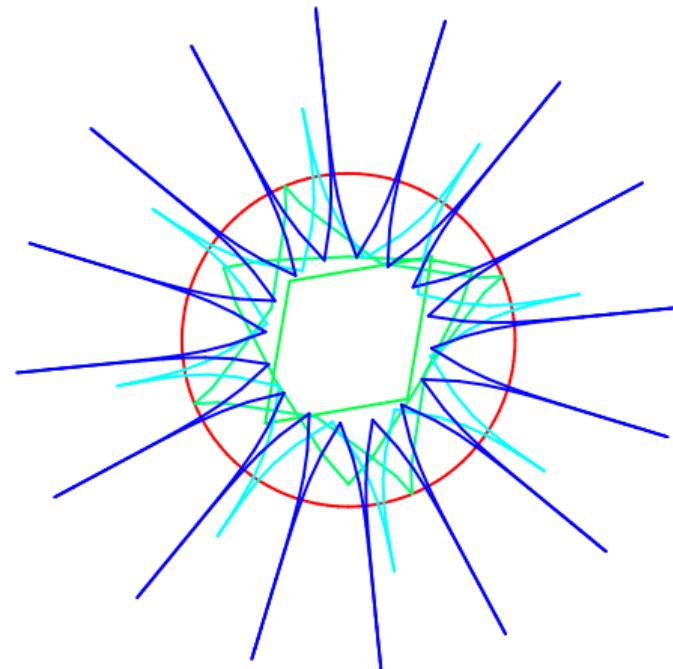
Volume near the equator

Sphere volume versus cube volume around one corner

Consider a **sphere** of radius 1
and a **d -hypercube** of 'radius' $\frac{1}{2}$, ie.

with corners $\frac{1}{2} \begin{bmatrix} \pm 1 \\ \pm 1 \\ \vdots \\ \pm 1 \end{bmatrix}$.

We draw a $2d$ cut of the sphere.
And we look at a hypercube and draw
some neighbours of one of its corners
each as far out as it should and such
that the areas reflect the actual
volumes in their vicinity.



$$d = 2, 3, 4, 8, 16$$

High-dimensional space:

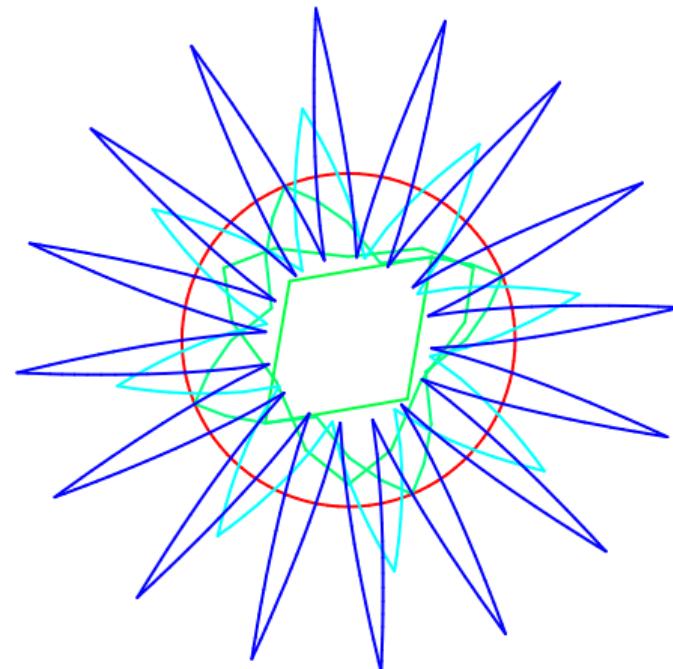
Volume near the equator

Sphere volume versus cube volume accumulated

Consider a **sphere** of radius 1
and a **d -hypercube** of 'radius' $\frac{1}{2}$, ie.

with corners $\frac{1}{2} \begin{bmatrix} \pm 1 \\ \pm 1 \\ \vdots \\ \pm 1 \end{bmatrix}$.

We draw a $2d$ cut of the sphere.
And we look at a hypercube and draw
some neighbours of one of its corners
each as far out as it should and such
that the areas reflect the associated
volumes accumulated over all corners.



$$d = 2, 3, 4, 8, 16$$

Section overview

Organizational

Introduction

High-dimensional space

Gaussians in high dimensions

Generating points uniformly at random from
a ball

Interludium: Inverse transform sampling

Gaussians

Fitting a spherical Gaussian to data

Separating Gaussians

Eigenvalues and eigenvectors

Best-fit subspaces and SVD

Power method for SVD

Applications of SVD

Machine learning

*Clustering

Summary / Outro

Long data vectors...

Given $x \in \mathbb{R}^d$, a measurement of certain features of some person, object or experiment.

► data of an epidemic

x_0 timestamp,
 x_1 susceptible persons,
 x_2 infected persons,
 x_3 recovered persons,
 x_4 deceased persons,
...

► data of a city

x_0 longitude,
 x_1 latitude,
 x_2 elevation,
 x_3 area,
 x_4 total population,
 x_5 average temperature,
 x_6 average rainfall,
 x_7 taxes,
 x_8 traffic expenses,
 x_9 school expenses,
...

► data of a weather station

x_0 timestamp,
 x_1 longitude,
 x_2 latitude,
 x_3 elevation,
 x_4 temperature,
 x_5 atmospheric pressure,
 x_6 humidity,
 x_7 wind speed,
 x_8 wind direction,
 x_9 solar radiation,
 x_{10} liquid precipitation,
 x_{11} drop size,
 x_{12} visibility,
...

► sensitive medical data

x_0 height,
 x_1 weight,

x_2 body temperature,
 x_3 blood pressure,
 x_4 heart rate,
 x_5 lung volume,
 x_6 respiratory rate,
 x_7 blood sugar level,
 x_8 respiratory rate,
 x_9 sodium (Na) concentration,
 x_{10} potassium (K) concentration,
 x_{11} cholesterol level,
 x_{12} plasma viscosity,
 x_{13} thyroid-stimulating hormone level,
 x_{14} parathyroid hormone level,
 x_{15} erythrocyte sedimentation rate,
 x_{16} urea concentration,
 x_{17} white blood cell count,
 x_{18} red blood cell count,
 x_{19} hemoglobin level,
...

Gaussians in high dimensions

Long data vectors...

Given $x \in \mathbb{R}^d$, a measurement of certain features of some person, object or experiment.

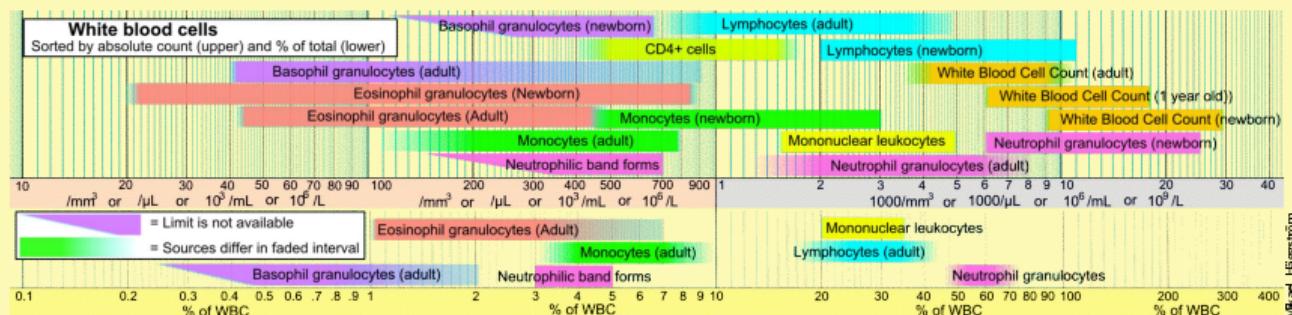
► data of an epidemic

- x_0 timestamp,
- x_1 susceptible persons,
- x_2 infected persons,
- x_3 recovered persons,
- x_4 deceased persons
- ...

► data of a city

- x_0 longitude,
- x_1 latitude,
- x_2 elevation,
- x_3 area,
- x_4 total population,
- x_5 average temperature,
- x_6 average rainfall,
- x_7 taxes,
- x_8 traffic expenses,
- x_9 school expenses,
- ...

Example of reference ranges of white blood cells



Notice: all these numbers are assumed to vary in certain ranges.

We interpret them as random values.

► sensitive medical data

- x_{10} rate,
- x_{11} urea concentration,
- x_{12} white blood cell count,
- x_{13} red blood cell count,

Often a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ with an adapted mean μ and standard deviation σ is the first choice.

Gaussians in high dimensions:

Generating points uniformly at random from a ball

Application

Question

How to generate a random point x on the surface ∂B^d of a d -dimensional ball?

That is, pick $x \in \mathbb{R}^d$ with $\|x\|_2 = 1$.

Gaussians in high dimensions:

Generating points uniformly at random from a ball

Method 1 for a random point on the surface

- ▶ Repeat
- ▶ Pick $x_i \xleftarrow{\text{IID}} [-1, 1]$ uniformly for each $i \leq d$.
- ▶ until $0 < \|x\|_2 \leq 1$.
- ▶ Return the normalized vector $\frac{1}{\|x\|_2}x$.

Problem: Method 0 is non-uniform! There are more results near the corners of the cube.

Resort: Discard vectors that are too long.

Problem: In high dimensions we throw away too many points
since $\text{vol}(B^d) \xrightarrow{d \rightarrow \infty} 0$.

Resort: Use a different distribution for $x_i \dots$

The easy part: uniform sampling

- ▶ How to generate $U \leftarrow [0, 1[$ uniformly by an algorithm?
- ▶ We can at best have some, say, fixed-point approximation accurate to m bits.

Given: m unbiased random bits $X_i \leftarrow \{0, 1\}$, $i \in \mathbb{N}_{<m}$.

1. **Return** $\tilde{U} \leftarrow \sum_{0 < i \leq m} X_{i-1} 2^{-i}$, ie. in binary: $\tilde{U} = 0.X_0X_1\dots X_{m-1}$.

- ▶ Then for $0 \leq a \leq b \leq 1$

$$\text{prob}\left(a \leq \tilde{U} < b\right) = \frac{\lceil 2^m b \rceil - \lceil 2^m a \rceil}{2^m} = (b - a) + \mu 2^{-m}$$

with $\mu \in [-1, 1]$.

This approximates uniform distribution on $[0, 1[$ as good as could be expected.

- ▶ It uses m unbiased random bits and no other operation.

Gaussians in high dimensions:

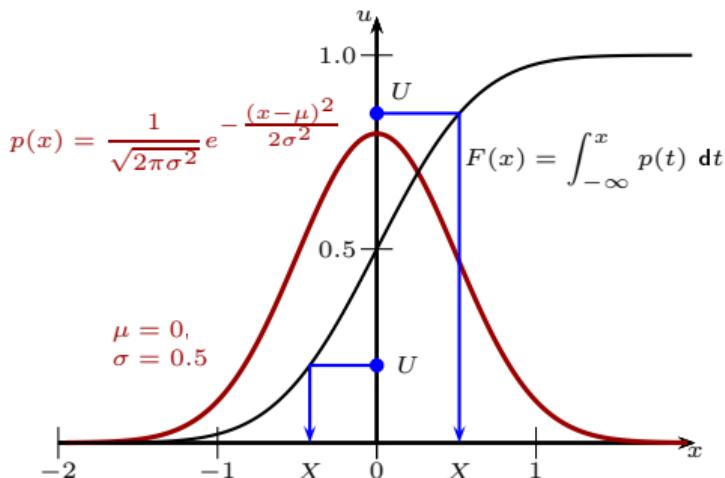
Interludium: Inverse transform sampling

Want¹: rv. X with density p .

1. Let $F(x) = \int_{-\infty}^x p(t) dt$ be the cumulative distribution function.
2. Pick $U \leftarrow [0, 1[$ uniformly.
3. Determine X such that $F(X) = U$.

Then

$$\begin{aligned}\text{prob } (a \leq X < b) \\ &= \text{prob } (F(a) \leq U < F(b)) \\ &= F(b) - F(a) = \int_a^b p(t) dt.\end{aligned}$$



Theorem

Inverse transform sampling yields a random variable X with density p .

¹ See Knuth (1969–1998). The Art of Computer Programming. Volume 2.

Leftovers

- ▶ What is the cost of the basic algorithm?
- ▶ How good can inverse transform sampling be when only a discrete approximation \tilde{U} can be used instead of U ?
- ▶ Which algorithm is used for solving $F(X) = U$?
- ▶ What is the cost with appropriate error control?
- ▶ How does that depend on the machine accuracy m ?

Method 2 for a random point on the surface

- ▶ Pick x_i according to a Gaussian normal distribution with mean 0 and variance 1, namely acc.to $\frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}}$.
- ▶ Return the normalized vector $\frac{1}{\|x\|_2}x$.

The probability density of x is

$$p(x) = \frac{1}{(2\pi)^{\frac{d}{2}}} e^{-\frac{1}{2}(x_0^2+x_1^2+\dots+x_{d-1}^2)},$$

which is spherically symmetric, ie. depends only on the radius $r = \|x\|_2$ but not on the angles. Thus the normalized vector is uniformly distributed over the surface of the sphere.

Gaussians in high dimensions:

Generating points uniformly at random from a ball

Application

Question

How to generate a random point x in a d -dimensional ball B^d ?

That is, pick $x \in \mathbb{R}^d$ with $\|x\|_2 \leq 1$.

Gaussians in high dimensions:

Generating points uniformly at random from a ball

Method for a uniformly random point in the ball

- ▶ Pick X_i according to a Gaussian normal distribution with mean 0 and variance 1, namely acc.to density $\frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}}$.
- ▶ Pick a radius $R \in [0, 1]$ with density dr^{d-1} .
- ▶ Return the vector $Y = R \frac{X}{\|X\|_2}$.

As the surface area scales with r^{d-1} we have to choose the density dr^{d-1} on $r \in [0, 1]$.
[We need it to scale with r^{d-1} and $\int_0^1 cr^{d-1} dr = 1$.]

Leftovers

- ▶ Derive the density of Y from those of X_i and R .
- ▶ It should be constant on B^d and 0 outside. (It is!)
- ▶ What about machine accuracy approximations?

Gaussians in high dimensions:

Gaussians

Definition

The d -dimensional spherical Gaussian distribution with mean $\mu \in \mathbb{R}^d$ and variance $\sigma^2 \in \mathbb{R}_{>0}$ in each coordinate has the density function

$$\mathcal{N}_d(\mu, \sigma^2)(x) = \frac{1}{(2\pi\sigma^2)^{\frac{d}{2}}} e^{-\frac{1}{2\sigma^2}\|x-\mu\|_2^2}.$$

Theorem

Let $X \xleftarrow{\text{IID}} \mathcal{N}_d(\mu, \sigma^2)$. Then

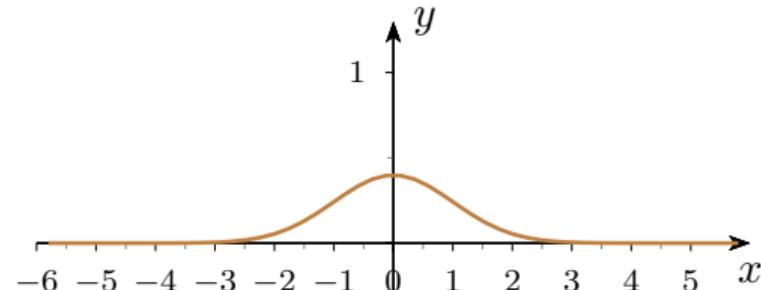
1. $E(X) = \mu$.
2. $X_i \sim \mathcal{N}(\mu_i, \sigma^2)$.
3. $\text{var}(X_i) = \sigma^2$.

Gaussians in high dimensions:

Gaussians

The Gaussian in low dimensions

The mass of a Gaussian in dimension 1, say,
is close to the origin.



The Gaussian in high dimensions (spoilers)

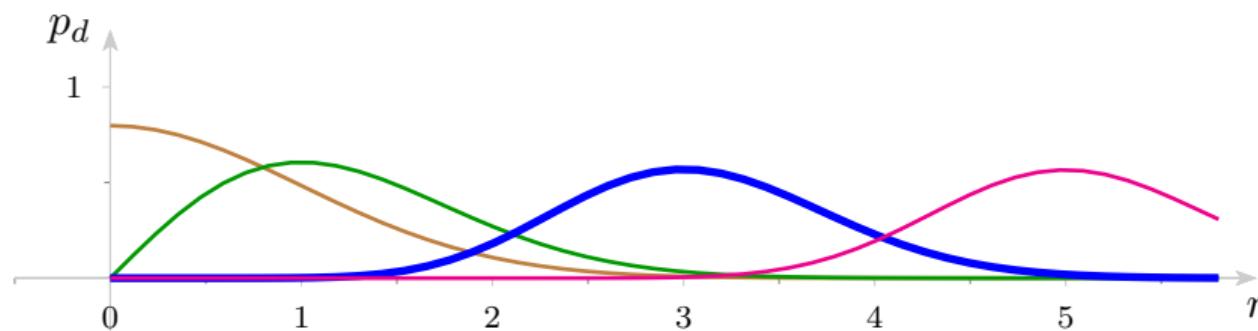
- ▶ The Gaussian density is maximal near the origin but there is almost no volume near 0 since the unit ball has such small volume.
⇒ almost no mass.
- ▶ Increasing the ball to radius nearly \sqrt{d} to obtain significant volume.
⇒ significant probability mass.
- ▶ Beyond radius \sqrt{d} density very low.
⇒ almost no further increase in mass.

Gaussians in high dimensions:

Gaussians

What is the density of $R = \|X\|_2$ if X is picked from a d -dimensional Gaussian $\mathcal{N}_d(0, 1)$?

Here is the answer $p_d(r) = \frac{2^{1-\frac{d}{2}}}{\Gamma(\frac{d}{2})} r^{d-1} e^{-\frac{1}{2}r^2}$ in dimensions 1, 2, 10, 26:



$$\text{Actually: } \operatorname{argmax}_r p_d(r) = \sqrt{d-1} \leq \mathbb{E}(R) = \frac{\sqrt{2}\Gamma(\frac{d+1}{2})}{\Gamma(\frac{d}{2})} \leq \sqrt{E(R^2)} = \sqrt{d}.$$

Intuition

Given $X \leftarrow \mathcal{N}_d(0, 1)$, that is $X \in \mathbb{R}^d$ with $X_i \leftarrow \mathcal{N}(0, 1)$. Then

$$\mathsf{E}(R^2) = \mathsf{E}(\|X\|_2^2) = \sum_i \underbrace{\mathsf{E}(X_i^2)}_{=\text{var } X_i=1} = d,$$

in other words:

Definition

The **radius of the Gaussian** is the square root of the expected squared radius, namely \sqrt{d} for $\mathcal{N}_d(0, 1)$.

Gaussians in high dimensions:

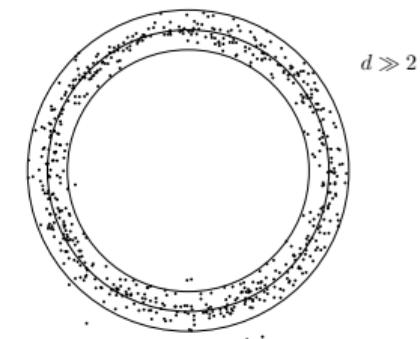
Gaussians

Theorem (Gaussian annulus theorem)

For a d -dimensional spherical Gaussian with unit variance in each direction, for any $\beta \leq \sqrt{d}$, we find for the probability to land in the Gaussian annulus $G^d(\beta) = \{x \in \mathbb{R}^d \mid \sqrt{d} - \beta \leq \|x\|_2 \leq \sqrt{d} + \beta\}$:

$$\text{prob}\left(X \in G^d(\beta)\right) \geq 1 - 3e^{-\frac{1}{48}\beta^2}$$

for $X \xleftarrow{\text{IID}} \mathcal{N}_d(0, 1)$.



$$\beta = \sqrt{\frac{d}{50}}, 3e^{-\frac{1}{48}\beta^2} = 3e^{-\frac{d}{2400}} \approx 1\% \text{ for } d = 13\,690.$$

Proof. . . .

□

Proof.

Consider $X = (X_0, \dots, X_{d-1})$ with $X_i \xrightarrow{\text{IID}} \mathcal{N}(0, 1)$. Define $R := \|X\|_2$.

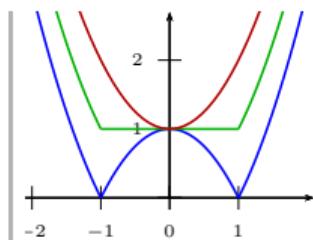
If $|R - \sqrt{d}| \geq \beta$ then also $|R^2 - d| \geq \beta(R + \sqrt{d}) \geq \beta\sqrt{d}$.
Thus

$$\text{prob}(|R - \sqrt{d}| \geq \beta) \leq \text{prob}(|R^2 - d| \geq \beta\sqrt{d}).$$

Now, $R^2 - d = \sum_{i < d} (X_i^2 - 1)$. Put $Y_i := X_i^2 - 1$, $W_i := \frac{1}{2}(X_i^2 - 1)$. Want $\text{prob}\left(\left|\sum_{i < n} W_i\right| \geq \frac{\beta}{2}\sqrt{d}\right)$. Have:

- $E(W_i) = \frac{1}{2}(E(X_i^2) - 1) = 0$.
- $E((2W_i)^s) \leq 1 + E(X_i^{2s})$.

Consider $y = x^2 - 1$. For $|x| \leq 1$ obviously $|y|^s \leq 1$. For $|x| \geq 1$ we have $|y|^s \leq |x|^{2s}$. Thus $|y|^s \leq 1 + |x|^{2s}$.



Next,

$$\begin{aligned} E(X_i^{2s}) &= \frac{1}{\sqrt{2\pi}} \cdot 2 \int_0^\infty x^{2s-1} e^{-\frac{x^2}{2}} x \, dx \\ &\stackrel{x^2=2z, x \, dx = dz}{=} 2^s \frac{1}{\sqrt{\pi}} \underbrace{\int_0^\infty z^{s-\frac{1}{2}} e^{-z} \, dz}_{=\Gamma(s+\frac{1}{2})=\frac{(2s-1)!!}{2^s}\sqrt{\pi}} \\ &= (2s-1)!! \leq (2s)!! - 1 = 2^s s! - 1. \end{aligned}$$

And so

$$E(W_i^s) \leq s!.$$

Now, apply the master tail bounds theorem with $\sigma^2 = 1$, $a = \frac{\beta}{2}\sqrt{d}$ and $n = d$ to obtain

$$\begin{aligned} \text{prob}(|R - \sqrt{d}| \geq \beta) &\leq \text{prob}\left(\left|\sum W_i\right| \geq \frac{\beta}{2}\sqrt{d}\right) \\ &\leq 3e^{-\frac{\beta^2}{48}}. \end{aligned}$$

□

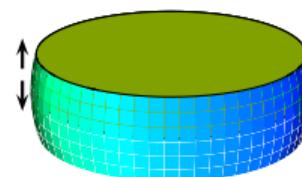
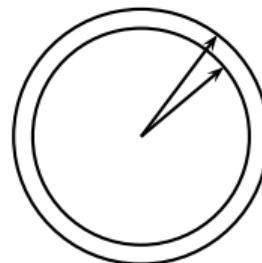
Summary

- ▶ A spherical Gaussian distribution helps picking a random point on the surface of a ball or in a ball.
- ▶ The expected radius of a random point with spherical Gaussian distribution is approximately \sqrt{d} .
- ▶ Most of the mass of a Gaussian is in an annulus with radii $\sqrt{d} + [-1, 1]\beta$, namely all but at most $3e^{-\frac{1}{48}\beta^2}$.

Gaussians in high dimensions:

Gaussians

High dimension results



ball B^d has most volume in	✓ annulus $A^d(\varepsilon)$, $e^{-\varepsilon}$	✓ tropical slice $T^d\left(\frac{c}{\sqrt{d-1}}\right)$, $\frac{2}{c}e^{-\frac{c^2}{2}}$
spherical Gaussian $\mathcal{N}_d(\mu, \sigma^2)$ has most mass in	✓ Gaussian annulus $G^d(\beta)$, $3e^{-\frac{1}{48}\beta^2}$	✓ Gaussian slice $S^d(\gamma)$, $e^{-\frac{1}{2}\gamma^2}$

Lemma (Gaussian slice)

Assume the random variable X has unit spherical Gaussian distribution $\mathcal{N}_d(0, 1)$. For the Gaussian slice $S^d(\gamma) = \{x \in \mathbb{R}^d \mid |x_0| \leq \gamma\}$ we then have

$$\text{prob}\left(X \in S^d(\gamma)\right) \geq 1 - e^{-\frac{1}{2}\gamma^2}.$$

Note $X_0 = \langle e_0 \mid X \rangle$. Thus by symmetry we have

$$\text{prob}(|\langle Y \mid X \rangle| \leq \gamma \|Y\|_2) \geq 1 - e^{-\frac{1}{2}\gamma^2}$$

for an independently random or fixed vector Y .

Proof. . .

□

Proof.

Since X has spherical unit Gaussian distribution $\mathcal{N}_d(0, 1)$, X_0 has unit Gaussian distribution $\mathcal{N}(0, 1)$. And so

$$\begin{aligned}\text{prob}(|X_0| > \gamma) &= 2 \int_{\gamma}^{\infty} \mathcal{N}(0, 1)(x_0) dx_0 \\ &= 2 \int_{\gamma}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{x_0^2}{2}} dx_0 \\ &< 1 \underbrace{\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{(x_0-0)^2}{2}} dx_0}_{=1} \cdot e^{-\frac{1}{2}\gamma^2} \\ &= e^{-\frac{1}{2}\gamma^2}.\end{aligned}$$

□

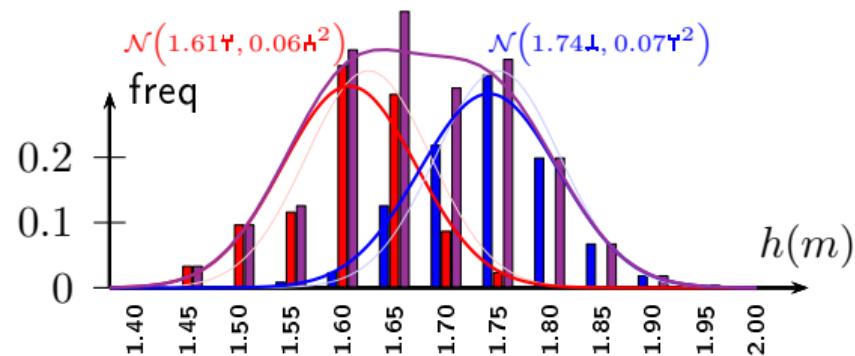
Remark: The cumulative normal distribution $\int_{-\infty}^{\gamma} \mathcal{N}(0, 1)(x_0) dx_0$ is closely related to the **error function** $\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-\tau^2} d\tau$.

Gaussians in high dimensions:

Fitting a spherical Gaussian to data

Motivation: height of individuals

Typically, men tend to be taller than women: $P(x) = w_f P_f(x) + w_m P_m(x)$ where P_i is a Gaussian density representing the typical heights of women or men, respectively, and w_f, w_m are the mixture weights representing the proportion of women and men.



Frequency of heights of fathers and mothers in data collected by Galton (1822-1911) to analyze correlations between parents and children.

Parameter estimation problem

What are good parameters?

In the figure: These? Or these? And in which sense? ... and how to separate them?

Gaussians in high dimensions:

Fitting a spherical Gaussian to data

This raises some questions:

Best approximation

Which Gaussian distribution —given by mean μ and variance σ^2 — is best suited to approximate a given random variable?

Best fit

How to fit when only experiments are possible, ie. multiple results of a random variable?
In other words, we are given data produced by multiply 'calling' a random variable.

Separation

Assume a random variable is an overlay of two or more Gaussian distributions. How to separate the different Gaussians?

Samples

People often talk about a *sample*; without further explanation.

Sometimes it is a mere set or list. More sophisticated is this:

- ▶ We assume that there is a random variable X with a given distribution.
 - ▶ In practice, the distribution is usually unknown.
 - ▶ Instead, we try to estimate it from a **sample**.
 - ▶ Usually, we assume that the distribution is from a certain class.
Eg. an overlay of two or more spherical Gaussian distributions.
- ▶ Taking a sample of size n now means:
 - ▶ Take n iid. random variables X_i , $i < n$, each with the distribution identical to X .
 - ▶ The sample \mathcal{X} is the set (or list or multi-set) of outcomes of these variables:

$$\mathcal{X} = \{X_i \mid i < n\}.$$

Technically, \mathcal{X} is again a random variable.

Gaussians in high dimensions:

Fitting a spherical Gaussian to data

Lemma

Given a finite set $\mathcal{X} \subset \mathbb{R}^d$ of n points. Then

$$\sum_{x \in \mathcal{X}} \|x - \mu\|_2^2$$

is minimized when μ is the **centroid** of the points, namely

$$\mu = \frac{1}{\#\mathcal{X}} \sum_{x \in \mathcal{X}} x.$$

Proof. . . .



Proof.

The gradient of $\sum_{i < n} \|x_i - \mu\|_2^2$ wrt. μ is

$$\sum_{i < n} -2(x_i - \mu)$$

which is zero at $\mu = \frac{1}{n} \sum_{i < n} x_i$.

□

Gaussians in high dimensions:

Fitting a spherical Gaussian to data

Task

Given a finite set $\mathcal{X} \subset \mathbb{R}^d$ of n points. Find the Gaussian $\mathcal{N}_d(\mu, \sigma^2)$ that maximizes the density

$$\prod_{x \in \mathcal{X}} \mathcal{N}_d(\mu, \sigma^2)(x) = (2\pi\sigma^2)^{-\frac{nd}{2}} \exp\left(-\frac{\sum_{x \in \mathcal{X}} \|x - \mu\|_2^2}{2\sigma^2}\right),$$

that is, in some sense, the probability for this particular sample \mathcal{X} to show up.

Gaussians in high dimensions:

Fitting a spherical Gaussian to data

Lemma

This maximum likelihood spherical Gaussian for a sample \mathcal{X} is the Gaussian with center equal to the sample mean μ and variance σ^2 equal to sample variance.

Note: The sample mean is given by $E(\mathcal{X}) = \frac{1}{n} \sum_{x \in \mathcal{X}} x$.

The sample variance is $\text{var}(\mathcal{X}) = \frac{1}{nd} \sum_{x \in \mathcal{X}} \|x - E(\mathcal{X})\|_2^2$.

(By abuse of notation we are overloading E and var here.)

Proof. . . .



Proof.

Let μ be the centroid as for any fixed σ this gives the minimal variance $v = \frac{1}{nd} \sum_{x \in \mathcal{X}} \|x - \mu\|_2^2$ of the data and so the maximal density

$$\frac{1}{(2\pi\sigma^2)^{\frac{nd}{2}}} \exp\left(-\frac{1}{2\sigma^2} ndv\right).$$

It remains to determine σ maximizing this. Substitute

$$\nu = \frac{1}{2\sigma^2}:$$

$$\pi^{-\frac{nd}{2}} \nu^{\frac{nd}{2}} \exp(-\nu ndv).$$

To find the maximum, take its logarithm and then set the derivative wrt. ν to zero:

$$\frac{nd}{2\nu} - ndv = 0.$$

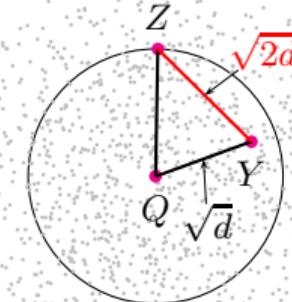
This yields $\sigma^2 = \frac{1}{2\nu} = v$ as the maximizing parameter, as claimed. □

Gaussians in high dimensions:

Separating Gaussians

How far apart must two d -dimensional Gaussians be to distinguish its points with large probability?

Heuristic: How far apart are two random points drawn from one unit Gaussians?



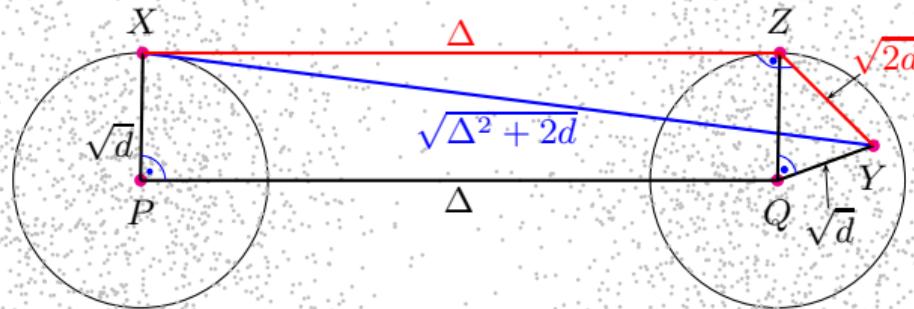
Pick $Y, Z \leftarrow \mathcal{N}_d(Q, 1)$.

Gaussians in high dimensions:

Separating Gaussians

How far apart must two d -dimensional Gaussians be to distinguish its points with large probability?

Heuristic: How far apart are two random points drawn from two unit Gaussians with distance Δ ?



Pick $Y, Z \xleftarrow{\text{IID}} \mathcal{N}_d(Q, 1)$ and put $X - P = Z - Q$ with $\|P - Q\|_2 = \Delta$.

Gaussians in high dimensions:

Separating Gaussians

How far apart must two d -dimensional Gaussians be to distinguish its points with large probability?

Heuristic: We need that two random points from the *different* Gaussians are further apart than two random points from the *same* Gaussian.

Given $X \xleftarrow{\text{IID}} \mathcal{N}_d(P, 1)$, $Y, Z \xleftarrow{\text{IID}} \mathcal{N}_d(Q, 1)$ with $\|P - Q\|_2 = \Delta$, we have

- ▶ $\|Y - Z\|_2 \in \sqrt{2d} \pm \mathcal{O}(1)$ whp.
- ▶ $\|X - Y\|_2 \in \sqrt{\Delta^2 + 2d} \pm \mathcal{O}(1)$ whp.

To ensure $\|X - Y\|_2 > \|Y - Z\|_2$ we thus

should have $\sqrt{2d} + \mathcal{O}(1) \leq \sqrt{\Delta^2 + 2d} - \mathcal{O}(1)$ or $2d + \mathcal{O}(\sqrt{d}) \leq \Delta^2 + 2d$. This is guaranteed by

$$\Delta \in \omega(d^{\frac{1}{4}}).$$

That's for a single pair. To separate n points whp., namely with probability $1 - \frac{1}{\text{poly}(n)}$, we need Δ by a factor $\sqrt{\log n}$ larger.

Tentative algorithm for separating points from two Gaussians

Given n points in \mathbb{R}^d from (a mix of) two Gaussians with distance at least $\omega\left(d^{\frac{1}{4}}\sqrt{\log n}\right)$.

Calculate all pairwise distances between points. The cluster of smallest pairwise distances must come from a single Gaussian. Remove these points. The remaining points come from the second Gaussian.

...

Section overview

Organizational

Introduction

High-dimensional space

Gaussians in high dimensions

Eigenvalues and eigenvectors

Dirac's bra-ket notation

Basics

Symmetric matrices

Extremal properties of eigenvalues

Eigenvalues of the sum of two symmetric
matrices

Norms

Additional linear algebra

Best-fit subspaces and SVD

Power method for SVD

Applications of SVD

Machine learning

*Clustering

Summary / Outro

Dirac's bra-ket notation

- ▶ A **ket** is a column vector

$$|y\rangle \in \mathbb{R}^n.$$

- ▶ A **bra** is a row vector or the transpose of a vector

$$\langle x| \in (\mathbb{R}^m)^\vee.$$

- ▶ Multiplying a bra with a ket only works if $m = n$ and this is a **scalar product**:

$$\langle x | y \rangle = \langle x | | y \rangle = \sum_{i < n} x_i y_i \in \mathbb{R}.$$

- ▶ Multiplying a ket with a bra returns a **rank-1 matrix**:

$$|y\rangle \langle x| = [y_i x_j]_{i < m, j < n} \in \mathbb{R}^{m \times n}.$$

Dirac's bra-ket notation

- ▶ Applying a matrix $A = \sum_{i,j} A_{ij} |e_i\rangle \langle e_j|$ to a vector $|x\rangle = \sum_k x_k |e_k\rangle$ now looks as follows:

$$\begin{aligned} A|x\rangle &= \sum_{i,j} A_{ij} |e_i\rangle \underbrace{\langle e_j | x\rangle}_{=x_j} \\ &= \sum_i \left(\sum_j A_{ij} x_j \right) |e_i\rangle. \end{aligned}$$

- ▶ **Side effect:** Assume that $\{|v_0\rangle, \dots, |v_{n-1}\rangle\}$ is an orthogonal basis. Then

$$\sum_{i < n} |v_i\rangle \langle v_i| = \mathbb{1}.$$

Eigenvalues and eigenvectors:

Basics

Consider a real $n \times n$ matrix $A \in \mathbb{R}^{n \times n}$.

Definition (Eigenvalue and eigenvector)

The scalar α is an **eigenvalue of A** if there is a *non-zero* vector $|x\rangle$ with

$$A \cdot |x\rangle = \alpha |x\rangle.$$

Such a vector $|x\rangle$ is called an **eigenvector of A** associated with α .

All eigenvectors associated with a given α form a subspace, the **eigenspace of α** :

$$\ker(A - \alpha \mathbb{1}) = \{|x\rangle \in \mathbb{R}^n \mid A|x\rangle = \alpha|x\rangle\}.$$

Note: The equation $A|x\rangle = \alpha|x\rangle$ has a non-zero solution iff
the **characteristic equation** $\det(A - \alpha \mathbb{1}) = 0$ is fulfilled.

Matrices A and B are **similar** if there is an invertible matrix P with $A = P^{-1}BP$.

Theorem

If A and B are similar then they have the same eigenvalues.



Theorem

A is diagonalizable, ie. similar to a diagonal matrix,

iff

A has n linearly independent eigenvectors.



Eigenvalues and eigenvectors:

Symmetric matrices

In general, eigenvalues may be complex even though the matrix has only real entries, eg. the eigenvalues of the rotation through an angle φ

$$\begin{bmatrix} \cos \varphi & -\sin \varphi \\ \sin \varphi & \cos \varphi \end{bmatrix}$$

are $e^{\pm i\varphi}$.

Eigenvalues and eigenvectors:

Symmetric matrices

Symmetric matrices are nicer: they are always diagonalizable. Even more:

A symmetric matrix is always **orthogonally diagonalizable**, ie. there is an **orthogonal** matrix P , namely with $P^T P = \mathbb{1}$, such that

$$D = P^T A P$$

is diagonal.

Example

Consider $A = \begin{bmatrix} 1 & 2 & 0 \\ 2 & 4 & 0 \\ 0 & 0 & 0 \end{bmatrix}$.

It is symmetric.

- ▶ Clearly, $|x^{(0)}\rangle := |e_2\rangle$ is an eigenvector with eigenvalue 0: $A|e_2\rangle = 0$.
- ▶ Further, $\sqrt{5}|x^{(1)}\rangle := |-2e_0 + e_1\rangle$ is another eigenvector with eigenvalue 0.
- ▶ And $\sqrt{5}|x^{(2)}\rangle := |e_0 + 2e_1\rangle$ is an eigenvector with eigenvalue 5.

Normalize them and put them in a matrix

$$Q = \begin{bmatrix} -\frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}} \\ \frac{1}{\sqrt{5}} & \frac{2}{\sqrt{5}} \\ 1 & 0 \end{bmatrix}.$$

Then $Q^T Q = \mathbb{1}$, $Q^{-1} = Q^T$ and

$$Q^{-1}AQ = \begin{bmatrix} 0 & & \\ & 0 & \\ & & 5 \end{bmatrix}.$$

Theorem (Real spectral theorem)

Let $A \in \mathbb{R}^{n \times n}$ be a real symmetric matrix. Then

1. The matrix has n real eigenvalues $\alpha_0, \alpha_1, \dots, \alpha_{n-1}$ and corresponding linearly independent real eigenvectors $|v_0\rangle, |v_1\rangle, \dots, |v_{n-1}\rangle$.
2. *Spectral Decomposition*: A is orthogonally diagonalizable and indeed

$$A = VDV^T = \sum_{i < n} \alpha_i |v_i\rangle \langle v_i|$$

where V is the orthogonal matrix with columns $|v_0\rangle, |v_1\rangle, \dots, |v_{n-1}\rangle$, $\|v_i\|_2 = 1$ and D is a diagonal matrix with entries $\alpha_0, \alpha_1, \dots, \alpha_{n-1}$.

Proof. . .



Proof.

It suffices to show that $A = VDV^T$ for some orthonormal matrix V . Namely, then $AV = VD$, the diagonal entries of D are the eigenvalues $\alpha_0, \dots, \alpha_{n-1}$, the columns of V are the corresponding eigenvectors.

We proceed by induction on the dimension.

Dimension $n = 0$ or $n = 1$: Nothing to do.

Dimension $n > 1$: Take any eigenvalue α , namely any root of the polynomial $\det(A - \alpha\mathbb{1})$.

This eigenvalue α must be real. [With $A^* := \overline{A}^T$, here $A^* = \overline{A} = A$. Notice that $(AB)^* = B^*A^*$, $|v\rangle^* = \langle v|$, $\langle v|^* = |v\rangle$ and for a number $\alpha^* = \overline{\alpha}$. So $\overline{\langle v|A|v\rangle} = \langle v|A^*|v\rangle = \langle v|A^*|v\rangle$. With $A^* = A$ then $\overline{\langle v|A|v\rangle} = \langle v|A|v\rangle$. Since for any eigenvector $|v\rangle$ with this eigenvalue $\langle v|A|v\rangle = \alpha\langle v|v\rangle$, this implies $\overline{\alpha} = \alpha$ and α is real as claimed.]

Since α is real, there exists a non-trivial solution to $(A - \alpha\mathbb{1})|v\rangle = 0$ in \mathbb{R}^n .

Take an orthonormal basis $[|v\rangle, |w_1\rangle, \dots, |w_{n-1}\rangle]$. Define $P = [|v\rangle, |w_1\rangle, \dots, |w_{n-1}\rangle]$. Then P is orthonormal, ie.

$$P^{-1} = P^T, \text{ and } P|e_0\rangle = |v\rangle.$$

Thus $P^TAP|e_0\rangle = P^TA|v\rangle = \alpha P^{-1}|v\rangle = \alpha|e_0\rangle$. Since P^TAP is symmetric and its first column is $\alpha|e_0\rangle$, its first row is $\alpha\langle e_0|$. Thus P^TAP maps the space $\text{span}\{|e_1\rangle, \dots, |e_{n-1}\rangle\}$ into itself. Together,

$$P^TAP = \left[\begin{array}{c|c} \alpha & \xrightarrow{\quad} \\ \hline | & A' \end{array} \right] \quad \text{or} \quad A = P \begin{bmatrix} \alpha & \\ & A' \end{bmatrix} P^T$$

for a symmetric matrix $A' \in \mathbb{R}^{(n-1) \times (n-1)}$.

By induction hypothesis $A' = QD'Q^T$ for some orthonormal matrix $Q \in \mathbb{R}^{(n-1) \times (n-1)}$ and diagonal matrix D' and so:

$$A = P \underbrace{\begin{bmatrix} 1 & \\ & Q \end{bmatrix}}_{=:V} \underbrace{\begin{bmatrix} \alpha & \\ & D' \end{bmatrix}}_{=:D} \underbrace{\begin{bmatrix} 1 & \\ & Q^T \end{bmatrix}}_{=:V^T} P^T$$

and so $A = VDV^T$. □



Theorem (Fundamental theorem of symmetric matrices)

A real matrix A is orthogonally diagonalizable iff A is symmetric.

Proof. . . .



Proof.

\Rightarrow Assume A is orthogonally diaonalizable. That is, $A = PDP^{-1}$ with $P^{-1} = P^T$ and D diagonal.
And so $A^T = (PDP^T)^T = PDP^T = A$, ie. A is symmetric.

\Leftarrow Previous theorem. □

Eigenvalues and eigenvectors:

Extremal properties of eigenvalues

Now, that we know that a symmetric matrix A is always diagonalizable with real eigenvalues, sort its eigenvalues in decreasing order

$$\alpha_0 \geq \alpha_1 \geq \cdots \geq \alpha_{n-1}.$$

Now,

$$A = P \begin{bmatrix} \alpha_0 & & \\ & \ddots & \\ & & \alpha_{n-1} \end{bmatrix} P^T$$

where P is an orthogonal matrix whose columns are an orthonormal basis consisting of eigenvectors.

Theorem (Min max theorem)

For a symmetric matrix $A \in \mathbb{R}^{n \times n}$ we have

$$\alpha_s = \min_{\substack{R \subset \mathbb{R}^n, \\ \#R=s}} \max_{\substack{|x\rangle \in \mathbb{R}^n, \\ |x\rangle \perp R}} \langle x | A | x \rangle,$$

where the minimum is over all sets $R = \{|r_0\rangle, \dots, |r_{s-1}\rangle\}$ of s (non-zero) vectors and the maximum is over all unit vectors $|x\rangle$ orthogonal to the s non-zero vectors.

Let's abbreviate for $R = \{|r_0\rangle, \dots, |r_{s-1}\rangle\}$:

$$|x\rangle \perp R \iff \| |x\rangle \|_2 = 1 \wedge \forall i < s: |x\rangle \perp |r_i\rangle.$$

Proof. . .

□

Theorem (Min max theorem, case $s = 0$)

For a symmetric matrix $A \in \mathbb{R}^{n \times n}$ we have

$$\alpha_0 = \min_{R=\{\}} \max_{\| |x\rangle \|_2=1} \langle x | A | x \rangle.$$

Note that $\langle x | A | x \rangle$ measures which fraction of the image vector $A | x \rangle$ points in direction $| x \rangle$.
The theorem now says:

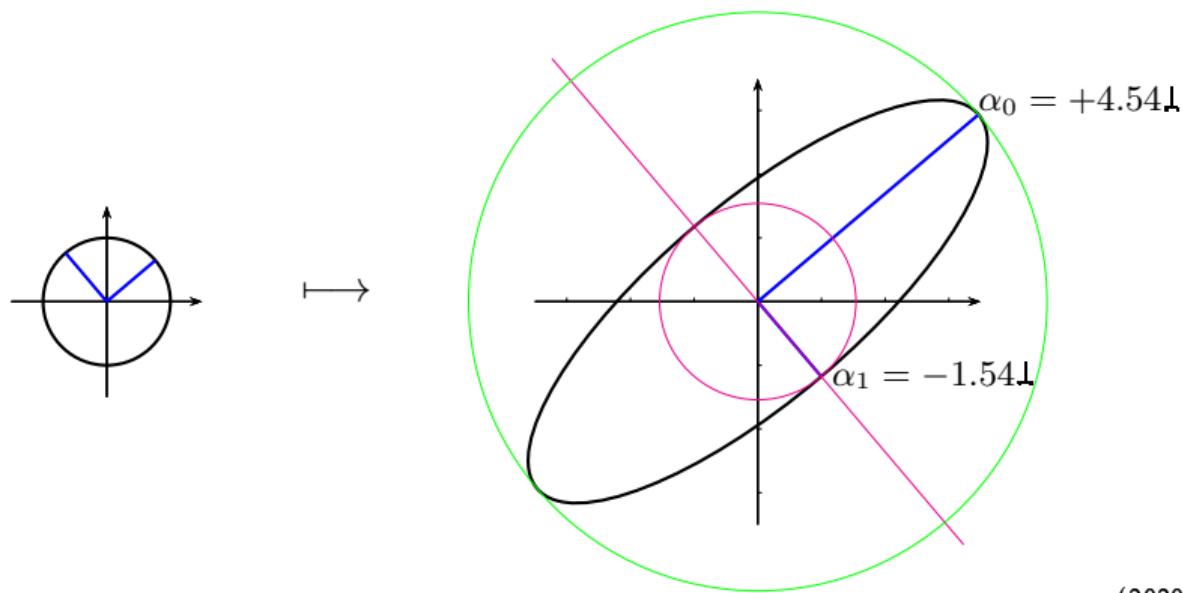
- ▶ The largest eigenvalue is equal to $\langle x | A | x \rangle$ for some vector $| x \rangle$.
- ▶ All other values $\langle x | A | x \rangle$ are smaller or equal.

Eigenvalues and eigenvectors:

Extremal properties of eigenvalues

Example: How does a matrix stretch vectors?

Take $A = \begin{bmatrix} 1 & 3 \\ 3 & 2 \end{bmatrix}$. It maps its eigenvectors and the unit circle like this:



Eigenvalues and eigenvectors:

Extremal properties of eigenvalues

Theorem (Min max theorem, case $s = 1$)

For a symmetric matrix $A \in \mathbb{R}^{n \times n}$ we have

$$\alpha_1 = \min_{|r_0\rangle \in \mathbb{R}^n} \max_{\substack{|x\rangle \in \mathbb{R}^n, \\ |x\rangle \perp \{ |r_0\rangle \}}} \langle x | A | x \rangle.$$

Note that $\langle x | A | x \rangle$ measures which fraction of the image vector $A |x\rangle$ points in direction $|x\rangle$. The theorem now says that there is a vector $|r_0\rangle$ making this an equality. Take $|r_0\rangle$ as an eigenvector with eigenvalue α_0 .

- ▶ The second largest eigenvalue is equal to $\langle x | A | x \rangle$ for some vector $|x\rangle \perp |r_0\rangle$.
- ▶ All other values $\langle x | A | x \rangle$ with $|x\rangle \perp |r_0\rangle$ are smaller or equal.

Proof (Min max theorem).

Consider A wrt. the eigenbasis with decreasing eigenvalues $\alpha_0 \geq \alpha_1 \geq \dots \geq \alpha_{n-1}$. Thus A is diagonal,

$$A = \begin{bmatrix} \alpha_0 & & & \\ & \alpha_1 & & \\ & & \ddots & \\ & & & \alpha_{n-1} \end{bmatrix}.$$

If $|x\rangle = \sum_{i < n} \xi_i |e_i\rangle$ is a unit vector then

$$\langle x| A |x\rangle = \sum_{i < n} \alpha_i \xi_i^2 \quad \text{and} \quad \sum_i \xi_i^2 = 1.$$

We want to minimize

$$\max_{|x\rangle \perp R} \langle x| A |x\rangle$$

where $|x\rangle$ runs over all unit vectors orthogonal to a given set R of s vectors $|r_i\rangle$, $i < s$.

\geq : Choosing $R = \{|e_i\rangle \mid i < s\}$ forces $\xi_i = 0$ for $i < s$. Among the remaining vectors $|x\rangle = |e_s\rangle$ maximizes $\langle x| A |x\rangle = \alpha_s$, and so $\alpha_s = \max_{|x\rangle \perp R} \langle x| A |x\rangle \geq \min_R \max_{|x\rangle \perp R} \langle x| A |x\rangle$.

\leq : Fix some set $R = \{|r_i\rangle \mid i < s\}$. You may assume $\text{span}\{|r_i\rangle \mid i < s\} \neq \text{span}\{|e_i\rangle \mid i < s\}$ as other choices are in the first case, but we do not need that. We are going to show that $\alpha_s \leq \max_{|x\rangle \perp R} \langle x| A |x\rangle$.

Extend R by $|r_i\rangle = |e_{i+1}\rangle$ for $s \leq i < n-1$. There still exists a non-trivial vector $|x\rangle$ fulfilling the $n-1$ linear conditions $|r_i\rangle \cdot |x\rangle = 0$ for $i < n-1$. Normalize it: $\| |x\rangle \| = 1$. By the added conditions we have $\xi_i = 0$ for $i > s$. Then

$$\langle x| A |x\rangle = \sum_{i \leq s} \alpha_i \xi_i^2 \geq \alpha_s \sum_{i \leq s} \xi_i^2 = \alpha_s.$$

So $\max_{|x\rangle \perp R} \langle x| A |x\rangle \geq \alpha_s$ and that's it. □

Eigenvalues and eigenvectors:

Eigenvalues of the sum of two symmetric matrices

Theorem

Let A and B be $n \times n$ symmetric matrices.

Let $C = A + B$. Let α_i , β_i and γ_i denote the eigenvalues of A , B and C respectively, where $\alpha_0 \geq \alpha_1 \geq \dots \geq \alpha_{n-1}$ and similarly for β_i , γ_i . Then

$$\alpha_s + \beta_{n-1} \leq \gamma_s \leq \alpha_s + \beta_0.$$

Lemma

Let A and B be $n \times n$ symmetric matrices.

Let $C = A + B$. Let α_i , β_i and γ_i denote the eigenvalues of A , B and C respectively, as above. Then

$$\gamma_{s+t} \leq \alpha_s + \beta_t.$$

Proof. . .



Proof. . .



Actually, the lemma implies the theorem.

Proof (Theorem).

Choose a list $R = \{|r_i\rangle \mid i < s\}$ minimizing $\max_{|x\rangle \perp R} \langle x| A |x\rangle$, ie. $\alpha_s = \max_{|x\rangle \perp R} \langle x| A |x\rangle$. By the min-max theorem we can estimate

$$\begin{aligned}\gamma_s &\leq \max_{|x\rangle \perp R} \langle x| (A + B) |x\rangle \\ &\leq \underbrace{\max_{|x\rangle \perp R} \langle x| A |x\rangle}_{=\alpha_s} + \underbrace{\max_{|x\rangle \perp \{x\}} \langle x| B |x\rangle}_{=\beta_0} = \alpha_s + \beta_0.\end{aligned}$$

Using the same estimate for $A = C + (-B)$ yields $\alpha_s \leq \gamma_s + (-\beta_{n-1})$ which is the other inequality. □

Proof (Lemma).

Choose R with r vectors minimizing $\max_{|x\rangle \perp R} \langle x| A |x\rangle$ and thus $\max_{|x\rangle \perp R} \langle x| A |x\rangle = \alpha_r$.

Choose S with s vectors minimizing $\max_{|x\rangle \perp S} \langle x| B |x\rangle$ and thus $\max_{|x\rangle \perp S} \langle x| B |x\rangle = \beta_s$.

Now, consider T versus $R \cup S$. Then

$$\begin{aligned}\gamma_{r+s} &= \min_{T \in \binom{\mathbb{R}^d}{r+s}} \max_{|x\rangle \perp T} \langle x| C |x\rangle \\ &\leq \max_{|x\rangle \perp R \cup S} \langle x| C |x\rangle \\ &\leq \max_{|x\rangle \perp R \cup S} \langle x| A |x\rangle + \max_{|x\rangle \perp R \cup S} \langle x| B |x\rangle \\ &\leq \max_{|x\rangle \perp R} \langle x| A |x\rangle + \max_{|x\rangle \perp S} \langle x| B |x\rangle \\ &= \alpha_r + \beta_s.\end{aligned}$$

□

Definition

A **norm** on \mathbb{R}^n is a function $\|\cdot\| : \mathbb{R}^n \rightarrow \mathbb{R}$ satisfying

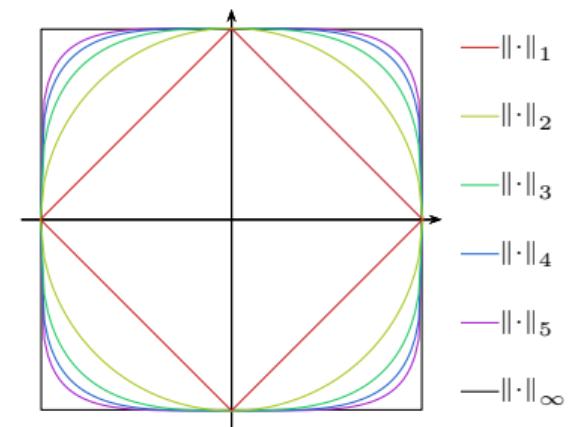
1. $\|x\| \geq 0$ with equality only for $|x\rangle = 0$,
2. $\|\alpha |x\rangle\| = |\alpha| \|x\|$, and
3. triangle inequality: $\|x\rangle + |y\rangle\| \leq \|x\rangle\| + \|y\rangle\|$.

A norm always provides a **distance** function

$$\text{dist}(|x\rangle, |y\rangle) := \|x\rangle - |y\rangle\|.$$

Examples

- ▶ p -norm: $\|x\|_p := (\sum_i |x_i|^p)^{\frac{1}{p}}$.
- ▶ 1-norm: $\|x\|_1 = |x_1| + \dots + |x_n|$,
- ▶ 2-norm: $\|x\|_2 = \sqrt{|x_1|^2 + \dots + |x_n|^2}$,
- ▶ ∞ -norm: $\|x\|_\infty := \max \{|x_1|, \dots, |x_n|\}$.
- ▶ Not a norm: “0-norm”:
 $\|x\|_0 := \#\{i \mid x_i \neq 0\}$.



Lemma

For any $1 \leq p < q$ we have $\|x\|_p \geq \|x\|_q$.

Matrix norms

For matrices we usually use the corresponding **operator p -norm** or **p -matrix-norm**

$$\|A\|_p := \max_{\|x\|_p=1} \|A|x\rangle\|_p = \max_{|x\rangle \neq 0} \frac{\|A|x\rangle\|_p}{\|x\|_p}.$$

and the **Frobenius norm**

$$\|A\|_F := \sqrt{\sum_{i,j} A_{i,j}^2}.$$

Corollary

The p -matrix-norm does not depend on the chosen basis, ie. $\|B^{-1}AB\|_p = \|A\|_p$. □

Lemma

Consider A orthogonally diagonalizable with eigenvalues α_i , $i < n$. Then:

1. $\|A\|_2 = \max_{i < n} |\alpha_i|$.
2. $\|A\|_F^2 = \text{trace}(A^T A) = \text{trace}(AA^T)$.
3. $\|A\|_F^2 = \sum_{i < n} \alpha_i^2$.

Proof. . . .



Corollary

The Frobenius norm does not depend on the chosen orthogonal basis:

$$\|Q^{-1}AQ\|_F = \|A\|_F \text{ for } Q \text{ orthogonal.}$$



Proof.

$\|A\|_2 = \max_{i < n} |\alpha_i|$: Suppose $V = (|v_0\rangle, \dots, |v_{n-1}\rangle)$ is an orthonormal basis of eigenvectors of A . Consider a unit vector $|x\rangle$, ie. with $\| |x\rangle \|_2 = 1$, and write $|x\rangle = \sum_{i < n} \xi_i |v_i\rangle$. Then

$$A|x\rangle = \sum_{i < n} \alpha_i \xi_i |v_i\rangle \quad \text{and} \quad 1 = \| |x\rangle \|_2^2 = \sum_{i < n} \xi_i^2.$$

And as the eigenvectors are orthonormal then

$$\begin{aligned} \|A|x\rangle\|_2^2 &= \sum_{i < n} |\alpha_i|^2 |\xi_i|^2 \\ &\leq \max \left\{ |\alpha_i|^2 \mid i < n \right\} \cdot \overbrace{\sum_{i < n} \xi_i^2}^{=1} \\ &= \max \{ |\alpha_i| \mid i < n \}^2. \end{aligned}$$

Wlog. $|\alpha_0| = \max \{ |\alpha_i| \mid i < n \}$. Then this holds with equality for $\xi_0 = 1$ and $\xi_i = 0$ for $0 < i < n$.

$\|A\|_F^2 = \text{trace}(A^T A) = \text{trace}(AA^T)$: Simply expand:

$$\begin{aligned} \|A\|_F^2 &= \sum_{i,j < n} A_{ij}^2 = \sum_{j < n} (|A_{\cdot,j}\rangle)^T |A_{\cdot,j}\rangle \\ &= \text{trace}(A^T A). \end{aligned}$$

As $\|A^T\|_F = \|A\|_F$ this gives also the other equality. Or you repeat the previous on rows.

$\|A\|_F^2 = \sum_{i < n} \alpha_i^2$: The last statement follows by writing A wrt. an orthogonal eigenbasis. And the matrix then is a diagonal matrix with the eigenvalues α_i on its diagonal. Now, the statement is immediate from the previous. \square

Lemma (Relations)

If A is symmetric and rank r then

$$\|A\|_2^2 \leq \|A\|_F^2 \leq r \|A\|_2^2.$$

Proof. Exercise. □

Lemma

- ▶ $\|AB\|_2 \leq \|A\|_2 \|B\|_2.$
- ▶ If Q is orthonormal then for each $|x\rangle$ we have $\|Q|x\rangle\|_2 = \||x\rangle\|_2$ and so $\|Q\|_2 = 1.$
- ▶ $\|QA\|_2 = \|A\|_2.$

Lemma

- ▶ $\|AB\|_F \leq \|A\|_F \|B\|_F.$
- ▶ If Q is orthonormal then $\|Q\|_F = \sqrt{n}.$
- ▶ $\|QA\|_F = \|A\|_F.$

Proofs. . .



Proof.

$\|AB\|_2 \leq \|A\|_2 \|B\|_2$: For any operator norm, ie. $\|f\| = \max_{\|x\| \neq 0} \frac{\|f(x)\|}{\|x\|}$, we have $\|AB\| \leq \|A\| \|B\|$. From the definition we infer that

$$\|A|x\rangle\| \leq \|A\| \|x\|.$$

Thus $\|AB|x\rangle\| \leq \|A\| \|B\| \|x\|$. And the other side of the definition gives: $\|AB\| = \max \left\| AB \frac{|x\rangle}{\|x\|} \right\| \leq \|A\| \cdot \|B\|$.

$\|Q\|_2 = 1$: If Q is orthonormal, we have $Q^T Q = \mathbb{1}$ and so

$$\|Q|x\rangle\|_2^2 = \langle x | Q^T Q | x \rangle = \langle x | x \rangle = \|x\|_2^2$$

and thus $\|Q\|_2 = 1$.

$\|QA\|_2 = \|A\|_2$: With the just proven inequality we obtain $\|QA\|_2 \leq \|Q\|_2 \|A\|_2 = \|A\|_2$. Similarly, $\|Q^{-1}QA\|_2 \leq \|Q^{-1}\|_2 \|QA\|_2$. Together the desired equality follows. \square

As the Frobenius norm is **not** an operator norm, we have

to prove the Frobenius lemma separately.

Proof.

$\|AB\|_F \leq \|A\|_F \|B\|_F$: Consider $C = AB$ and note that each entry $C_{ij} = A_{i,\cdot} B_{\cdot,j}$ is the product of a row of A and a column of B . The Cauchy-Schwartz inequality tells us that $|C_{ij}| \leq \|A_{i,\cdot}\| \|B_{\cdot,j}\|$. Thus

$$\begin{aligned} \|AB\|_F^2 &= \sum_{i,j} |C_{ij}|^2 \leq \sum_i \sum_j \|A_{i,\cdot}\|^2 \|B_{\cdot,j}\|^2 \\ &= \sum_i \|A_{i,\cdot}\|^2 \cdot \sum_j \|B_{\cdot,j}\|^2 \\ &= \|A\|_F^2 \cdot \|B\|_F^2. \end{aligned}$$

$\|Q\|_F = \sqrt{n}$: Next, consider Q orthonormal. Clearly, $\|Q\|_F = \sqrt{\text{trace}(Q^T Q)} = \sqrt{\text{trace}(\mathbb{1})} = \sqrt{n}$.

$\|QA\|_F = \|A\|_F$: Since $\|Q\|_F > 1$ we cannot repeat the trick from above. Instead, we use the trace description:

$$\|QA\|_F^2 = \text{trace}(A^T Q^T Q A) = \text{trace}(A^T A) = \|A\|_F^2. \quad \square$$

Lemma

Let A be symmetric. Then $\|A\|_2 = \max_{\|x\|_2=1} |\langle x | A | x \rangle|.$

Lemma

For each column $|A_{\cdot,j}\rangle$ of A we have $\|A_{\cdot,j}\|_2 \leq \|A\|_2$.

Proofs. Exercise.



Lemma

Let A be a symmetric $n \times n$ -matrix. Then

1. $\det(A) = \prod_{i < n} \alpha_i = \alpha_0 \alpha_1 \dots \alpha_{n-1}$.
2. $\text{trace}(A) = \sum_{i < n} \alpha_i = \alpha_0 + \alpha_1 + \dots + \alpha_{n-1}$.

Proof. . . .



Proof.

Just notice that \det does not depend on the chosen basis, so $\det(PAP^{-1}) = \det(A)$ and thus we can assume that $A = \text{diag}(\alpha_0, \dots, \alpha_{n-1})$. But for such a simple matrix the lemma obviously holds.

Same for the trace. □

Alternatively, the lemma follows recalling that

$$\begin{aligned}\det(A - \lambda \mathbb{1}) &= \prod_{i < n} (\alpha_i - \lambda) \\ &= (-1)^n \lambda^n + (-1)^{n-1} \text{trace}(A) \lambda^{n-1} + \cdots + \det(A) \lambda^0.\end{aligned}$$

Traces and rectangular matrices

Lemma

Let A be an $n \times m$ -matrix and B an $m \times n$ -matrix. Then $\text{trace}(AB) = \text{trace}(BA)$.

Proof. . . .



Proof.

Compute both and compare:

$$\text{trace}(AB) = \sum_{i < n} \sum_{j < m} A_{ij}B_{ji},$$

$$\text{trace}(BA) = \sum_{j < m} \sum_{i < n} B_{ji}A_{ij}.$$



A symmetric matrix is **positive semidefinite** iff for all $|x\rangle$ we have

$$\langle x| A |x\rangle \geq 0.$$

Theorem

The following are equivalent

1. *A is positive semidefinite.*
2. *All eigenvalues of A are non-negative.*
3. *$A = R^T R$ for some square matrix R.*

Proof. . . .



Actually, if A is **positive definite**, ie. positive semidefinite and $\langle x| A |x\rangle = 0$ only if $|x\rangle = 0$, then $\langle x| y\rangle_A := \langle x| A |y\rangle$ is a scalar product. And $\| |x\rangle \|_A := \sqrt{\langle x| x\rangle_A}$ is a norm.

Proof.

1 \Rightarrow 2: Given that A is positive semidefinite we have to show that all eigenvalues of A are non-negative.

Say $A|x\rangle = \lambda|x\rangle$ with $|x\rangle \neq 0$. Then $0 \leq \langle x|A|x\rangle = \lambda\langle x|x\rangle = \lambda\|x\|^2$ and so $\lambda \geq 0$.

2 \Rightarrow 3: Given that all eigenvalues of A are non-negative we have to show that $A = R^T R$ for some square matrix R .

As A is symmetric, we can write $A = VDV^T$ with an orthogonal matrix V and a diagonal matrix D whose diagonal entries are the eigenvalues. As those are all non-negative we can write them as squares and so $D = D^{\frac{1}{2}}D^{\frac{1}{2}}$ which implies

$$A = VDV^T = VD^{\frac{1}{2}} \underbrace{D^{\frac{1}{2}}V^T}_{=:R}$$

so that now $A = R^T R$.

3 \Rightarrow 1: Given that $A = R^T R$ for some square matrix R we have to show that A is positive semidefinite.

Write $A = R^T R$ and take any $|x\rangle$. Then $\langle x|A|x\rangle = \langle x|R^T R|x\rangle = \|R|x\rangle\|_2^2 \geq 0$.

□

Summary: Eigenfeatures of symmetric matrices.

- ▶ Eigenvalues and eigenvectors.
- ▶ Symmetric matrices are orthogonally diagonalizable and vice versa.
- ▶ Min max theorem: characterize s -th largest eigenvalue of a symmetric matrix.
- ▶ Eigenvalues of $A + B$ for symmetric matrices.
- ▶ Norms and matrix norms.
- ▶ ... in terms of eigenvalues.
- ▶ Positive semidefinite matrices, scalar products and norms.

Section overview

Organizational

Introduction

High-dimensional space

Gaussians in high dimensions

Eigenvalues and eigenvectors

Best-fit subspaces and SVD

Introduction

Singular vectors

Singular value decomposition (SVD)

Best rank- k approximation

An application: approximating $A|x\rangle\dots$

Left singular vectors

Power method for SVD

Applications of SVD

Machine learning

*Clustering

Summary / Outro

Best-fit subspaces and SVD:

Introduction

- ▶ Consider a list of n data point in \mathbb{R}^d .

$$\begin{bmatrix} & \\ \bullet & \bullet & \bullet & \bullet \\ & \end{bmatrix}$$

- ▶ Put each as a row into an $n \times d$ -matrix $A =$

- ▶ Now, we may ask: Which k out of the d coordinates are the '**most important**' ones, ie. give the best approximation to the data.

More general:

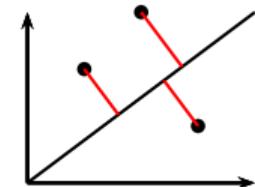
Question

*What is the **best-fitting** k -subspace (ie. k -dimensional subspace)?*

Best-fit subspaces and SVD:

Introduction

Precisely, by **best-fitting** we mean this: Project each data point to the chosen subspace and measure the loss by the distance, then we try to minimize the sum of the squares of those distances:



$$\text{minimize} \sum \| \langle A_{i,\cdot} | - \pi(\langle A_{i,\cdot} |) \|_2^2.$$

Equivalently, we maximize the sum of squares of the projections:

$$\text{maximize} \sum \| \pi(\langle A_{i,\cdot} |) \|_2^2.$$

We will show that this is connected to the **singular value decomposition** (SVD) of A .

Definition (Orthogonal projection)

Given a subspace $\mathcal{V} \subset \mathbb{R}^d$. Then for a vector $|a\rangle \in \mathbb{R}^d$ we define its **orthogonal projection** $\pi(|a\rangle) \in \mathbb{R}^d$ onto \mathcal{V} as the **only** vector $|p\rangle \in \mathcal{V}$ with

$$\langle v | a - p \rangle = 0 \text{ for all } |v\rangle \in \mathcal{V}.$$

Remark: Note that $\langle u | v \rangle = \langle v | u \rangle$. Thus we may transpose...

Lemma

The vector $|p\rangle$ is indeed unique.

Proof. ...



Proof.

Suppose $|p\rangle$ and $|q\rangle$ are two such vectors. Then

$$\langle v | a - p \rangle = 0 \quad \wedge \quad \langle v | a - q \rangle = 0$$

$|q - p\rangle \in \mathcal{V}$:

$$\| |q - p\rangle \|_2^2 = \langle p - q | p - q \rangle = 0.$$

for all $|v\rangle \in \mathcal{V}$. Take the difference and use $|v\rangle =$

But then $|q - p\rangle = 0$, ie. $|q\rangle = |p\rangle$.

□

Best-fit subspaces and SVD:

Introduction

Lemma

Given an orthonormal set $V = \{|v_i\rangle \mid i < j\} \subset \mathbb{R}^d$ spanning some subspace $\mathcal{V} \subset \mathbb{R}^d$.

1. For a vector $|a\rangle \in \mathbb{R}^d$ the projection is given by

$$\pi(|a\rangle) = \sum_{|v\rangle \in V} |v\rangle \langle v| a \rangle \in \mathcal{V} \subset \mathbb{R}^d.$$

In other words, $\pi = \sum_{|v\rangle \in V} |v\rangle \langle v|$.

2. The same vector can be described as the column $|w\rangle = [\langle v|a\rangle]_{|v\rangle \in V} = [\langle v_i|a\rangle]_{i < j} \in \mathbb{R}^j$ wrt. the orthonormal set V .
3. Its length can be computed in two ways:

$$\|\pi(|a\rangle)\|_2^2 = \||w\rangle\|_2^2.$$

Proof. . .

□

Proof.

1: Clearly, $\pi(|a\rangle) \in \mathcal{V}$.

Take any $|v'\rangle \in V$ and check

$$\begin{aligned}\langle v' | \pi(|a\rangle) &= \sum_{|v\rangle \in V} \underbrace{\langle v' | v \rangle}_{=\mathbb{1}_{|v'\rangle=|v\rangle}} \langle v | a \rangle \\ &= \langle v' | a \rangle.\end{aligned}$$

This shows $\langle v' | (|a\rangle - \pi(|a\rangle)) = 0$ for all $|v'\rangle \in V$ and thus by linearity for all $|v'\rangle \in \mathcal{V} = \text{span } V$.

So $\pi(|a\rangle)$ fulfills the definition, namely $\pi(|a\rangle) \in \mathcal{V}$ and $|a\rangle - \pi(|a\rangle) \perp \mathcal{V}$.

2: Just reformulated 1.

3: Well,

$$\begin{aligned}\|\pi(|a\rangle)\|_2^2 &= \sum_{|v\rangle \in V} \langle a | v \rangle \langle v | \cdot \sum_{|v'\rangle \in V} |v'\rangle \langle v' | a \rangle \\ &= \sum_{|v\rangle \in V} \langle a | v \rangle^2 = \||w\rangle\|_2^2\end{aligned}$$

since V is orthonormal, ie. $\langle v | v \rangle = 1$ and $\langle v | v' \rangle = 0$ for $|v\rangle, |v'\rangle \in V$ with $|v\rangle \neq |v'\rangle$. \square

Definition (best-fit j -subspace)

A j -subspace $\mathcal{V} \subset \mathbb{R}^d$ is a **best-fit j -subspace** to the points given by the rows of a matrix $A \in \mathbb{R}^{n \times d}$ iff it minimizes

$$\sum_{i < n} \| \langle A_{i,\cdot} | - \pi_{\mathcal{V}}(\langle A_{i,\cdot} |) \|_2^2.$$

We minimize the sum of squared distances of the points described by the rows of A projected to some subspace.

Best-fit subspaces and SVD:

Introduction

Lemma

Given a matrix $A \in \mathbb{R}^{n \times d}$ whose rows describe points in \mathbb{R}^d and an orthonormal basis V of a subspace $\mathcal{V} \subset \mathbb{R}^d$. Let $\pi_{\mathcal{V}}$ be the projection to \mathcal{V} .

Then the following are equivalent:

1. \mathcal{V} minimizes

$$\sum_{i < n} \|\langle A_{i,\cdot} | - \pi_{\mathcal{V}}(\langle A_{i,\cdot} |) \|_2^2.$$

$$\sum_{i < n} \|\langle A_{i,\cdot} | V \|_2^2$$

with the rows $\langle A_{i,\cdot} | V = [\langle A_{i,\cdot} | v \rangle]_{|v\rangle \in V}$.

2. \mathcal{V} maximizes

$$\sum_{i < n} \|\pi_{\mathcal{V}}(\langle A_{i,\cdot} |) \|_2^2.$$

Notice that

$$\sum_{i < n} \|\langle A_{i,\cdot} | V \|_2^2 = \sum_{|v\rangle \in V} \|A |v\rangle\|_2^2 = \|AV\|_F^2.$$

Proof. . . .

□

Proof.

We want to show equivalence of

1. \mathcal{V} minimizes $\sum_{i < n} \|\langle A_i, \cdot | - \pi_{\mathcal{V}}(\langle A_i, \cdot |)\|_2^2$.
2. \mathcal{V} maximizes $\sum_{i < n} \|\pi_{\mathcal{V}}(\langle A_i, \cdot |)\|_2^2$.
3. \mathcal{V} maximizes $\sum_{i < n} \|\langle A_i, \cdot | V\|_2^2$.

1 \iff 2: Just notice that

$$\underbrace{\|\langle A_i, \cdot | - \pi_{\mathcal{V}}(\langle A_i, \cdot |)\|_2^2}_{\mathcal{V} \text{ minimizes the sum of these}} + \underbrace{\|\pi_{\mathcal{V}}(\langle A_i, \cdot |)\|_2^2}_{\mathcal{V} \text{ maximizes the sum of these}} = \|\langle A_i, \cdot | V\|_2^2$$

and the right hand side sum is independent of \mathcal{V} . So minimizing one part is equivalent to maximizing the other.

2 \iff 3: Remember that $\|\pi_{\mathcal{V}}(\langle A_i, \cdot |)\|_2^2 = \|w\|_2^2$ where $w = (\langle A_i, \cdot | v\rangle)_{v \in V}$.

The final remark is just rewriting each of the three terms:

$$\sum_{i < n} \|\langle A_i, \cdot | V\|_2^2 = \sum_{i < n} \sum_{|v\rangle \in V} \langle A_i, \cdot | v\rangle^2,$$

$$\sum_{|v\rangle \in V} \|A|v\rangle\|_2^2 = \sum_{|v\rangle \in V} \sum_{i < n} \langle A_i, \cdot | v\rangle^2,$$

$$\|AV\|_F^2 = \sum_{i < n, |v\rangle \in V} \langle A_i, \cdot | v\rangle^2.$$

□

*Beware of the letters

Lower case

- ▶ v : latin vee
- ▶ v : math latin vee
- ▶ ν : greek nue

Upper case

- ▶ V : latin vee
- ▶ V : math latin vee
- ▶ \mathcal{V} : calligraphic latin vee

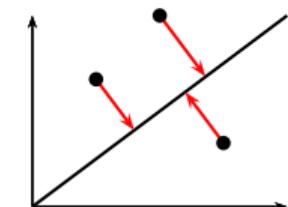
Be clear about which one you use — to yourself and your readers!

Best-fit subspaces and SVD:

Singular vectors

- ▶ Given $A \in \mathbb{R}^{n \times d}$ representing n data points $\langle A_{i,\cdot} |$ in $(\mathbb{R}^d)^\vee$.
- ▶ Let the unit vector $|v\rangle$ span the best-fit line through the origin.
- ▶ Then $|\langle A_{i,\cdot} | v \rangle|$ is the length of the projection of the i -th row $\langle A_{i,\cdot} |$ to that line.
- ▶ The sum of squares of these lengths is

$$\|A|v\rangle\|_2^2 = \sum_i \langle A_{i,\cdot} | v \rangle^2.$$



- ▶ The best-fit line
 - ▶ maximizes this sum of squared projections and
 - ▶ minimizes the sum of squared *distances*.

Definition

A/The first singular vector $|v_0\rangle$ of A is

$$|v_0\rangle := \underset{\|v\|_2=1}{\operatorname{argmax}} \|A|v\rangle\|_2.$$

The first singular value $\sigma_0(A)$ of A is $\|A|v_0\rangle\|_2$.

Note that $\sigma_0^2(A) = \sum_i \langle A_{i,\cdot} | v_0 \rangle^2$ is the sum of squared projection lengths.

A word on the 'A/The' ambiguity

...

Best-fit subspaces and SVD:

Singular vectors

Example

Take the points from the running figure

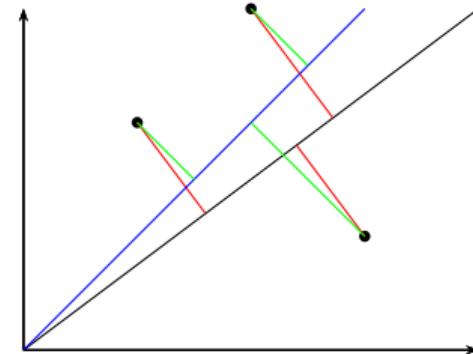
$$A = \begin{bmatrix} 1 & 2 \\ 3 & 1 \\ 2 & 3 \end{bmatrix}.$$

For $|v\rangle = \begin{bmatrix} 4 \\ 5 \\ 3 \\ 5 \end{bmatrix}$ we find $\|A|v\rangle\|_2^2 = 24.56$.

We can do better by rotating $|v\rangle$ counter-clockwise:

$$|v_0\rangle = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{\sqrt{2}}{2} \\ \frac{1}{\sqrt{2}} \\ \frac{\sqrt{2}}{2} \end{bmatrix}$$

with $\sigma_0(A)^2 = \|A|v_0\rangle\|_2^2 = 25$.



Best-fit subspaces and SVD:

Singular vectors

Definition

A/The $(j + 1)$ -th (right) singular vector $|v_j\rangle$ of A is

$$|v_j\rangle := \underset{|v\rangle \perp \{ |v_i\rangle \mid i < j\}}{\operatorname{argmax}} \|A|v\rangle\|_2$$

unless $\|A|v\rangle\|_2 = 0$ for all these vectors. The $(j + 1)$ -th singular value $\sigma_j(A)$ of A is $\|A|v_j\rangle\|_2$.

We are going to show later ([▶ rank result](#)) that in case no further right singular vector exists, ie. $\|A|v\rangle\|_2 = 0$ for all vectors $|v\rangle \perp \{ |v_i\rangle \mid i < j\}$, then $j = \operatorname{rank} A$.

Best-fit subspaces and SVD:

Singular vectors

Lemma (Properties of singular vectors)

Suppose $V = \{|v_i\rangle \mid i < r\}$ is a *complete* set of right singular vectors of $A \in \mathbb{R}^{n \times d}$ with corresponding singular values σ_i . Then

0. The set V is orthonormal and
 $\sigma_0 \geq \sigma_1 \geq \dots \geq \sigma_{r-1} > 0$.
1. For all $|w\rangle \perp V$ we have $A|w\rangle = 0$.
For short: $\ker V^T \subset \ker A$.
2. The transpose of each row
 $\langle A_{j,\cdot} | = \langle e_j |$ A is a linear combination
of singular vectors.
For short: $\text{im } A^T \subset \text{span } V$.

3. Explicitely, we have

$$A^T |e_j\rangle = \sum_{|v\rangle \in V} |v\rangle \langle v| A^T |e_j\rangle.$$

4. $\text{rank } A \leq \#V = r$.

Proof. . .



Proof.

0. The set V is orthonormal and $\sigma_0 \geq \dots \geq \sigma_{r-1} > 0$:

By definition, each right singular vector is a unit vector and orthogonal to all previous ones. Since the argmax extends over smaller and smaller sets it cannot increase. The final one is still positive as otherwise no further right singular vector is taken.

1. For all $|w\rangle \perp V$ we have $A|w\rangle = 0$: By definition, another singular vector is determined until that condition holds.

2. The transpose of each row $\langle A_{j,\cdot}| = \langle e_j| A$ is a linear combination of singular vectors: Choose an orthonormal basis \widehat{V} for \mathbb{R}^d extending V . Then $1 = \sum_{|v\rangle \in \widehat{V}} |v\rangle \langle v|$

and so

$$A^T |e_j\rangle = \sum_{|v\rangle \in \widehat{V}} |v\rangle \langle v| A^T |e_j\rangle .$$

(This means: Write $A^T |e_j\rangle$ in this basis.)

Now, for $|w\rangle \in \widehat{V} \setminus V$ we have $|w\rangle \perp V$ and so $A|w\rangle = 0$. Consequently, $\langle w| A^T |e_j\rangle = 0$.

We can thus shorten the above combination:

$$A^T |e_j\rangle = \sum_{|v\rangle \in V} |v\rangle \langle v| A^T |e_j\rangle .$$

In other words, the transpose of row j is in the span of V , as claimed.

4. **rank $A \leq \#V$** : The rank is the dimension of the row space of A —or, equivalently, its column space. And by the previous, this row space is contained in $\text{span } V$. Since V is orthonormal that span has dimension $\#V$. We conclude $\text{rank } A \leq \#V$. \square

Best-fit subspaces and SVD:

Singular vectors

Idea (The greedy algorithm)

Find singular vectors $|v_0\rangle, |v_1\rangle, \dots, |v_{r-1}\rangle$ with singular values $\sigma_0, \sigma_1, \dots, \sigma_{r-1}$ one-by-one according to this definition, ie.

$$|v_j\rangle := \underset{|v\rangle \perp \{ |v_i\rangle \mid i < j\}}{\operatorname{argmax}} \|A|v\rangle\|_2, \quad \sigma_j := \|A|v_j\rangle\|_2.$$

We hope that this will also give us a best-fit k -subspace by considering the space

$$\mathcal{V}_k = \operatorname{span} \{ |v_i\rangle \mid i < k\}$$

spanned by the first k singular vectors.

Theorem (The greedy algorithm works)

Let A be an $n \times d$ -matrix with singular vectors $|v_i\rangle$, $i < r$, and $1 \leq k \leq r$.

Let $\mathcal{V}_k = \text{span} \{ |v_i\rangle \mid i < k\}$ be the subspace spanned by the first k singular vectors.
Then

- ▶ \mathcal{V}_k is the best-fit k -subspace for A .

Proof. . . .

□

Proof.

We proceed by induction on k . We have to show that

$$\sum_{i < k} \|A|v_i\rangle\|^2$$

is maximal when $\{|v_i\rangle \mid i < k\}$ could vary over all orthonormal k -sets in \mathbb{R}^d .

$k = 1$: Nothing to prove for $k = 1$.

$k > 1$: By assumption $\mathcal{V}_{k-1} = \text{span}\{|v_i\rangle \mid i < k-1\}$ is a best-fit $(k-1)$ -subspace.

Take any best-fit k -subspace \mathcal{W} . Pick an orthonormal basis $\{|w_i\rangle \mid i < k\}$ of \mathcal{W} so that $|w_{k-1}\rangle$ is orthogonal to \mathcal{V}_{k-1} . $\lceil |w_{k-1}\rangle \exists$: Vectors in \mathcal{W} have k degrees of freedom and we are requiring $k-1$ conditions, namely $|w_{k-1}\rangle \perp |v_i\rangle$

for $i < k-1$. Then

$$\sum_{i < k-1} \|A|w_i\rangle\|_2^2 \leq \sum_{i < k-1} \|A|v_i\rangle\|_2^2$$

since \mathcal{V}_{k-1} is a best-fit $(k-1)$ -subspace.

Next, by definition of $|v_{k-1}\rangle$ and since $|w_{k-1}\rangle \perp \mathcal{V}_{k-1}$ we have $\|A|w_{k-1}\rangle\|_2^2 \leq \|A|v_{k-1}\rangle\|_2^2$. But then

$$\sum_{i < k} \|A|w_i\rangle\|_2^2 \leq \sum_{i < k} \|A|v_i\rangle\|_2^2$$

and so \mathcal{V}_k is at least as good as \mathcal{W} and thus optimal. □

Best-fit subspaces and SVD:

Singular vectors

Example

Given $A = \begin{bmatrix} 4 & & \\ & 1 & \\ & & 0 \\ & & 0 \end{bmatrix}$.

What are its singular vectors?

Well, $|v_0\rangle = \begin{bmatrix} 1 \\ \cdot \\ \cdot \end{bmatrix}$ with $\sigma_0 = 4 = \left\| \begin{bmatrix} 4 \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} \right\|_2$.

And $|v_1\rangle = \begin{bmatrix} \cdot \\ 1 \\ \cdot \end{bmatrix}$ with $\sigma_1 = 1 = \left\| \begin{bmatrix} \cdot \\ 1 \\ \cdot \end{bmatrix} \right\|_2$.

No further singular vector exists.

The best-fit line: is the line

$$\mathbb{R}|v_0\rangle.$$

We find $\|A|v_0\rangle\|_2 = 4$ which is clearly maximum as no other vector is stretched further.

The best-fit 2-subspace: is the subspace

$$\text{span}\{|v_0\rangle, |v_1\rangle\}.$$

(The transpose of) each row is inside and thus their accumulated squared distances sum up to 0 which is clearly minimum.

Best-fit subspaces and SVD:

Singular vectors

Recall: For a symmetric square matrix A with ev. $\alpha_0, \dots, \alpha_{n-1}$ we have $\|A\|_F^2 = \sum_{i < n} \alpha_i^2$.

Lemma

For any matrix A , the sum of squares of the singular values equals the square of the Frobenius norm. That is,

$$\|A\|_F^2 = \sum_{i < r} \sigma_i^2(A).$$

Proof. . . .

□

Proof.

Consider a row $\langle A_{j,\cdot} |$ of A and let V be the set of all its singular vectors. By the properties of singular vectors (3) we know that

$$A^T |e_j\rangle = \sum_{|v\rangle \in V} |v\rangle \langle v| A^T |e_j\rangle.$$

We use that to compute the length of row j twice:

$$\begin{aligned}\|A^T |e_j\rangle\|_2^2 &= \sum_{|v\rangle \in V} \langle v| A^T |e_j\rangle^2 = \sum_{|v\rangle \in V} \langle A_{j,\cdot} | v\rangle^2 \\ &= \sum_{k < d} \langle e_k | A^T |e_j\rangle^2 = \sum_{k < d} A_{j,k}^2.\end{aligned}$$

Summing over all rows j yields

$$\|A\|_F^2 = \sum_{j < n} \sum_{k < d} A_{j,k}^2$$

$$\begin{aligned}&= \sum_{j < n} \sum_{|v\rangle \in V} \langle A_{j,\cdot} | v\rangle^2 \\ &= \sum_{|v\rangle \in V} \sum_{j < n} \langle A_{j,\cdot} | v\rangle^2 \\ &= \sum_{|v\rangle \in V} \|A |v\rangle\|_2^2 \\ &= \sum_{|v\rangle \in V} \sigma_v^2(A).\end{aligned}$$

where $\sigma_v(A) = \sigma_v = \|A |v\rangle\|_2$ is the singular value corresponding to the singular vector $|v\rangle \in V$. \square

Definition

Given the right singular vectors $|v_0\rangle, \dots, |v_{r-1}\rangle$, we define
the $(i+1)$ -th left singular vector

$$|u_i\rangle := \frac{1}{\sigma_i(A)} A |v_i\rangle,$$

i.e. the normalized image of the $(i+1)$ -th right singular vector.

Later, we will show that —despite their completely different definition— the left singular vectors fulfill a similar property than the right singular vectors. Namely:

- ▶ $|u_i\rangle$ maximizes $\|\langle u_i | A\|_2$ over all unit vectors orthogonal to $|u_0\rangle, \dots, |u_{i-1}\rangle$.
- ▶ The left singular vectors are orthogonal.

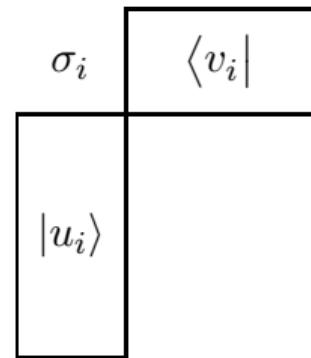
Best-fit subspaces and SVD:

Singular value decomposition (SVD)

Theorem (Singular value decomposition (SVD))

Let A be an $n \times d$ -matrix with right singular vectors $|v_0\rangle, |v_1\rangle, \dots, |v_{r-1}\rangle$ and left singular vectors $|u_0\rangle, |u_1\rangle, \dots, |u_{r-1}\rangle$ with corresponding singular values $\sigma_0, \sigma_1, \dots, \sigma_{r-1}$. Then

$$A = \sum_{i < r} \sigma_i |u_i\rangle \langle v_i|.$$



Proof. . .

Proof.

Given the right singular vectors $|v_i\rangle$, $i < r$, the left singular vectors $|u_i\rangle = \frac{1}{\sigma_i}A|v_i\rangle$ we want:

$$A = \sum_{i < r} \sigma_i |u_i\rangle \langle v_i|$$

Apply both sides to $|v_j\rangle$:

$$\sum_{i < r} \sigma_i |u_i\rangle \underbrace{\langle v_i | v_j \rangle}_{=1_{i=j}} = \sigma_j |u_j\rangle = A|v_j\rangle.$$

For any vector $|v\rangle$ orthogonal to all $|v_i\rangle$ we have

$$\sum_{i < r} \sigma_i |u_i\rangle \langle v_i | v \rangle = 0 = A|v\rangle$$

since there is no further singular value. But now both sides coincide on a basis and thus anywhere. \square

Index saving variant: Describe the SVD by a set $S \subset \mathbb{R} \times \partial B^n \times \partial B^d$ of **singular triples** $[\sigma, |u\rangle, |v\rangle]$, each consisting of a right singular vector $|v\rangle$, the corresponding singular value $\sigma = \|A|v\rangle\|_2$ and the corresponding left singular vector $|u\rangle = \frac{1}{\sigma}A|v\rangle$. In particular, this defines the space $\mathcal{V}_S = \text{span}\{|v\rangle \mid [\sigma, |u\rangle, |v\rangle] \in S\}$ spanned by the right singular vectors.

Proof.

Given the SVD $S \subset \mathbb{R} \times \partial B^n \times \partial B^d$ of singular triples $[\sigma, |u\rangle, |v\rangle]$, we want

$$A = \sum_{[\sigma, |u\rangle, |v\rangle] \in S} \sigma |u\rangle \langle v|.$$

Apply both sides to $|\hat{v}\rangle$ from any singular triple $[\hat{\sigma}, |\hat{u}\rangle, |\hat{v}\rangle] \in S$:

$$\sum_{[\sigma, |u\rangle, |v\rangle] \in S} \sigma |u\rangle \langle v | \hat{v}\rangle = \hat{\sigma} |\hat{u}\rangle = A|\hat{v}\rangle.$$

Further, for any vector $|v'\rangle$ orthogonal to \mathcal{V}_S we have

$$\sum_{[\sigma, |u\rangle, |v\rangle] \in S} \sigma |u\rangle \langle v | v'\rangle = 0 = A|v'\rangle$$

since there is no further singular value. But now both sides coincide on a basis and thus anywhere. \square

Best-fit subspaces and SVD:

Singular value decomposition (SVD)

Thus any $n \times d$ -matrix A of rank r has a **singular value decomposition** (SVD):

$$\begin{array}{c|c} A & \\ \hline n \times d & \end{array} = \begin{array}{c|c} U & \\ \hline n \times r & \end{array} \cdot \begin{array}{c|c} D & \\ \hline r \times r & \end{array} \cdot \begin{array}{c|c} V^T & \\ \hline r \times d & \end{array}$$

The columns of V , ie. rows of V^T , are the right singular vectors, the diagonal elements of D are the singular values and the columns of U are the left singular vectors.

Best-fit subspaces and SVD:

Best rank- k approximation

Let $A = \sum_{i < r} \sigma_i |u_i\rangle \langle v_i|$ be the SVD of A and consider the **k -truncated SVD**

$$A_k := \sum_{i < k} \sigma_i |u_i\rangle \langle v_i|,$$

i.e. the sum truncated after k terms, $k \leq r$.

Lemma

The rows of A_k are the projections of the rows of A to the best-fit k -subspace \mathcal{V}_k spanned by the first k singular vectors of A .

Proof. . .



Proof.

For any row vector $\langle a |$ the projection to \mathcal{V}_k is given by

$$\pi_{\mathcal{V}_k}(\langle a |) = \sum_{i < k} \langle a | v_i \rangle \langle v_i |.$$

Using this for each row $\langle A_{j,\cdot} | = \langle e_j | A$ of A we get

$$\sum_{i < k} A |v_i\rangle \langle v_i| = \sum_{i < k} \sigma_i |u_i\rangle \langle v_i| = A_k.$$

Consequently, row j of A_k is

$$\langle e_j | A_k = \sum_{i < k} \langle e_j | A | v_i \rangle \langle v_i | = \pi_{\mathcal{V}_k} (\langle e_j | A),$$

that is, the projection of row j of A .

□

Theorem

*The k -truncated SVD A_k is the best rank- k approximation of A wrt. the Frobenius norm:
For any matrix B of rank at most k we have*

$$\|A - A_k\|_F \leq \|A - B\|_F.$$

Proof. . . .



Proof.

Assume B minimizes $\|A - B\|_F^2$ among all at most rank k matrices.

Consider the space $\mathcal{V} = \text{span } B^T$ spanned by the rows of B . Clearly, $\dim \mathcal{V} = \text{rank } B \leq k$.

Claim

Wlog. each row of B is the projection of the corresponding row of A to \mathcal{V} .

[If not, replace the row of B by that projection of the corresponding row of A to \mathcal{V} .

The spanned space is still in \mathcal{V} and so the rank of the modified B is still at most k .

But this reduces $\|A - B\|_F^2$. Why? Well, because the change only affects the summand measuring the squared distance of the corresponding row in A . And that is optimal when we take the projection.]

With the claim, $\|A - B\|_F^2$ is the sum of the squared distances of rows of A to the subspace \mathcal{V} spanned by the rows of B .

But since the greedy algorithm works, this is minimized by \mathcal{V}_k spanned by A_k and thus $\|A - A_k\|_F \leq \|A - B\|_F$. □

An application: approximating $A|x\rangle\dots$

Task

Given many vectors $|x\rangle \in \mathbb{R}^d$ compute all results $A|x\rangle$.

Time per vector (without optimizations): $\mathcal{O}(nd)$.

Best-fit subspaces and SVD:

Best rank- k approximation

An application: approximating $A|x\rangle\dots$

Task

Given many vectors $|x\rangle \in \mathbb{R}^d$ approximate all results $A|x\rangle$.

What if we can tolerate a small error?

Solution

- ▶ Compute the k -truncated SVD $A_k = \sum_{i < k} \sigma_i |u_i\rangle \langle v_i|$.
- ▶ Compute $A_k|x\rangle$ instead of the correct value.

Time per vector: $\mathcal{O}(kn + kd)$.

For small/bounded k this is much faster than the previous $\mathcal{O}(nd)$.

Best-fit subspaces and SVD:

Best rank- k approximation

An application: approximating $A|x\rangle\dots$

Question

How large is the error?

We do not know $|x\rangle$. So we measure the maximum of

$$\|(A_k - A)|x\rangle\|_2$$

over all $\| |x\rangle \|_2 \leq 1$. This maximum is the **2-operator norm** or **spectral norm**

$$\|A_k - A\|_2$$

of $A_k - A$.

In our case it equals σ_k (we are going to prove that soon).

Theorem

The left singular vectors are pairwise orthogonal.

Proof. . . .



The key to this proof is to **disturb the situation** slightly. We assume that the left singular vectors are **not** orthogonal. Then we achieve a **contradiction** by showing that one of the right singular vectors does not maximize its goal.

Proof.

We proceed by induction and consider the smallest index i such that $|u_i\rangle$ is **not** orthogonal to $|u_j\rangle$ for some $j \neq i$. Clearly, $i < j$.

Wlog. $\delta := \langle u_i | u_j \rangle > 0$. [Otherwise replace $|v_j\rangle$ with $-|v_j\rangle$ and thus $|u_j\rangle$ with $-|u_j\rangle$.]

Now, for $\varepsilon > 0$ disturb $|v_i\rangle$ and renormalize:

$$|v'_i\rangle := \frac{1}{\||v_i\rangle + \varepsilon|v_j\rangle\|_2}(|v_i\rangle + \varepsilon|v_j\rangle).$$

Applying A yields

$$A|v'_i\rangle = \frac{1}{\sqrt{1+\varepsilon^2}}(\sigma_i|u_i\rangle + \varepsilon\sigma_j|u_j\rangle).$$

Finally, estimate the length $\|A|v'_i\rangle\|_2$ by computing its

component along $|u_i\rangle$:

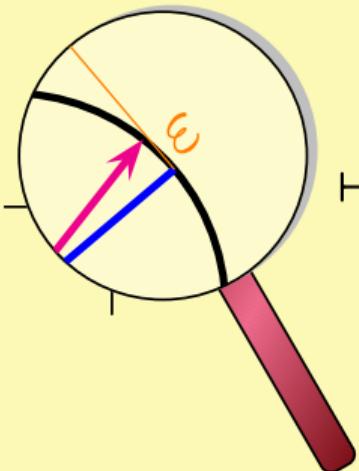
$$\begin{aligned}\|A|v'_i\rangle\|_2 &\geq \langle u_i | A|v'_i\rangle = \frac{\sigma_i + \delta\varepsilon\sigma_j}{\sqrt{1+\varepsilon^2}} \\ &= \frac{\sigma_i}{\sqrt{1+\varepsilon^2}} + \frac{\delta\sigma_j}{\sqrt{1+\varepsilon^{-2}}}.\end{aligned}$$

The rhs at $\varepsilon = 0$ equals σ_i and the rhs's derivative wrt. ε (namely $\frac{-\varepsilon\sigma_i + \delta\sigma_j}{(1+\varepsilon^2)^{\frac{3}{2}}}$) at $\varepsilon = 0$ equals $\delta\sigma_j$. Since that is positive the rhs must increase for small $\varepsilon > 0$ and we obtain $\|A|v'_i\rangle\|_2 > \sigma_i$ for small $\varepsilon > 0$.

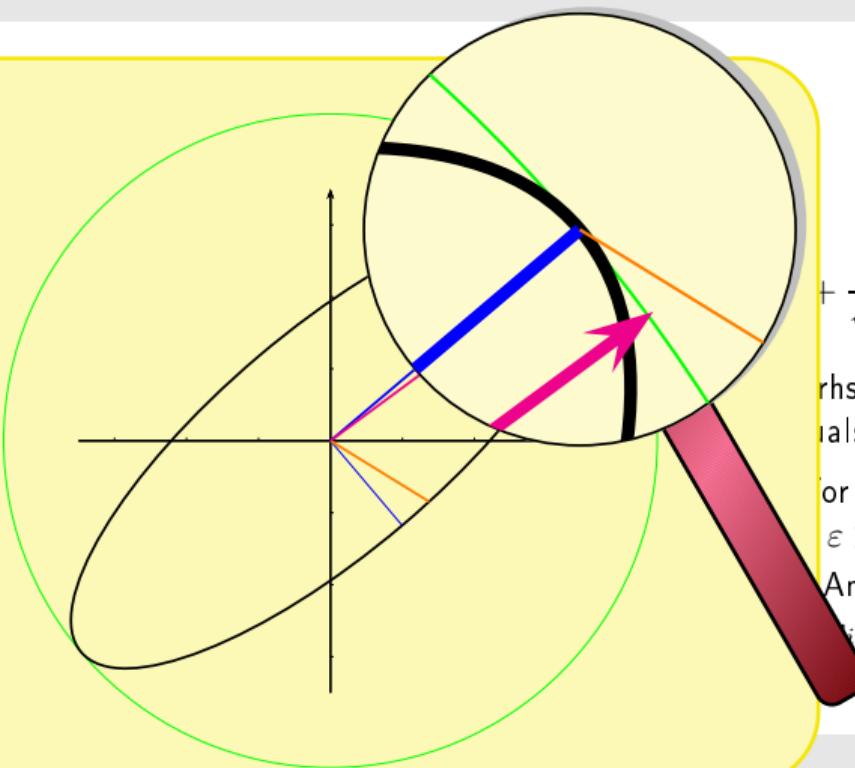
By construction $|v'_i\rangle \perp \{|v_k\rangle \mid k < i\}$. And it yields a larger streching factor $\|A|v'_i\rangle\|_2$ than $|v_i\rangle$. A contradiction. □

The key to this proof is to **disturb the situation** slightly. We assume that the left singular vectors are **not** orthogonal. Then we achieve a **contradiction** by showing that one of the right singular vectors does not maximize its goal.

Pr



\xrightarrow{A}



$$+ \frac{\delta\sigma_j}{\sqrt{1 + \varepsilon^{-2}}}.$$

rhs's derivative
ials $\delta\sigma_j$. Since
or small $\varepsilon > 0$
 $\varepsilon > 0$.

And it yields a
 $\langle \cdot, \cdot \rangle$. A contra-

□

Corollary

Let $V = \{|v_i\rangle \mid i < r\}$ be a complete set of right singular vectors. In particular, $\|A|v\rangle\|_2 = 0$ for all vectors $|v\rangle \perp V$, ie. there is no further singular vector. Then

$$\operatorname{rank} A = r.$$

Proof. . . .



Proof.

We know that

$$A = \sum_{|v\rangle \in V} \sigma_v |u_v\rangle \langle v|$$

where $V = \{|v_0\rangle, \dots, |v_{r-1}\rangle\}$ is the set of the r right singular vectors, and $\sigma_v = \|A|v\rangle\|_2$, $|u_v\rangle = \frac{1}{\sigma_v} A|v\rangle$

for $|v\rangle \in V$. By the previous theorem, also the set $\{|u_v\rangle \mid |v\rangle \in V\}$ of left singular vectors is orthonormal. Since orthonormal implies linearly independent and

$$\text{im } A = \text{span } \{|u_v\rangle \mid |v\rangle \in V\},$$

we obtain $\text{rank } A = \dim \text{im } A = r$. □

Best-fit subspaces and SVD:

Left singular vectors

Lemma (Analog of eigenvalues and eigenvectors)

$$A |v_i\rangle = \sigma_i |u_i\rangle \quad \text{and} \quad A^T |u_i\rangle = \sigma_i |v_i\rangle.$$

Proof. . . .

□

This directly yields eigenvalues and eigenvectors of $A^T A$ and AA^T :

Corollary

$$A^T A |v_i\rangle = \sigma_i^2 |v_i\rangle \quad \text{and} \quad AA^T |u_i\rangle = \sigma_i^2 |u_i\rangle.$$

□

Proof.

$A|v_i\rangle = \sigma_i|u_i\rangle$: This first equation is the definition of the left singular vectors.

$A^T|u_i\rangle = \sigma_i|v_i\rangle$: For the second one, apply the SVD to each $|u_i\rangle$:

$$A^T|u_i\rangle = \sum_j \sigma_j|v_j\rangle \underbrace{\langle u_j | u_i \rangle}_{=\mathbb{1}_{i=j}} = \sigma_i|v_i\rangle$$

using that the $|u_j\rangle$ are orthonormal.

□

Best-fit subspaces and SVD:

Left singular vectors

Theorem (SVD from a twosided orthonormal decomposition)

Given a decomposed matrix

$$A := \sum_{i < r} \sigma_i |u_i\rangle \langle v_i|$$

where $V = [|v_i\rangle \mid i < r] \subset \mathbb{R}^d$ and $U = [|u_i\rangle \mid i < r] \subset \mathbb{R}^n$ are **orthonormal systems of r vectors** and $\sigma = [\sigma_i \mid i < r] \in \mathbb{R}_{>0}^r$ is non-increasing. Then for each $k < r$

- ▶ $|v_k\rangle$ is the $(k + 1)$ -th right singular vector,
- ▶ $|u_k\rangle$ is the $(k + 1)$ -th left singular vector, and
- ▶ σ_k the $(k + 1)$ -th singular value.

In other words, the given decomposition is the singular value decomposition.

Proof. . .



Proof.

Well, suppose

$$A = \sum_{i < r} \sigma_i |u_i\rangle \langle v_i|$$

where $V = [|v_i\rangle | i < r] \subset \mathbb{R}^d$ and $U = [|u_i\rangle | i < r] \subset \mathbb{R}^n$ are orthonormal systems of r vectors and $\sigma = [\sigma_i | i < r] \in \mathbb{R}_{>0}^r$ is non-increasing.

Clearly, $A|v_k\rangle = \sigma_k|u_k\rangle$ and $\|A|v_k\rangle\|_2 = \sigma_k$. So it only remains to prove that $|v_k\rangle$ is the $(k+1)$ -th right singular vector. [Namely, by the previous, $|u_k\rangle$ is the corresponding left singular vector and σ_k is the corresponding singular value.]

Consider $|v\rangle \perp \{ |v_i\rangle | i < k\}$.

Write $|v\rangle = \sum_{i < r} \nu_i |v_i\rangle + |v'\rangle$ with $\nu_i \in \mathbb{R}$, $|v'\rangle \perp V$.

We always have

$$1 = \| |v\rangle \|_2^2 = \sum_{i < r} \nu_i^2 + \| |v'\rangle \|_2^2. \quad (1)$$

Since $|v\rangle \perp \{ |v_i\rangle | i < k\}$ we have $\nu_i = 0$ for $i < k$. Further, $A|v'\rangle = 0$. Since U is orthonormal we obtain

$$\begin{aligned} \|A|v\rangle\|_2^2 &= \left\| \sum_{k \leq i < r} \nu_i \sigma_i |u_i\rangle \right\|_2^2 \\ &= \sum_{k \leq i < r} \nu_i^2 \sigma_i^2 \leq \sigma_k^2. \end{aligned}$$

Equality holds for $\nu_k = 1$, which with (1) implies $\nu_i = 0$ for $k < i < r$ and $|v'\rangle = 0$. Thus

$$|v_k\rangle = \operatorname*{argmax}_{|v\rangle \perp \{ |v_i\rangle | i < k\}} \|A|v\rangle\|_2$$

and $|v_k\rangle$ is a $(k+1)$ -th right singular vector. \square

\square

Corollary

The left singular vectors of A are the right singular vectors of A^T . In particular, they maximize scaling factors like the right singular vectors, namely the $(j+1)$ -th left singular vector fulfills

$$\langle u_j | = \operatorname{argmax}_{|u\rangle \perp \{|u_i\rangle | i < j\}} \| \langle u | A \|_2 .$$

Proof. . . .

□

Note: Using this iterative way as definition for both left and right vectors would result in a matching problem, since uniqueness is only weakly granted. Thus we stick with the introduced asymmetric definition.

Proof.

Well, just notice that given an SVD $A = \sum_{i < r} \sigma_i |u_i\rangle \langle v_i|$ of A with the previous theorem we also have an SVD of A^T :

$$A^T = \sum_{i < r} \sigma_i |v_i\rangle \langle u_i|.$$



Best-fit subspaces and SVD:

Left singular vectors

Lemma

$$\|A - A_k\|_2 = \sigma_k.$$

Proof. . . .



... this completes the answer on how large the error is at most when approximating $A|x\rangle$ with $A_k|x\rangle$. Namely, the error $\frac{1}{\| |x\rangle \|_2} \|A_k |x\rangle - A |x\rangle\|_2$ relative to the length of $|x\rangle$ is at most

$$\|A - A_k\|_2 = \sigma_k.$$

Proof.

Obviously,

$$A - A_k = \sum_{k \leq i < r} \sigma_i |u_i\rangle \langle v_i|$$

is the SVD of $A - A_k$.

Again, writing a vector as $|v\rangle = \sum_{j < r} \nu_j |v_j\rangle + |v'\rangle$ with $|v'\rangle \perp \{|v_i\rangle \mid i < r\}$ we have

$$(A - A_k) |v\rangle = \sum_{k \leq i < r} \sigma_i \nu_i |u_i\rangle.$$

And thus

$$\underbrace{\max_{\| |v\rangle \|_2 = 1} \| (A - A_k) |v\rangle \|_2}_{= \|A - A_k\|_2} = \sigma_k.$$

And that's the claim. □



Left singular vectors

Actually, A_k is not only the best rank- k approximation

- ▶ wrt. the Frobenius norm
- ▶ but also wrt. 2-norm:

Theorem

Let A be an $n \times d$ -matrix. For any matrix B of rank at most k we have

$$\|A - A_k\|_2 \leq \|A - B\|_2.$$

Proof. . .



Proof.

Take a unit vector $|z\rangle$ in

$$\underbrace{\{|z\rangle \mid B|z\rangle = 0\}}_{\substack{\dim(\%)=d-\text{rank } B \\ \geq d-k}} \cap \underbrace{\text{span}\{|v_0\rangle, \dots, |v_k\rangle\}}_{\dim(\%)=k+1}.$$

Since B has rank at most k this intersection is non-trivial. Now, $\|A - B\|_2^2 \geq \|(A - B)|z\rangle\|_2^2 = \|A|z\rangle\|_2^2$. Using the SVD of A we obtain

$$\|A|z\rangle\|_2^2 = \left\| \sum_i \sigma_i |u_i\rangle \langle v_i| z \right\|_2^2$$

$$\begin{aligned} &= \sum_i \sigma_i^2 \langle v_i | z \rangle^2 \\ &= \sum_{i \leq k} \sigma_i^2 \langle v_i | z \rangle^2 \\ &\geq \sigma_k^2 \sum_{i \leq k} \langle v_i | z \rangle^2 = \sigma_k^2. \end{aligned}$$

Combining this with the previous lemma stating $\sigma_k = \|A - A_k\|_2$, we are done. \square

Best-fit subspaces and SVD:

Left singular vectors

Summary on [reduced] SVD

- ▶ **Construction:** iteratively define right singular vectors

$$|v_j\rangle := \underset{|v\rangle \perp \{ |v_i\rangle \mid i < j\}}{\operatorname{argmax}} \|A|v\rangle\|_2,$$

singular values $\sigma_j = \|A|v_j\rangle\|_2$ and left singular vectors $|u_j\rangle = \frac{1}{\sigma_j} A|v_j\rangle$.

- ▶ **Existence:** Any matrix has an SVD:

$$A = \sum_{i < r} \sigma_i |u_i\rangle \langle v_i| = UDV^T$$

with

- ▶ $U = [|u_i\rangle]_{i < r}$ orthonormal,
- ▶ $V = [|v_i\rangle]_{i < r}$ orthonormal, and
- ▶ $D = \operatorname{diag}(\sigma_i \mid i < r)$ descending positive.

- ▶ **Best-fit k -subspace:** Greedy works!

$\mathcal{V}_k := \operatorname{span} A_k$ is best-fit k -subspace.

- ▶ **Inverse:** Any decomposition with the properties listed with the existence **is** an SVD.

- ▶ **Symmetry!**

- ▶ **Least squares rank- k approximation:**

$$\|A - A_k\|_F = \text{minimum}.$$

- ▶ **Minimal approximation error rank- k approx.:**

$$\|A - A_k\|_2 = \text{minimum}.$$

- ▶ Analog of eigenvalues and eigenvectors:

$$A|v_i\rangle = \sigma_i|u_i\rangle, \quad A^T|u_i\rangle = \sigma_i|v_i\rangle.$$

Section overview

Organizational

Introduction

High-dimensional space

Gaussians in high dimensions

Eigenvalues and eigenvectors

Best-fit subspaces and SVD

Power method for SVD

Applications of SVD

Machine learning

*Clustering

Summary / Outro

Consider $B = A^T A$. Using an SVD of A we have

$$\begin{aligned} B = A^T A &= \left(\sum_i \sigma_i |v_i\rangle \langle u_i| \right) \left(\sum_j \sigma_j |u_j\rangle \langle v_j| \right) \\ &= \sum_i \sigma_i^2 |v_i\rangle \langle v_i| \end{aligned}$$

by using that $(|u_i\rangle)$ is orthonormal, ie. $\langle u_i | u_j \rangle = \mathbb{1}_{i=j}$.

Similarly,

$$\begin{aligned}B^2 &= (A^T A)^2 = \left(\sum_i \sigma_i^2 |v_i\rangle \langle v_i| \right) \left(\sum_j \sigma_j^2 |v_j\rangle \langle v_j| \right) \\&= \sum_i \sigma_i^4 |v_i\rangle \langle v_i|\end{aligned}$$

since $(|v_i\rangle)$ is orthonormal.

Iterating we arrive at

$$B^k = \sum_i \sigma_i^{2k} |v_i\rangle \langle v_i|.$$

Using

$$B^k = \sum_i \sigma_i^{2k} |v_i\rangle \langle v_i|.$$

and assuming $\sigma_0 > \sigma_1$ we obtain

$$\begin{aligned}\sigma_0^{-2k} B^k &= |v_0\rangle \langle v_0| + \left(\frac{\sigma_1}{\sigma_0}\right)^{2k} |v_1\rangle \langle v_1| + \dots \\ &\xrightarrow{k \rightarrow \infty} |v_0\rangle \langle v_0|.\end{aligned}$$

To approximate $|v_0\rangle$ we can thus simply

- ▶ take any column of B^k and
- ▶ normalize it to a unit vector.

However, we have to multiply matrices and that is rather **slow**, in particular, when the matrices are huge.

A faster variant

Idea

When applying $B = A^T A$ to a vector $|v\rangle$, its behaviour is still ruled by the SVD.

Say, $|v\rangle = \sum_{k < r} \gamma_k |v_k\rangle + |v'\rangle$ with $|v'\rangle \perp V$, then

$$\begin{aligned} B^\ell |v\rangle &= \left(\sum_{i < r} \sigma_i^{2\ell} |v_i\rangle \langle v_i| \right) \left(\sum_{k < r} \gamma_k |v_k\rangle + |v'\rangle \right) \\ &= \sum_{k < r} \sigma_k^{2\ell} \gamma_k |v_k\rangle \end{aligned}$$

since $(|u_i\rangle)$ and $(|v_j\rangle)$ are both orthonormal, ie. $\langle u_i | u_j \rangle = \mathbb{1}_{i=j}$ and $\langle v_j | v_k \rangle = \mathbb{1}_{j=k}$. Clearly, $\sigma_0^{2\ell}$ is the largest scaling factor, so the component in direction $|v_0\rangle$ of the largest singular value is amplified compared to all others.

Idea

To approximate $|v_0\rangle$ we compute

$$\frac{1}{\|(A^T A)^\ell |v\rangle\|_2} (A^T A)^\ell |v\rangle$$

with cheaper matrix vector products.

This is particularly important if A is sparse.

Algorithm PM0

Input: A, ℓ .

Output: An approximation $[\tilde{\sigma}_0, |\tilde{u}_0\rangle, |\tilde{v}_0\rangle]$ to $[\sigma_0, |u_0\rangle, |v_0\rangle]$.

1. Pick a random vector $|v\rangle$.
2. **For** ℓ times **do** 3–4
3. $|v\rangle \leftarrow A|v\rangle$.
4. $|v\rangle \leftarrow A^T|v\rangle$.
5. $|\tilde{v}_0\rangle \leftarrow \frac{1}{\| |v\rangle \|_2} |v\rangle$.
6. $|u\rangle \leftarrow A \cdot |\tilde{v}_0\rangle$.
7. $\tilde{\sigma}_0 \leftarrow \| |u\rangle \|_2$.
8. $|\tilde{u}_0\rangle \leftarrow \frac{1}{\tilde{\sigma}_0} |u\rangle$.
9. **Return** $[\tilde{\sigma}_0, |\tilde{u}_0\rangle, |\tilde{v}_0\rangle]$.

- ▶ Next, you may want to find or approximate [Algorithm PM0.k](#)
 $\sigma_1, |u_1\rangle, |v_1\rangle$.
- ▶ To do so, consider

$$R \leftarrow A - \tilde{\sigma}_0 \cdot |\tilde{u}_0\rangle \langle \tilde{v}_0|.$$

As $A = \sum_i \sigma_i |u_i\rangle \langle v_i|$ this matrix is close to

$$A - A_1 = \sum_{i \geq 1} \sigma_i |u_i\rangle \langle v_i|.$$

- ▶ Actually, $|v_0\rangle$ will usually still be almost a singular vector but with a tiny singular value.
- ▶ How tiny depends on the quality of the approximation...
- ▶ Iterating should yield suitable approximations for the topmost singular values.

Input: A, k, ℓ .

Output: Approximations

$[\tilde{\sigma}_i, |\tilde{u}_i\rangle, |\tilde{v}_i\rangle]$ to the largest k singular values.

1. Let $R \leftarrow A$.
2. **For** $i < k$ **do** 3–4
3. Call [Algorithm PM0](#) on R, ℓ to obtain $[\tilde{\sigma}_i, |\tilde{u}_i\rangle, |\tilde{v}_i\rangle]$.
4. Compute
$$R \leftarrow R - \tilde{\sigma}_i \cdot |\tilde{u}_i\rangle \langle \tilde{v}_i|.$$
5. **Return** $[\tilde{\sigma}_i, |\tilde{u}_i\rangle, |\tilde{v}_i\rangle]_{i < k}$

- ▶ Can we obtain k singular vectors all at once?

Idea: The previous method should also work for subspaces!

- ▶ Other approaches first try to reduce the matrix size, eg. by first finding a QR-factorization —say, by Gram-Schmidt orthogonalization.

Namely, if $A = Q \cdot R$ with Q orthogonal and R upper triangular and we find the SVD $R = U' \cdot \Sigma \cdot V^T$, then

$$A = Q U' \cdot \Sigma \cdot V^T.$$

Ideally, R is only $d \times d \dots$

- ▶ The python routine `numpy.linalg.svd` actually only

- ▶ wraps C code `umath_linalg.c` which
- ▶ wraps FORTRAN code `dgesdd.f` from LAPACK — Linear Algebra PACKAGE.

There various techniques are employed:

- ▶ QR-factorization $A = Q \cdot R$,
- ▶ bidiagonalize R and
- ▶ then decompose by a divide-and-conquer algorithm specially optimized for bidiagonal matrices.

- ▶ Various variants of the power method work good if $\sigma_0 \geq \sigma_1$ are far enough apart.
- ▶ Problematic, if σ_0 too close to σ_1 .
- ▶ But still the basic method works as follows by the following theorem.

Theorem

Let A be an $n \times d$ matrix and $|x\rangle$ a unit length vector in \mathbb{R}^d with $|\langle v_0 | x \rangle| \geq \delta$, where $\delta > 0$. Let $\mathcal{V}_{1-\varepsilon}$ be the space spanned by the right singular vectors of A corresponding to singular values greater than $(1 - \varepsilon)\sigma_0$. Let $|w\rangle$ be the unit vector after $\ell \geq \frac{-\ln(\varepsilon\delta)}{2\varepsilon}$ iterations of the power method, namely,

$$|w\rangle = \frac{(A^T A)^\ell |x\rangle}{\|(A^T A)^\ell |x\rangle\|_2}.$$

Then the component of $|w\rangle$ perpendicular to $\mathcal{V}_{1-\varepsilon}$ has length at most ε .

Proof. . .

□

- ▶ The theorem does **not** tell us that $|w\rangle$ converges!
- ▶ It does tell us that after enough iterations $|w\rangle$ is **almost** a combination of the singular vectors with **large** singular values.
- ▶ It assumes that the starting vector $|x\rangle$ has a suitably large component in direction $|v_0\rangle$.

Proof.

Let $A = \sum_{i < r} \sigma_i |u_i\rangle \langle v_i|$ be the SVD of A .
 For convenience complete $\{|v_i\rangle \mid i < r\}$ into an orthonormal basis $\{|v_i\rangle \mid i < d\}$ of d -space.
 Write $|x\rangle = \sum_i \xi_i |v_i\rangle$. Then

$$(A^T A)^\ell |x\rangle = \sum_{i < r} \sigma_i^{4\ell} \xi_i |v_i\rangle.$$

We have $\xi_0 = \langle v_0 | x \rangle$, by hypothesis $|\xi_0| \geq \delta$. Then

$$\left\| (A^T A)^\ell |x\rangle \right\|_2^2 = \sum_{i < r} \sigma_i^{4\ell} \xi_i^2 \geq \sigma_0^{4\ell} \xi_0^2 \geq \sigma_0^{4\ell} \delta^2$$

Suppose m singular values are larger than $(1 - \varepsilon)\sigma_0$,
 ie. $\sigma_{m-1} \geq (1 - \varepsilon)\sigma_0 > \sigma_m$. Now, consider the
 component of $(A^T A)^\ell |x\rangle$ perpendicular to $\mathcal{V}_{1-\varepsilon} =$

$\text{span } \{|v_i\rangle \mid i < m\}$, namely $\sum_{m \leq i < r} \sigma_i^{2\ell} \xi_i |v_i\rangle$. It has squared length

$$\sum_{m \leq i < r} \sigma_i^{4\ell} \xi_i^2 \leq (1 - \varepsilon)^{4\ell} \sigma_0^{4\ell} \underbrace{\sum_{m \leq i < r} \xi_i^2}_{\leq 1} \leq (1 - \varepsilon)^{4\ell} \sigma_0^{4\ell}.$$

Taking the normalization into account, the component of $|w\rangle = \frac{1}{\|(A^T A)^\ell |x\rangle\|_2} (A^T A)^\ell |x\rangle$ perpendicular to $\mathcal{V}_{1-\varepsilon}$ has length at most

$$\frac{(1 - \varepsilon)^{2\ell} \sigma_0^{2\ell}}{\sigma_0^{2\ell} \delta} = \frac{(1 - \varepsilon)^{2\ell}}{\delta} \leq \frac{e^{-2\ell\varepsilon}}{\delta} \leq \varepsilon$$

by the assumption $\ell \geq \frac{-\ln(\varepsilon\delta)}{2\varepsilon}$. □

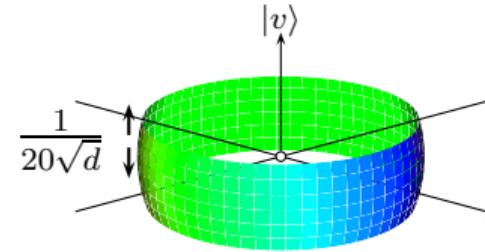


To complete the picture we investigate the probability that a random unit vector has $|\langle v_0 | x \rangle| \geq \delta$.

Lemma

Let $|v\rangle$ be any unit vector in \mathbb{R}^d . Pick $|y\rangle \leftarrow \mathcal{N}_d(0, 1)$ with the unit variance spherical Gaussian probability density. Normalize $|y\rangle \in \mathbb{R}^d$ to get a unit vector by setting $|x\rangle = \frac{|y\rangle}{\| |y\rangle \|_2}$. Then

$$\text{prob}\left(|\langle v | x \rangle| \leq \frac{1}{20\sqrt{d}}\right) \leq \frac{1}{10} \sqrt{\frac{2}{\pi}} + 3e^{-\frac{d}{96}}.$$



Proof. . . .

□

Plugging in numerical values with slightly tighter computations:

- ▶ $d = 95$: $\text{prob}(|\langle v | x \rangle| \leq 0.00514) \leq 0.504$.
- ▶ $d = 200$: $\text{prob}(|\langle v | x \rangle| \leq 0.00354) \leq 0.134$.
- ▶ $d = 200$: $\text{prob}(|\langle v | x \rangle| \leq 0.00714) \leq 0.214$ (using $\frac{2}{10}$ instead of $\frac{1}{10}$).

Proof.

It suffices to show that

1. $\text{prob} \left(\|y\|_2 \geq 2\sqrt{d} \right) \leq 3e^{-\frac{d}{96}}$ and
2. $\text{prob} \left(|\langle v | y \rangle| \leq \frac{1}{10} \right) \leq \frac{1}{10} \sqrt{\frac{2}{\pi}}$.

Namely, if $\|y\|_2 < 2\sqrt{d}$ and $|\langle v | y \rangle| > \frac{1}{10}$ then $|\langle v | x \rangle| > \frac{1}{20\sqrt{d}}$. Contrapositing: if $|\langle v | x \rangle| \leq \frac{1}{20\sqrt{d}}$ then $\|y\|_2 \geq 2\sqrt{d}$ or $|\langle v | y \rangle| \leq \frac{1}{10}$. Thus $\text{prob} \left(|\langle v | x \rangle| \leq \frac{1}{20\sqrt{d}} \right) \leq \text{prob} \left(\|y\|_2 \geq 2\sqrt{d} \right) + \text{prob} \left(|\langle v | y \rangle| \leq \frac{1}{10} \right)$ and that yields the result then:

$$\text{prob} \left(|\langle v | x \rangle| \leq \frac{1}{20\sqrt{d}} \right) \leq \frac{1}{10} \sqrt{\frac{2}{\pi}} + 3e^{-\frac{d}{96}}.$$

(1): follows from the Gaussian Annulus theorem with $\beta = \sqrt{d}$. The annulus then is the ball of radius $2\sqrt{d}$.

(2): follows since $\langle v | y \rangle$ is a random number chosen with zero mean, unit variance Gaussian density. Namely, this density is at most $\frac{1}{\sqrt{2\pi}}$ in the interval $[-\frac{1}{10}, \frac{1}{10}]$ and thus

$$\begin{aligned} \text{prob} \left(\langle v | y \rangle \in \left[-\frac{1}{10}, \frac{1}{10} \right] \right) &= \int_{-\frac{1}{10}}^{\frac{1}{10}} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx \\ &\leq \frac{2}{10} \cdot \frac{1}{\sqrt{2\pi}}. \end{aligned} \quad \square$$

Section overview

Organizational

Introduction

High-dimensional space

Gaussians in high dimensions

Eigenvalues and eigenvectors

Best-fit subspaces and SVD

Power method for SVD

Applications of SVD

Centering data

Unmixing a mixture of spherical Gaussians

Principal component analysis

Ranking documents and web pages

Latent semantic analysis

Hubs and authorities in the WWW

Machine learning

*Clustering

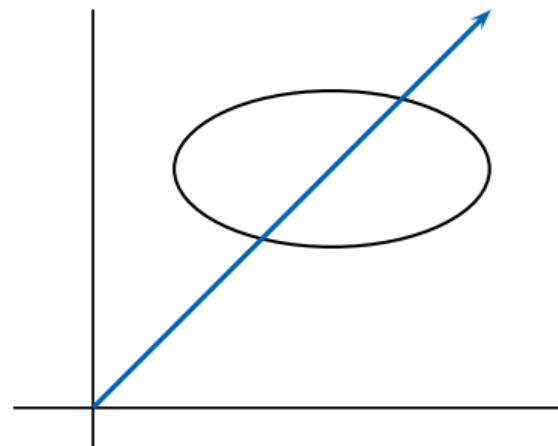
Summary / Outro

Applications of SVD:

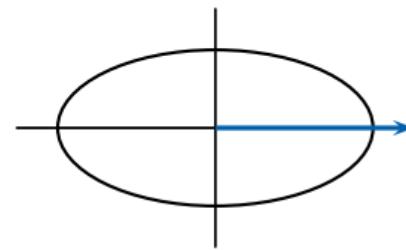
Centering data

Depending on the task you may want to have the best-fit line

- ▶ through the data or
- ▶ through the centered data:



best-fit line



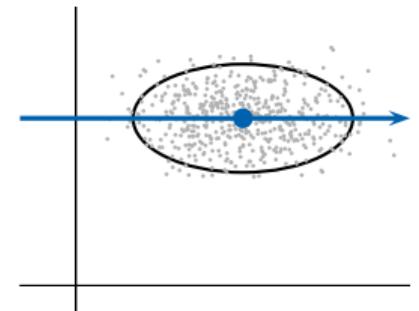
best-fit line through centered
= best-fit affine line

Applications of SVD:

Centering data

Lemma

The best-fit affine line (minimizing the sum of perpendicular squared distances) of a set of data points must pass through the centroid of the points.



Proof. . . .

□

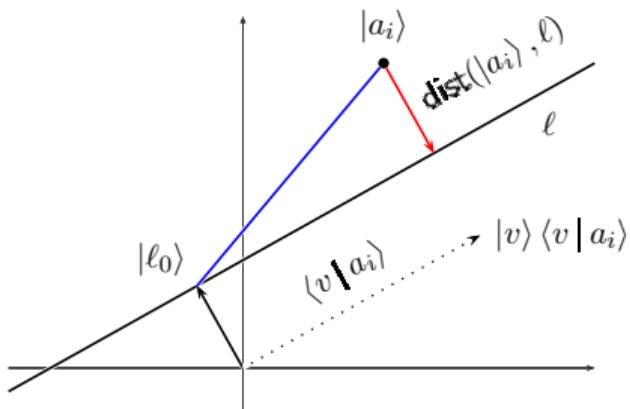
A similar statement holds for k -dimensional best-fit affine spaces.

Proof.

Wlog. the data $\{|a_i\rangle \mid i < n\} \subset (\mathbb{R}^d)^\vee$ has centroid 0. (If your data is given by a matrix A , then each $\langle a_i| = \langle A_{i,\cdot}| = \langle e_i| A$ is a row of the matrix.)

Say $\ell = \{|\ell_0\rangle + \lambda|v\rangle \mid \lambda \in \mathbb{R}\}$ is the best-fit line with $|\ell_0\rangle$ closest to $|0\rangle$ on ℓ and $|v\rangle$ a unit vector. Then $|\ell_0\rangle$ and $|v\rangle$ are perpendicular.

Let $\text{dist}(|a_i\rangle, \ell)$ be the perpendicular distance of $|a_i\rangle$ to ℓ . Then $\| |a_i\rangle - \ell_0 \rangle \|_2^2 = \text{dist}(|a_i\rangle, \ell)^2 + \langle v | a_i \rangle^2$.



Summing over all points:

$$\begin{aligned} & \sum_i \text{dist}(|a_i\rangle, \ell)^2 \\ &= \sum_i \left(\| |a_i\rangle - \ell_0 \rangle \|_2^2 - \langle v | a_i \rangle^2 \right) \\ &= \sum_i \| |a_i\rangle \|_2^2 + n \| \ell_0 \rangle \|_2^2 - 2 \underbrace{\langle \ell_0 | \sum_i |a_i\rangle}_{=0} - \sum_i \langle v | a_i \rangle^2 \\ &= \left(\sum_i \| |a_i\rangle \|_2^2 - \sum_i \langle v | a_i \rangle^2 \right) + n \| \ell_0 \rangle \|_2^2 \end{aligned}$$

This is clearly minimized when $|\ell_0\rangle = 0$ and so the best-fit affine line passes through the centroid. \square

Applications of SVD:

Unmixing a mixture of spherical Gaussians

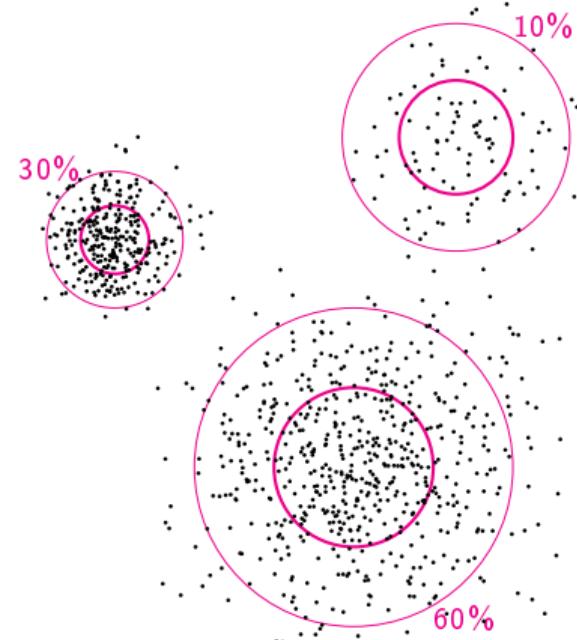
Task (Clustering)

Partition a set of points into k clusters. Each cluster consists of nearby points.

Model fitting problem: Find the parameters for the hidden generation density

$$f = w_0 p_0 + \cdots + w_{k-1} p_{k-1}$$

where p_i is a Gaussian density and $w_i \geq 0$,
 $\sum w_i = 1$.



Applications of SVD:

Unmixing a mixture of spherical Gaussians

Hidden generation density $f = w_0 p_0 + \cdots + w_{k-1} p_{k-1}$.

- Basic densities p_i : known. Spherical Gaussians:

$$\begin{aligned}\mathcal{N}_d\left(\left|\mu^{(i)}\right\rangle, \sigma_i^2\right)(|x\rangle) &= \prod_j \left[\frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(x_j - \mu_j^{(i)})^2}{2\sigma_i^2}\right) \right] \\ &= \frac{1}{(2\pi\sigma_i^2)^{\frac{d}{2}}} \exp\left(-\frac{\| |x\rangle - \mu^{(i)}\|^2}{2\sigma_i^2}\right)\end{aligned}$$

- Parameters for p_i like means $|\mu^{(i)}\rangle$ and variances σ_i^2 : unknown.
- Weights w_i : unknown.

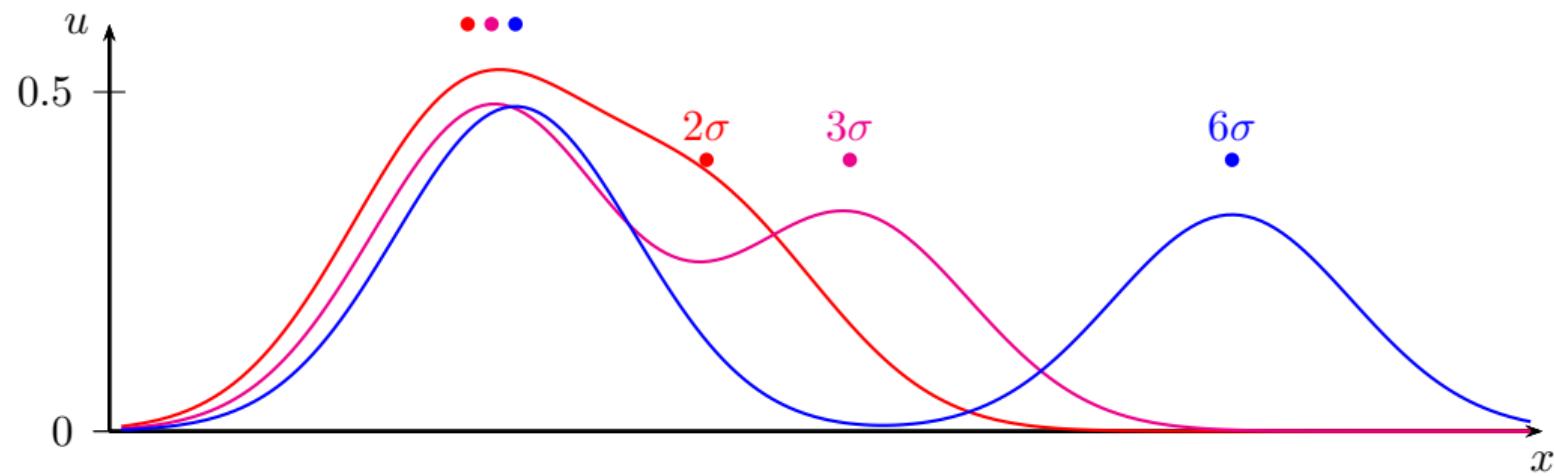
An approach to model fitting

1. Cluster the set of samples into k clusters C_1, \dots, C_k where C_i is the set of samples generated acc.to p_i by the hidden process.
2. Fit a single Gaussian distribution to each cluster.

The second problem is solved by taking the mean and empirical standard deviation as we saw (slide 69).

The first problem is harder.

Separating one-dimensional Gaussians



Intuitively, we need at least 6 standard deviations to separate 1-dimensional Gaussians reliably...

Applications of SVD:

Unmixing a mixture of spherical Gaussians

Separating Gaussians from the sampling angle (compare slide 70)

Instead of looking at nice smooth curves, we have to ask whether random points from the same Gaussian or different Gaussian behave differently wrt. the clustering:

Many clustering methods rely on the distances between points.

- ▶ Consider two samples $|X^{(0)}\rangle, |X^{(1)}\rangle$ from the same spherical Gaussian. Then whp.

$$\left\| |X^{(0)} - X^{(1)}\rangle\right\|_2^2 \approx 2 \left(\sqrt{d} \pm \mathcal{O}(1) \right)^2 \sigma^2.$$

- ▶ Consider samples $|X^{(2)}\rangle, |Y\rangle$ from different spherical Gaussians, each of standard deviation σ and separated by distance Δ . Then whp.

$$\left\| |X^{(2)} - Y\rangle\right\|_2^2 \approx 2 \left(\sqrt{d} \pm \mathcal{O}(1) \right)^2 \sigma^2 + \Delta^2.$$

- ▶ Thus whp.

$$\left\| |X^{(0)} - X^{(1)}\rangle\right\|_2^2 < \left\| |X^{(2)} - Y\rangle\right\|_2^2$$

provided $\frac{\Delta}{\sigma} > c\sqrt[4]{d}$.

Applications of SVD:

Unmixing a mixture of spherical Gaussians

For short:

Claim

For a successful distance-based clustering we need that the distance between the involved Gaussians is at least

$$c\sqrt[4]{d} \cdot \sigma$$

where σ is the largest standard deviation in the set.

This is **bad news** for high dimensions.

Question

Can we improve?

Applications of SVD:

Unmixing a mixture of spherical Gaussians

Assumption

$k \ll d$: The number k of clusters is small compared to the dimension d .

Idea: Concentrate on an 'important' subspace.

- ▶ Suppose we can find the subspace spanned by the k centers.
- ▶ Project all samples to this subspace.
- ▶ Projection of a spherical Gaussian is again a spherical Gaussian.
- ▶ The separation remains the same if the centers are in the subspace.
- ▶ So we only need $\Delta > c\sqrt[4]{k} \cdot \sigma$ which is much easier for $k \ll d$.

Applications of SVD:

Unmixing a mixture of spherical Gaussians

Lemma

Suppose p is a d -dimensional spherical Gaussian with center $|\mu\rangle$ and standard deviation σ . The density of p projected onto a k -subspace V is a spherical Gaussian with the same standard deviation.

Proof. . . .



Proof.

Wlog. the subspace is spanned by the first k unit vectors. [Otherwise rotate the coordinate system appropriately.] Write points $|x\rangle = |x', x''\rangle$ with $|x'\rangle = |(x_0, \dots, x_{k-1})\rangle$ and $|x''\rangle = |(x_k, \dots, x_{d-1})\rangle$. The density of a projected Gaussian $\mathcal{N}_d(|\mu\rangle, \sigma^2)$ is

$$\int_{x''} \mathcal{N}_d(|\mu\rangle, \sigma^2) \, dx'' = ce^{-\frac{1}{2}\left\| \frac{|x' - \mu'}{\sigma} \right\|_2^2} \underbrace{\int_{x''} e^{-\frac{1}{2}\left\| \frac{|x'' - \mu''}{\sigma} \right\|_2^2} \, dx''}_{=c''}.$$

This is $\mathcal{N}_k(|\mu'\rangle, \sigma^2)$.

□

Applications of SVD:

Unmixing a mixture of spherical Gaussians

Lemma

Suppose p is a d -dimensional spherical Gaussian with center $|\mu\rangle$ and standard deviation σ . The density of p projected onto a k -subspace V is a spherical Gaussian with the same standard deviation.

Proof. . . .



Conjecture

The top k singular vectors span the space of the k centers.

0. Redefine 'best-fit' for distributions.
1. The best-fit line to a single spherical Gaussian $\mathcal{N}_d(|\mu\rangle, \sigma^2)$ is the line through its center $|\mu\rangle$ and the origin 0.
2. Any k -subspace containing the line is a best-fit k -subspace for the Gaussian.
3. The best-fit k -subspace for k spherical Gaussians is the subspace containing their centers.

Applications of SVD:

Unmixing a mixture of spherical Gaussians

Definition

If $|X\rangle \xleftarrow{\text{ iid }} p$ with a probability density p in \mathbb{R}^d the best-fit line for $|X\rangle$ is the line in the $|v_0\rangle$ direction where

$$|v_0\rangle = \operatorname*{argmax}_{\| |v\rangle \|_2 = 1} E \left[\langle v | X \rangle^2 \right].$$

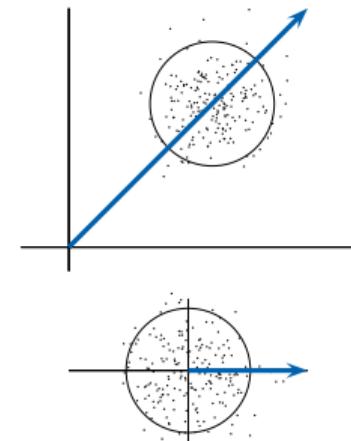
Applications of SVD:

Unmixing a mixture of spherical Gaussians

Lemma

Let the probability density p be a spherical Gaussian with center $|\mu\rangle$.

- ▶ If $|\mu\rangle \neq 0$ the unique best-fit 1-subspace is the line passing through $|\mu\rangle$ and the origin.
- ▶ If $|\mu\rangle = 0$ then any line through the origin is a best-fit line.



Proof. . . .

□

Proof.

Let $|X\rangle \leftarrow \mathcal{N}_d(|\mu\rangle, \sigma^2)$ and $|v\rangle$ be any unit vector. Then

$$\begin{aligned}\mathsf{E}[(\langle v | X \rangle)^2] &= \mathsf{E}[(\langle v | X - \mu \rangle + \langle v | \mu \rangle)^2] \\ &= \underbrace{\mathsf{E}[\langle v | X - \mu \rangle^2]}_{=\sigma^2} + 2 \langle v | \mu \rangle \underbrace{\mathsf{E}[\langle v | X - \mu \rangle]}_{=0} + \langle v | \mu \rangle^2 \\ &= \sigma^2 + \langle v | \mu \rangle^2.\end{aligned}$$

In case $|\mu\rangle \neq 0$ this is maximized when $|v\rangle \propto |\mu\rangle$.

In case $|\mu\rangle = 0$ this is constantly equal to σ^2 . □

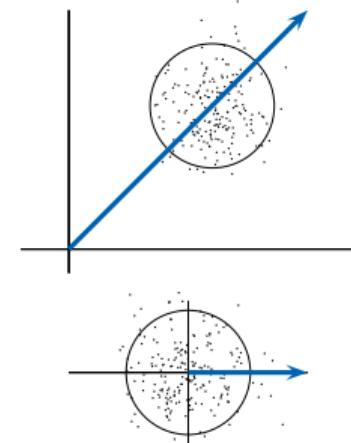
Applications of SVD:

Unmixing a mixture of spherical Gaussians

Lemma

Let the probability density p be a spherical Gaussian with center $|\mu\rangle$.

- ▶ If $|\mu\rangle \neq 0$ the unique best-fit 1-subspace is the line passing through $|\mu\rangle$ and the origin.
- ▶ If $|\mu\rangle = 0$ then any line through the origin is a best-fit line.



Proof. . . .

□

Applications of SVD:

Unmixing a mixture of spherical Gaussians

Definition

If $|X\rangle \xleftarrow{\text{ iid }} p$ with probability density p in \mathbb{R}^d the best-fit k -subspace \mathcal{V}_k for $|X\rangle$ is

$$\mathcal{V}_k = \operatorname*{argmax}_{\substack{\mathcal{V} \\ \dim \mathcal{V}=k}} \mathbb{E} \left[\|\pi_{\mathcal{V}}(|X\rangle)\|_2^2 \right],$$

where $\pi_{\mathcal{V}}$ is the orthogonal projection onto \mathcal{V} .

Note: In case \mathcal{V} is the line in the $|v_0\rangle$ direction, we have $\|\langle v_0 | X \rangle\|_2^2 = \|\pi_{\mathcal{V}}(X)\|$.

Applications of SVD:

Unmixing a mixture of spherical Gaussians

Lemma

For a spherical Gaussian with center $|\mu\rangle$, a k -subspace is a best-fit k -subspace if and only if it contains $|\mu\rangle$.

Proof. . . .



Proof.

Let $|X\rangle \leftarrow \mathcal{N}_d(|\mu\rangle, \sigma^2)$ and \mathcal{V} be a k -subspace. Then

$$\begin{aligned} \mathbb{E} [\|\pi_{\mathcal{V}}|X\rangle\|_2^2] &= \mathbb{E} [\|\pi_{\mathcal{V}}(|X - \mu\rangle) + \pi_{\mathcal{V}}(|\mu\rangle)\|_2^2] \\ &= \underbrace{\mathbb{E} [\|\pi_{\mathcal{V}}(|X - \mu\rangle)\|_2^2]}_{=\sigma^2} + 2\pi_{\mathcal{V}}(\langle\mu|) \underbrace{\mathbb{E} [\pi_{\mathcal{V}}(|X - \mu\rangle)]}_{=0} + \|\pi_{\mathcal{V}}(|\mu\rangle)\|_2^2 \\ &= \sigma^2 + \|\pi_{\mathcal{V}}(|\mu\rangle)\|_2^2. \end{aligned}$$

In case $|\mu\rangle \neq 0$ this is maximized when $|\mu\rangle \in \mathcal{V}$.

In case $|\mu\rangle = 0$ this is constantly equal to σ^2 and $|0\rangle \in \mathcal{V}$ anyways.

□

Applications of SVD:

Unmixing a mixture of spherical Gaussians

Lemma

For a spherical Gaussian with center $|\mu\rangle$, a k -subspace is a best-fit k -subspace if and only if it contains $|\mu\rangle$.

Proof. . . .



Applications of SVD:

Unmixing a mixture of spherical Gaussians

Theorem

If p is a mixture of k spherical Gaussians, then the best-fit k -subspace contains the centers. In particular, if the means of the Gaussians are linearly independent, the space spanned by them is the unique best-fit k -subspace.

Proof. . . .



Proof.

Say $p = \sum_{i < k} w_i p_i$ and p_i is some spherical Gaussian with mean $|\mu^{(i)}\rangle$ and standard deviation σ_i . Consider $|X_i\rangle \leftarrow p_i, i \leftarrow w$ —reading w as a distribution on $\mathbb{N}_{<k}$ — and put $|X\rangle = |X_i\rangle$. Then $|X\rangle \sim p$.

Now we find

$$\mathbb{E} [\|\pi_{\mathcal{V}}(|X\rangle)\|_2] = \sum_i w_i \mathbb{E} [\|\pi_{\mathcal{V}}(|X_i\rangle)\|_2]$$

If \mathcal{V} contains the centers of each density p_i , by the previous Lemma, each term in the summation is individually maximized. That implies the entire summation is maximized and we are done. \square

Applications of SVD:

Unmixing a mixture of spherical Gaussians

Theorem

If p is a mixture of k spherical Gaussians, then the best-fit k -subspace contains the centers. In particular, if the means of the Gaussians are linearly independent, the space spanned by them is the unique best-fit k -subspace.

Proof. . . .



Applications of SVD:

Unmixing a mixture of spherical Gaussians

Intuitively, this settles the task:

- ▶ The theorem tells us that the k subspace containing the k centers of the involved Gaussians is the best-fit k -subspace.
- ▶ The top k singular vectors define the best-fit k -subspace for the given set of rows.

By the law of large numbers, in the limit when the number of samples grows,

all is well.

We skip the technical details...

2006–2009: Netflix launched the Netflix Prize

It was an open contest where the goal was to design state-of-the-art algorithms for predicting movie ratings. During 3 years, research teams developed many different prediction algorithms, among which matrix factorization techniques stood out by their efficiency.

Privacy concerns

Although the data sets were constructed to preserve customer privacy, the Prize has been criticized by privacy advocates. In 2007 two researchers from The University of Texas at Austin were able to identify individual users by matching the data sets with film ratings on the Internet Movie Database.

Applications of SVD:

Principal component analysis

Consider a movie recommendation setting.

- ▶ n customer.
- ▶ d movies.
- ▶ Let A_{cm} represent the amount that customer c likes movie m . (Pretending all these values are given!)

i latent factor. Humans would choose 'action', 'fun' and the like.

$v_{i,m}$ how much movie m is connected with feature i .

$u_{i,c}$ how much weight customer c gives to the feature i .

σ_i importance of the feature.

And now construct its SVD:

$$A = \sum_{i < r} \sigma_i |u_i\rangle \langle v_i|.$$

Applications of SVD:

Principal component analysis

That is just the beginning of the story...

- ▶ Onwards: Cut off the SVD after a few, say k , features and only use $A_k = \sum_{i < k} \sigma_i |u_i\rangle \langle v_i|$. The remainder $A - A_k$ is considered noise.
- ▶ Negative entries in $|u_i\rangle$, $|v_i\rangle$?
- ▶ What about undefined entries A_{ij} ?
 - ▶ Option 1: Fill them with some heuristic. Eg. some sort of mean or chosen randomly acc.to some estimated distribution.

- ▶ Option 2: Notice that the SVD minimizes

$$\sum_{c,m} \left(A_{cm} - \sum_{i < k} \sigma_i u_{i,c} v_{i,m} \right)^2.$$

Now, instead of the full sum, consider only those summands with A_{mc} defined. Minimize that shortened sum but without the orthonormality requirement for ($|u_i\rangle$) and ($|v_j\rangle$).

... called '**collaborative filtering**'.

Applications of SVD:

Ranking documents and web pages

Data set: term-document matrix

	doc1	doc2	doc3	...
term1	1	0	1	...
term2	0	1	0	...
:	:	:	:	

Latent Semantic Analysis,
Latent Semantic Indexing

Candidate definition for 'intrinsic relevance'

Projection onto the best-fit line spanned by
the top left singular vector of the term-
document matrix.

A teaser...

Applications of SVD:

Ranking documents and web pages

Latent semantic analysis²

Given five documents:

- + 1. d_1 : Romeo and Juliet.
- 2. d_2 : Juliet: O happy dagger!
- 1. d_3 : Romeo died by dagger.
- 2. d_4 : "Live free or die", that's the New-Hampshire's motto.
- 5. d_5 : Did you know, New-Hampshire is in New-England.

and a search query **die, dagger**.

Question

What should be in the result?

- ▶ d_3 contains both.
- ▶ d_2, d_4 contain one word each.
- ▶ d_1 : no word. But connected via Shakespeare's play, represented in our documents d_2, d_3 .
- ▶ d_5 : no word. But connected via d_4 .

Question

Can the machine deduce this?

²From Alex Thomo (2009). Latent Semantic Analysis (Tutorial).

Applications of SVD:

Ranking documents and web pages

Latent semantic analysis²

Given five documents:

- + 1. d_1 : Romeo and Juliet.
- 2. d_2 : Juliet: O happy dagger!
- 1. d_3 : Romeo died by dagger.
- 2. d_4 : "Live free or die", that's the New-Hampshire's motto.
- 5. d_5 : Did you know, New-Hampshire is in New-England.

Define the term-document matrix A :

	d_1	d_2	d_3	d_4	d_5
romeo	1	0	1	0	0
juliet	1	1	0	0	0
happy	0	1	0	0	0
dagger	0	1	1	0	0
live	0	0	0	1	0
die	0	0	1	1	0
free	0	0	0	1	0
new-hampshire	0	0	0	1	1

and a search query die, dagger.

²From Alex Thomo (2009). Latent Semantic Analysis (Tutorial).

Applications of SVD:

Ranking documents and web pages

Latent semantic analysis²

Given five documents:

- + d_1 : Romeo and Juliet.
- 2. d_2 : Juliet: O happy dagger!
- 1. d_3 : Romeo died by dagger.
- 2. d_4 : "Live free or die", that's the New-Hampshire's motto.
- 5. d_5 : Did you know, New-Hampshire is in New-England.

and a search query **die, dagger**.

Singular values of A : [2.29ʌ, 2.01ʌ, 1.36ʌ, 1.12ʌ, 0.80ʌ]. Keep only largest two, consider $A_2 = U_2 D_2 V_2^T$.

$$U_2 = \begin{bmatrix} \text{romeo} & 0.40\text{ʌ} & 0.28\text{l} \\ \text{juliet} & 0.31\text{ʌ} & 0.45\text{T} \\ \text{happy} & 0.18\text{T} & 0.27\text{T} \\ \text{dagger} & 0.44\text{T} & 0.37\text{T} \\ \text{live} & 0.26\text{ʌ} & -0.35\text{ʌ} \\ \text{die} & 0.52\text{ʌ} & -0.25\text{ʌ} \\ \text{free} & 0.26\text{ʌ} & -0.35\text{ʌ} \\ \text{new-hampshire} & 0.33\text{ʌ} & -0.46\text{T} \end{bmatrix}, \quad V_2 = \begin{bmatrix} d_1 & 0.31\text{l} & 0.36\text{ʌ} \\ d_2 & 0.41\text{T} & 0.54\text{l} \\ d_4 & 0.59\text{ʌ} & 0.20\text{l} \\ d_5 & 0.60\text{ʌ} & -0.70\text{ʌ} \\ d_3 & 0.14\text{ʌ} & -0.23\text{T} \end{bmatrix}$$

²From Alex Thomo (2009). Latent Semantic Analysis (Tutorial).

Applications of SVD:

Ranking documents and web pages

Latent semantic analysis²

Given five documents:

- + 1. d_1 : Romeo and Juliet.
- 2. d_2 : Juliet: O happy dagger!
- 1. d_3 : Romeo died by dagger.
- 2. d_4 : "Live free or die", that's the New-Hampshire's motto.
- 5. d_5 : Did you know, New-Hampshire is in New-England.

and a search query **die, dagger**.

Scale by singular values and represent terms by rows of $U_2 D_2$ and documents by columns of $D_2 V_2^T$.

$$U_2 D_2 = \begin{bmatrix} \text{romeo} & 0.91\downarrow & 0.56\downarrow \\ \text{juliet} & 0.72\uparrow & 0.90\downarrow \\ \text{happy} & 0.41\uparrow & 0.54\downarrow \\ \text{dagger} & 1.00\downarrow & 0.74\downarrow \\ \text{live} & 0.60\downarrow & -0.70\downarrow \\ \text{die} & 1.20\uparrow & -0.50\downarrow \\ \text{free} & 0.60\downarrow & -0.70\downarrow \\ \text{new-hampshire} & 0.75\downarrow & -0.92\downarrow \end{bmatrix}, \quad V_2 D_2 = \begin{bmatrix} d_1 & 0.71\downarrow & 0.73\uparrow \\ d_2 & 0.93\downarrow & 1.09\downarrow \\ d_4 & 1.36\uparrow & 0.40\downarrow \\ d_5 & 1.38\uparrow & -1.40\uparrow \\ d_3 & 0.33\downarrow & -0.46\uparrow \end{bmatrix}$$

²From Alex Thomo (2009). Latent Semantic Analysis (Tutorial).

Applications of SVD:

Ranking documents and web pages

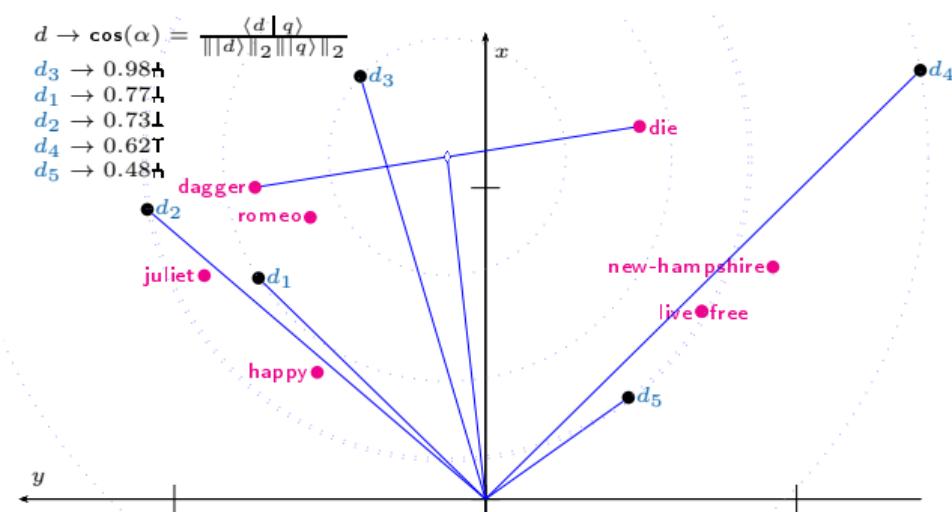
Latent semantic analysis²

Given five documents:

- + d_1 : Romeo and Juliet.
- 2. d_2 : Juliet: O happy dagger!
- 1. d_3 : Romeo died by dagger.
- 2. d_4 : "Live free or die", that's the New-Hampshire's motto.
- 5. d_5 : Did you know, New-Hampshire is in New-England.

and a search query **die, dagger**.

Interpret, add query $q = \text{centroid}(\text{die}, \text{dagger})$ and sort by angle:



²From Alex Thomo (2009). Latent Semantic Analysis (Tutorial).

Applications of SVD:

Ranking documents and web pages

Latent semantic analysis²

Given five documents:

- + d_1 : Romeo and Juliet.
- 2. d_2 : Juliet: O happy dagger!
- 1. d_3 : Romeo died by dagger.
- 2. d_4 : "Live free or die", that's the New-Hampshire's motto.
- 5. d_5 : Did you know,
New-Hampshire is in
New-England.

and a search query **die, dagger**.

Question

Can the machine deduce this?

Answer

- ▶ This procedure does give the desired ranking.
- ▶ There are a bunch of design decisions:
 - ▶ Why take first two singular values?
 - ▶ Why use the scaled versions for the picture?
 - ▶ Why consider the 'cosine measure', ie. the angle?
- ▶ Justification? Beyond "it works"?

²From Alex Thomo (2009). Latent Semantic Analysis (Tutorial).

Latent concepts²

- ▶ Latent Semantic Indexing (LSI) is a method for discovering **hidden concepts** in document data.
- ▶ Each document and term (word) is then expressed as a vector with elements corresponding to these concepts. Each element in a vector gives the degree of participation of the document or term in the corresponding concept.
- ▶ Goal is not to describe the concepts verbally, but to be able to represent the documents and terms in a unified way for exposing document-document, document-term, and term-term similarities which are **otherwise hidden**...

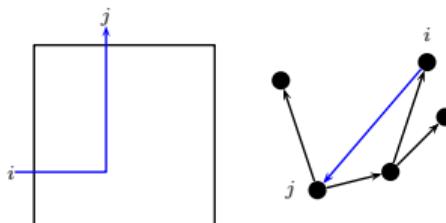
² From Alex Thomo (2009). Latent Semantic Analysis (Tutorial).

Applications of SVD:

Ranking documents and web pages

Hubs and authorities in the WWW

We consider each web page as a node. And we represent a link as an edge from the referer to the target. Let A be the adjacency matrix, ie. $A_{i,j} = 1$ iff there is an edge from i to j .



Authority most prominent sources on a given topic. Many hubs point to it.

Hub identify authorities on a topic. Points to many authorities.

Approach

Try to find

- ▶ a hub weight vector $|u\rangle \in \mathbb{R}^n$ and
- ▶ an authority weight vector $|v\rangle \in \mathbb{R}^n$.

Given the hub vector, the authority vector should be computed by

$$|v\rangle \propto A^T |u\rangle$$

as hubs point to authorities. The weight of authority j thus is proportional to the combination of the weights of hubs pointing to it.

Vice versa, the hub vector should be defined by the authority vector similarly as

$$|u\rangle \propto A |v\rangle .$$

Using these connections one can try to start with a random hub vector and just apply A^T , then A , and so on until this converges. As we have seen it does converge and the limit is a top singular vector.

...

Section overview

Organizational

Introduction

High-dimensional space

Gaussians in high dimensions

Eigenvalues and eigenvectors

Best-fit subspaces and SVD

Power method for SVD

Applications of SVD

Machine learning

Introduction

The perceptron algorithm

Kernel functions and non-linearly separable
data

Generalizing to new data

VC-dimension

VC-dimension and generalizing

*Deep learning

Gradient descent

Stochastic gradient descent

Tweaks

*Online learning

Online to batch conversion

*Expert advice

*Boosting

*Further current directions

*Clustering

Summary / Outro

Examples (Evaluate!)

- ▶ Learn to detect spam emails.
- ▶ Learn to detect cats in images.
- ⚡? Picture Editing apps.
- ▶ Speech recognition.
- ▶ Learn to answer questions about Shakespeare's work.
- ⚡? Virtual Personal Assistants.
- ▶ Predictions while Commuting.
- ⚡? Videos Surveillance.
- ⚡? Online Customer Support.
- ⚡? Search Engine Result Refining.
 - ▶ Marine Wildlife Preservation.
 - ▶ Predict potential heart failure.
-  Driverless cars.
-  AlphaGo.
- ▶ ...

Task

Suppose we are given *training data*, consisting of a list of data items each with a label. We want to predict the correct label for future data items.

In other words, the basic goal is to *generalize* a given classification.

This is done by *finding* a classification rule within a predetermined *set of rules* that somehow fits best to the training *data*.

To be specified:

- ▶ type of data,
- ▶ allowed rules (and how to describe them),
- ▶ definition for 'best-fit'.

Then devise an algorithm for finding or approximating a best-fit rule.

Example

- ▶ Data: Set of emails, each as a list of words.
- ▶ Label: Binary, eg. spam or not spam.

How to classify?

- ▶ We may use a list of indicative words.
- ▶ Some saying 'looks like spam'.
- ▶ Others saying 'look like non-spam'.

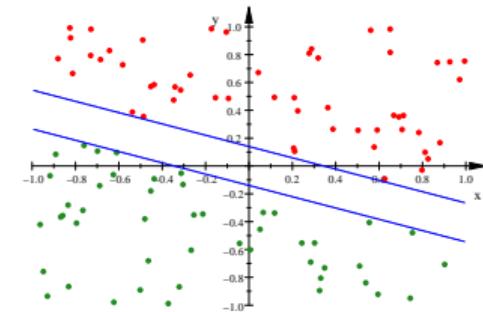
We might consider weighted frequency of such words.

How to find those words?

How to find the weights?

Task

- ▶ Each data item $|x_i\rangle$ is a point in \mathbb{R}^d .
- ▶ Each training item comes with a label $\ell_i \in \{-1, +1\}$.
- ▶ Each rule is a linear separator: $|w\rangle \in \mathbb{R}^d$ splitting the space into two parts



$$\langle w | + x \rangle \geq 1, \quad \langle w | - x \rangle \geq 1$$

and a separating corridor.

- ▶ Find any valid separator with

$$\langle w | \cdot \ell_i | x_i \rangle \geq +1 \quad \text{for each } i.$$

The perceptron algorithm

1. $|w\rangle \leftarrow 0$.
2. **While** there exists $|x_i\rangle$ with $\langle w| \cdot \ell_i |x_i\rangle \leq 0$ **do** 3–3
3. $|w\rangle \leftarrow |w\rangle + \ell_i |x_i\rangle$.

This is based on the following intuition:

- ▶ When updating $|w\rangle$ we have

$$(\langle w| + \ell_i \langle x_i|) \cdot \ell_i |x_i\rangle = \langle w| \cdot \ell_i |x_i\rangle + \||x_i\rangle\|_2^2$$

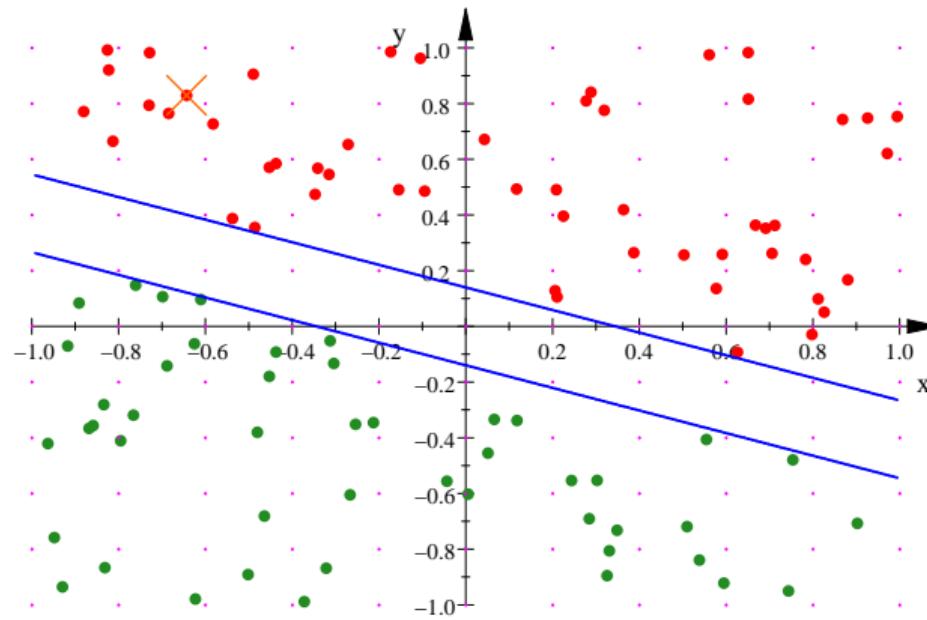
larger than $\langle w| \cdot \ell_i |x_i\rangle$.

- ▶ Of course, what is good for $|x_i\rangle$ needs not be good for other $|x_j\rangle$.
- ▶ It turns out that this approach **does work** and **the better the wider the corridor**.

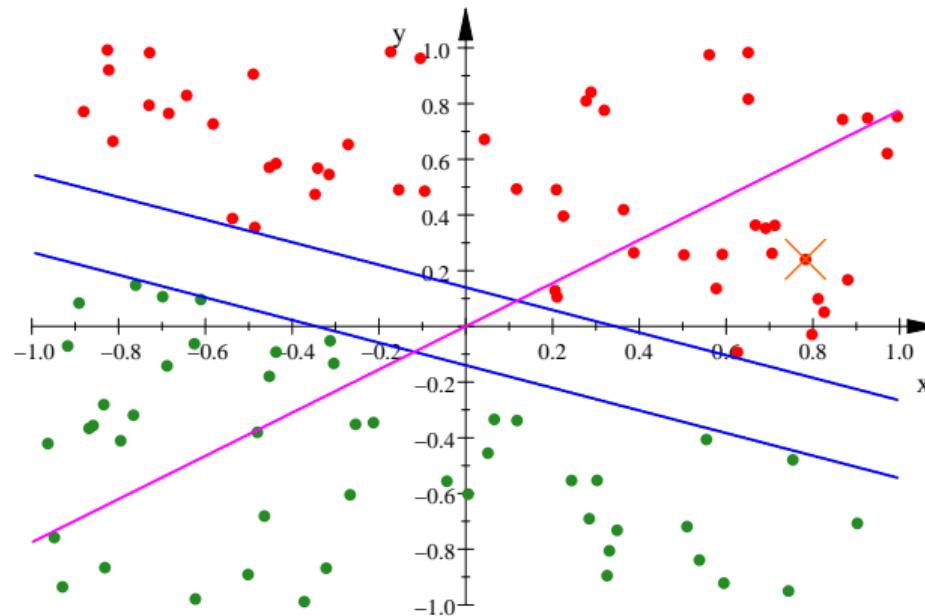
Machine learning:

The perceptron algorithm

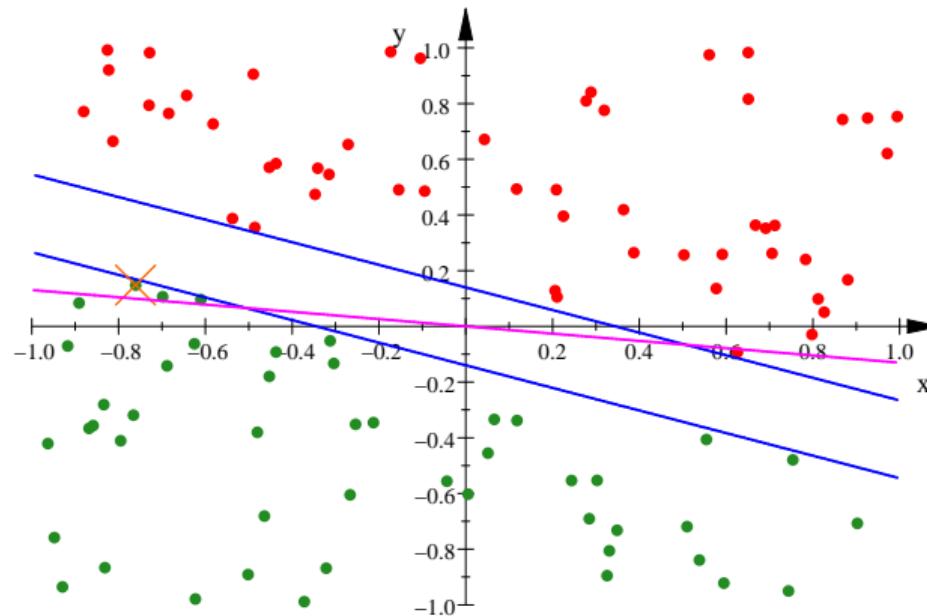
Example



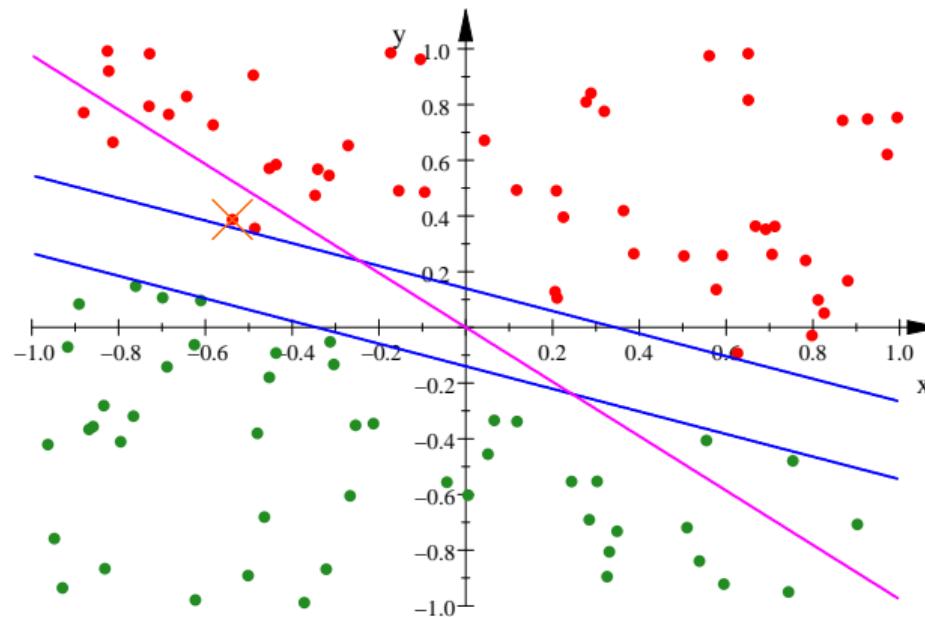
Example



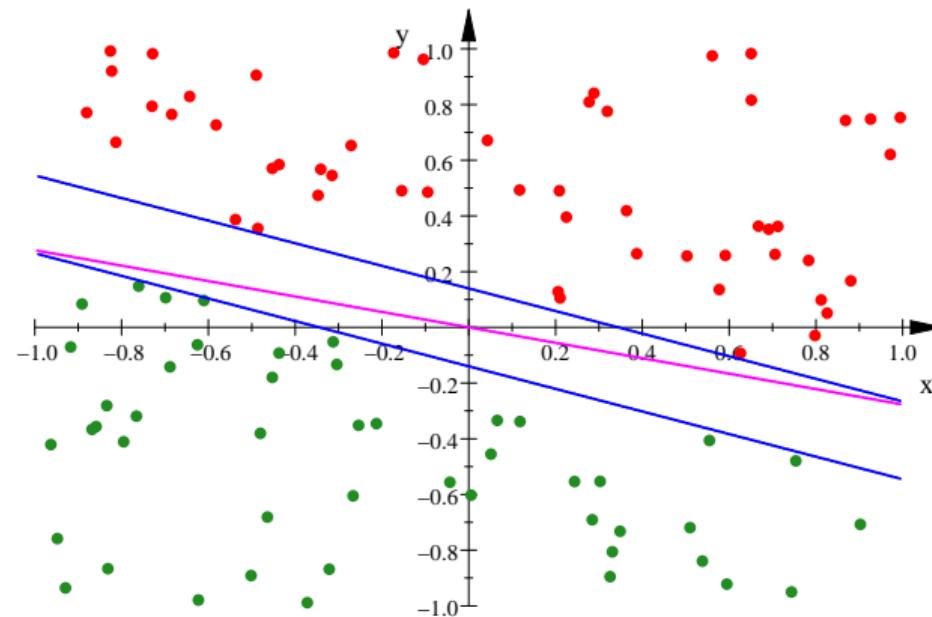
Example



Example



Example



Theorem

Suppose there is a $|w^*\rangle$ satisfying the inequalities $\langle w^* | \cdot \ell_i | x_i \rangle \geq 1$ for each i . Then, the Perceptron Algorithm finds a solution $|w\rangle$ with $\langle w | \cdot \ell_i | x_i \rangle > 0$ for each i in at most

$$r^2 \| |w^* \rangle \|_2^2 \text{ updates}$$

where $r = \max_i \| |x_i \rangle \|_2$.

Proof. . . .

□

For the previous example $\| |w^* \rangle \|_2 = 7.74$ and $r = 1.34$. Thus $r^2 \| |w^* \rangle \|_2^2 = 104.4$. Still, it only needed 4 updates which was even an untypically slow run.

Proof.

Let $\langle w^* |$ satisfy $\langle w^* | \cdot \ell_i |x_i\rangle \geq 1$ for each i .

We are going to keep track of the two quantities

- ▶ $\langle w^* | w \rangle$ and
- ▶ $\|w\|_2^2$.

Consider an update $|w\rangle \leftarrow |w\rangle + \ell_i |x_i\rangle$
with $\langle w | \cdot \ell_i |x_i\rangle \leq 0$.

It increases $\langle w^* | w \rangle$ by at least one:

$$\begin{aligned}\langle w^* | \cdot (|w\rangle + \ell_i |x_i\rangle) &= \langle w^* | w \rangle + \langle w^* | \cdot \ell_i |x_i\rangle \\ &\geq \langle w^* | w \rangle + 1.\end{aligned}$$

On the other hand $\|w\|_2^2$ increases by at most r^2 :

$$\begin{aligned}\|w\rangle + \ell_i |x_i\rangle\|_2^2 &= \|w\|_2^2 + 2 \underbrace{\langle w | \cdot \ell_i |x_i\rangle}_{\leq 0} + \underbrace{\|\ell_i |x_i\rangle\|_2^2}_{\leq r^2}\end{aligned}$$

$$\leq \|w\|_2^2 + r^2.$$

This relies on $\langle w | \cdot \ell_i |x_i\rangle \leq 0$ enabling an update.

Now after m updates

- ▶ $\langle w^* | w \rangle \geq m$ and
- ▶ $\|w\|_2^2 \leq mr^2$.

Thus

$$\begin{aligned}m^2 &\leq \langle w^* | w \rangle^2 \\ &\stackrel{\text{Cauchy-Schwarz}}{\leq} \|\langle w^* |\|_2^2 \|w\|_2^2 \\ &\leq mr^2 \|\langle w^* |\|_2^2\end{aligned}$$

and so the number of updates is bounded:

$$m \leq r^2 \|\langle w^* |\|_2^2.$$

□

Example



- ▶ Clearly, this is not linearly separable.
- ▶ But apply

$$\varphi: \begin{aligned} \mathbb{R}^2 &\longrightarrow \mathbb{R}^3, \\ |x, y\rangle &\longmapsto |x, y, x^2 + y^2\rangle . \end{aligned}$$

- ▶ Now, it is linearly separable since the inner circle points will have $z \approx 1$ and the outer ones $z \approx 4$.
- ▶ The perceptron algorithm can separate the modified data.

How to find φ ?

Well, . . . do not even try.

- ▶ The perceptron algorithm used after applying φ will produce a weight vector $|w\rangle$ by summing $\ell_i |\varphi(x_i)\rangle$. Thus once done we have

$$|w\rangle = \sum_i c_i |\varphi(x_i)\rangle.$$

- ▶ And for the classification we use $\langle w | \varphi(x_j) \rangle$, ie.

$$\langle w | \varphi(x_j) \rangle = \sum_i c_i \underbrace{\langle \varphi(x_i) | \varphi(x_j) \rangle}_{=K_{ij}=k(|x_i\rangle, |x_j\rangle)}.$$

- ▶ To compute this quantity we do not need φ but only its **kernel matrix** $K \in \mathbb{R}^{n \times n}$ or its **kernel function** $k: \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$.

The perceptron algorithm with a kernel function

1. $|c\rangle \leftarrow 0.$
2. **While** there exists $|x_i\rangle$ with $\sum_j c_j k(|x_j\rangle, |x_i\rangle) \ell_i \leq 0$ **do** 3–3
3. $c_i \leftarrow c_i + \ell_i.$

Clearly, you ‘only’ need the kernel matrix $K = [k(|x_j\rangle, |x_i\rangle)]_{ij}$ here.

Remark

- ▶ The kernel $k(x, y) = \langle x | y \rangle$ (dot product) yields exactly the original perceptron algorithm with linear separators.
- ▶ The kernel $k(x, y) = 1 + \langle x | y \rangle$ yields the variant with affine separators as in the technical modification. [Try out!]

Which matrix K can be a kernel matrix?

Lemma

*A matrix K is a **kernel matrix**, ie. there is a function φ such that $K_{ij} = \langle \varphi(x_i) | \varphi(x_j) \rangle$, if and only if K is positive semidefinite.*

Proof. ...



Proof.

\Leftarrow : If K is positive semidefinite, then $K = BB^T$.
Let $\varphi(|x_i\rangle)$ be the i -th column of B^T . Then $K_{ij} = \langle \varphi(x_i) | \varphi(x_j) \rangle$.

\Rightarrow : Given a map φ just construct B^T with i -th column $\varphi(|x_i\rangle)$. Then $K = BB^T$ and so K is positive semidefinite. \square

Definition

A **kernel function** $k: \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$, $(|x\rangle, |y\rangle) \mapsto k(|x\rangle, |y\rangle)$ is any such function with $k(|x\rangle, |y\rangle) = \langle \varphi(x) | \varphi(y) \rangle$ for some map $\varphi: \mathbb{R}^d \rightarrow \mathbb{R}^n$.

Then a kernel matrix K is a kernel function k applied to the data: $K_{ij} = k(|x_i\rangle, |x_j\rangle)$.

Combining kernel functions

Theorem

Suppose k_1 and k_2 are kernel functions. Then:

1. The constantly scaled function ck_1 is a legal kernel function for any constant $c \geq 0$. is a legal kernel function for any scalar function $f: \mathbb{R}^d \rightarrow \mathbb{R}$.
2. The scaled function k_3 with 3. The sum $k_1 + k_2$ is a legal kernel.
$$k_3(|x\rangle, |y\rangle) = f(|x\rangle)f(|y\rangle)k_1(|x\rangle, |y\rangle)$$
4. The product $k_1 \cdot k_2$ is a legal kernel.

Proof. . .



Popular kernels

- ▶ $k(|x\rangle, |y\rangle) = (1 + \langle x | y \rangle)^a$ is a legal kernel function for $a \in \mathbb{N}$.
- ▶ The Gaussian kernel $k(|x\rangle, |y\rangle) = e^{-c\|x-y\|_2^2}$ is a legal kernel function for $c \geq 0$.

Remark

Beware about the separation condition!

It rules the runtime of the perceptron algorithm...

...and we have not translated it to the kernel view so far.

Effect of a kernel

To understand which kind of separation a kernel can produce, examine the separation condition:

- ▶ $\left\{ |x\rangle \mid \sum_j c_j k(|x_j\rangle, |x\rangle) \ell \geq 1 \right\}$: Label ℓ area.
- ▶ $\left\{ |x\rangle \mid \sum_j c_j k(|x_j\rangle, |x\rangle) = 0 \right\}$: Separator.

Examples, say in \mathbb{R}^2

- ▶ $k(|w\rangle, |x\rangle) = \langle w | x \rangle$: Each separator is a line through the origin. ... an arbitrary quadratic curve, ie. a line or an ellipse, parabola or hyperbola.
- ▶ $k(|w\rangle, |x\rangle) = 1 + \langle w | x \rangle$: ... an arbitrary affine line. ▶ $k(|w\rangle, |x\rangle) = e^{-c\|x-w\|^2}$: ... an analytical curve. The levels $\sum c_j k(|x_j\rangle, |x\rangle)$ describe the hilly landscape with Gaussian hills around each $|x_j\rangle$ of height c_j .
- ▶ $k(|w\rangle, |x\rangle) = 1 + \langle w | x \rangle + \langle w | x \rangle^2$ or $k(|w\rangle, |x\rangle) = (1 + \langle w | x \rangle)^2$:

So far we have focussed on describing given labelled data.

Next step:

Generalizing to new data

A primer on prophecies

How does the sequence 2, 3, 5, ?, ? continue?

- ▶ Your guess is 7, 11? **Why?** Primes?
- ▶ For a child 6, 8 would be natural... Numbers with round parts.
- ▶ Algebraically, 8, 12 is most natural... Values of $\frac{x^2+x+4}{2}$.
- ▶ Recursion makes 8, 13 natural. Add previous two to get next element.

Fact

Maths: no finite part of a sequence can tell you the 'future'.

... unless you restrict the allowed explanations!

So far...

focus on finding an algorithm that performs well on training data.

But:

Want good predictions!

So we need to make some extra assumptions:

- ▶ Our data is sampled from some distribution.
- ▶ Occam: A simpler rule is better.

Formalizing

- ▶ Instance space Ω with a distribution D .
- ▶ Instances are classified by a target concept c^* , $c^* \subseteq \Omega$.
- ▶ Training set S drawn independently from D .
- ▶ We are given the classification ' $s \in c^*$ ' or ' $s \notin c^*$ ' for each $s \in S$.
- ▶ Objective: Predict well on new points drawn from D .
- ▶ Goal: Find a hypothesis h , $h \subseteq \Omega$, close to c^* wrt. D .
- ▶ True error $\text{err}_D(h) := \text{prob}(Z \in h \Delta c^* \mid Z \xleftarrow{D} D)$.
[Inaccessible in practice!]
- ▶ Training error $\text{err}_S(h) := \frac{1}{\#S} \#(S \cap (h \Delta c^*))$.

Inherent problem: We may have

- ▶ small training error $\text{err}_S(h)$ and
- ▶ large true error $\text{err}_D(h)$.

For example, putting $h \leftarrow S \cap c^*$ even $\text{err}_S(h) = 0$.



- ▶ Hypothesis class \mathcal{H} over Ω is a collection of subsets of Ω .

Examples:

- ▶ Intervals over $\Omega = \mathbb{R}$: $\mathcal{H} = \{[a, b] \mid a < b\}$.
- ▶ Linear separators over $\Omega = \mathbb{R}^d$: $\mathcal{H} = \{\{x \in \mathbb{R}^d \mid \langle w \mid x \rangle \geq 0\} \mid w \in \mathbb{R}^d\}$.
- ▶ Affine separators over $\Omega = \left\{ \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\}$:
 $\mathcal{H} = \{\{x \in \Omega \mid \langle w \mid x \rangle \geq t\} \mid w \in \mathbb{R}^d, t \in \mathbb{R}\}$.
Here, $\#\mathcal{H} = 14$, so two subsets of X are missing.

We will argue that

- ▶ if S is large enough compared to some property of \mathcal{H} then
- ▶ with high probability all $h \in \mathcal{H}$ have true error close to training error.

Theorem (First sample bound)

Let \mathcal{H} be a (finite) hypothesis class and let ε and δ be greater than zero. If a training set S of size

$$n \geq \frac{1}{\varepsilon} \ln \frac{\#\mathcal{H}}{\delta}$$

is drawn from distribution D , then with probability greater than or equal to $1 - \delta$ each $h \in \mathcal{H}$ with true error $\text{err}_D(h) \geq \varepsilon$ has training error $\text{err}_S(h) > 0$.

Equivalently, with probability greater than or equal to $1 - \delta$, each $h \in \mathcal{H}$ with training error zero has true error less than ε .

Proof. . .

It's a **PAC-learning guarantee**: The theorem states that a hypothesis consistent with the sample is probably approximately correct.

□

Proof.

Consider each $h \in \mathcal{H}$ with true error $\geq \varepsilon$.

Let $X_j \leftarrow \{0,1\}$ be the random variable indicating whether h makes a mistake on the j -th example in S .

Now, consider the event that h has training error 0:

$$\begin{aligned}\text{prob}(\text{err}_S(h) = 0) &= \text{prob}(\forall j: X_j = 0) \\ &= \prod_j \underbrace{\text{prob}(X_j = 0)}_{1 - \text{err}_D(h)} \\ &\leq (1 - \varepsilon)^n \leq e^{-\varepsilon n}.\end{aligned}$$

Then

$$\begin{aligned}&\text{prob}(\exists h \in \mathcal{H}: \text{err}_D(h) \geq \varepsilon \wedge \text{err}_S(h) = 0) \\ &= \text{prob}\left(\bigvee_{\substack{h \in \mathcal{H} \\ \text{err}_D(h) \geq \varepsilon}} \text{err}_S(h) = 0\right) \\ &\leq \#\mathcal{H} \cdot e^{-\varepsilon n} \\ &\leq \#\mathcal{H} \cdot e^{-\varepsilon \cdot \frac{1}{\varepsilon} \ln \frac{\#\mathcal{H}}{\delta}} = \delta.\end{aligned}$$

□

What if the training error is only small, rather than zero?

- ▶ What if the best h in \mathcal{H} has 5% error on S ?
- ▶ Can we still be confident that its true error is low, say at most 10%?

We are actually asking that for all hypothesis $h \in \mathcal{H}$ in our class, the training errors converges to the true error *uniformly*.

For getting on, we need a stronger tail bound.

Theorem (Hoeffding bounds)

Let $X_0, X_1, \dots, X_{n-1} \xleftarrow{\text{IID}} \{0, 1\}$ be independent $\{0, 1\}$ -valued random variables with $\text{prob}(X_i = 1) = p$, ie. $E(X_i) = p$. Then for $\varepsilon \in [0, 1]$

$$\text{prob} \left(\frac{1}{n} \sum_{i < n} X_i > p + \varepsilon \right) \leq e^{-2n\varepsilon^2},$$

$$\text{prob} \left(\frac{1}{n} \sum_{i < n} X_i < p - \varepsilon \right) \leq e^{-2n\varepsilon^2}.$$

□

This is a Chernoff type bound. It is another large deviation result.

With this tail bound we can improve our PAC-learning guarantee as follows.

Theorem (Uniform convergence)

Let \mathcal{H} be a hypothesis class and $\varepsilon > 0, \delta > 0$.

If a training set S of size

$$n \geq \frac{1}{2\varepsilon^2} \ln \frac{2\#\mathcal{H}}{\delta}$$

is drawn from distribution D , then with probability greater than or equal to $1 - \delta$, each $h \in \mathcal{H}$ satisfies $|\text{err}_D(h) - \text{err}_S(h)| \leq \varepsilon$.

Proof. . .



Proof.

Fix $h \in \mathcal{H}$.

Let $X_j \leftarrow \{0,1\}$ be the random variable indicating whether h makes a mistake on the j -th example in S . Clearly, (X_j) is independent and $\text{prob}(X_j = 1) = \text{err}_D(h)$.

Further, $\text{err}_S(h) = \frac{1}{n} \sum_j X_j$.

And, $\mathbb{E} \text{err}_S(h) = \frac{1}{n} \sum_j \mathbb{E} X_j = \text{err}_D(h)$.

Applying the Hoeffding bound yields

$$\text{prob}(|\text{err}_S(h) - \text{err}_D(h)| > \varepsilon) \leq 2e^{-2n\varepsilon^2}.$$

Summing over all h , the probability that for some h the claim is violated is at most

$$\#\mathcal{H} \cdot 2e^{-2n\varepsilon^2} \leq \delta$$

using the bound on n , namely $n \geq \frac{1}{2\varepsilon^2} \ln \frac{2\#\mathcal{H}}{\delta}$. □

Regularization, penalizing complexity

Say...

- ▶ There is no simple rule that is perfectly consistent with the training data.
- ▶ Some very simple rule achieves training error 20%.
- ▶ Some more complex rule achieves training error 10%.
- ▶ ...

Consequence

Instead of optimizing training error only, optimize a combination of training error and simplicity.

A regularizer is a penalty term that penalizes more complex hypotheses.

Corollary

Fix any description language and a training sample S drawn from distribution D . With probability greater than or equal to $1 - \delta$, all hypotheses h satisfy

$$|\text{err}_D(h) - \text{err}_S(h)| \leq \sqrt{\frac{\text{size}(h) \cdot \ln 2 + \frac{1}{2} \ln \frac{2}{\delta}}{\#S}}$$

where $\text{size}(h)$ denotes the number of bits needed to describe h in the given language.

Proof. . .



Proof.

Let \mathcal{H}_i be those hypotheses that can be described in i bits in the description language, ie. $i = \text{size}(h)$.

Clearly, $\#\mathcal{H}_i \leq 2^i$. Put $\delta_i = \delta 2^{-i}$. Notice that $\mathcal{H} = \bigcup_i \mathcal{H}_i$ and $\delta = \sum_i \delta_i$.

Now, the uniform convergence theorem tells us that for $h \in \mathcal{H}_i$ with probability at least $1 - \delta_i$

$$|\text{err}_D(h) - \text{err}_S(h)| \leq \sqrt{\frac{\ln \frac{2\#\mathcal{H}_i}{\delta_i}}{2\#S}}$$

$$\begin{aligned} &\leq \sqrt{\frac{\ln \frac{2 \cdot 2^i}{\delta \cdot 2^{-i}}}{2\#S}} \\ &= \sqrt{\frac{\text{size}(h) \ln 4 + \ln \frac{2}{\delta}}{2\#S}}. \end{aligned}$$

The union bound (applied to the complements) implies that with probability at least $1 - \delta$ we have this for each $h \in \mathcal{H}$. □

Based on this corollary to grant a small true error $\text{err}_D(h)$ we can search for a rule with a small combined value for

- ▶ the training error

$$\text{err}_S(h)$$

and

- ▶ the penalty term

$$\sqrt{\frac{\text{size}(h) \cdot \ln 2 + \frac{1}{2} \ln \frac{2}{\delta}}{\#S}}.$$

Notice also that the hypothesis class \mathcal{H} needs not to be finite any more.

Summary

- ▶ **First sample bound:** If $\#S$ is large compared to $\frac{1}{\varepsilon} \ln \frac{\#\mathcal{H}}{\delta}$ then w.h.p. each $h \in \mathcal{H}$ with true error at least ε will have training error greater than zero.
- ▶ **Uniform convergence:** If $\#S$ is large compared to $\frac{1}{2\varepsilon^2} \ln \frac{2\#\mathcal{H}}{\delta}$ then w.h.p. for each $h \in \mathcal{H}$ true error and training error differ by at most ε .
- ▶ **Corollary:** If $\#S$ is large and we look for a suitably small h such that the penalty term is small, ie.

$$\text{size}(h) \cdot \ln 2 + \frac{1}{2} \ln \frac{2}{\delta} \leq \varepsilon^2 \#S,$$

then w.h.p. for each $h \in \mathcal{H}$ true error and training error differ by at most ε .

The previous results used $\ln \#\mathcal{H}$ as a measure of complexity of the concept class \mathcal{H} .

VC-dimension is a tighter measure of complexity for a concept class.

In particular, we will find:

- ▶ $\text{VCdim}(\mathcal{H}) \leq \log_2 \#\mathcal{H}$.
- ▶ $\text{VCdim}(\mathcal{H})$ may be finite even for various infinite \mathcal{H} .

Definition (Vapnik & Chervonenkis 1971)

- ▶ A **set system** (Ω, \mathcal{H}) consists of a set Ω and a class \mathcal{H} of subsets of Ω .
- ▶ A set system (Ω, \mathcal{H}) **shatters** a set A if each subset of A can be expressed as a **shard** $A \cap h$ for some $h \in \mathcal{H}$.
- ▶ The **VC-dimension** $\text{VCdim}(\mathcal{H})$ of \mathcal{H} is the size of the largest set shattered by \mathcal{H} .

In other words, $\text{VCdim}(\mathcal{H}) = d$ iff

$\text{VCdim}(\mathcal{H}) \geq d$: There exists a set with d elements that is shattered.

$\text{VCdim}(\mathcal{H}) \leq d$: Each set with $d + 1$ elements is not shattered.

Lemma

Consider any Ω and finite \mathcal{H} . Then

$$\text{VCdim}(\mathcal{H}) \leq \log_2 \#\mathcal{H}.$$

If $\mathcal{H} = 2^A$ for some m -set $A \subseteq \Omega$ then

$$\text{VCdim}(\mathcal{H}) = \log_2 \#\mathcal{H} = m.$$

The second part shows that even for finite \mathcal{H} the VC dimension can be arbitrarily large.

Lemma

Consider $\Omega = \mathbb{R}^2$ and \mathcal{H} consists of all axis parallel rectangles. Then

$$\text{VCdim}(\mathcal{H}) = 4.$$

Proof. . . .



Lemma

$$\mathcal{H} \text{ shatters } A = \left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ -1 \end{bmatrix} \right\}.$$



Lemma

\mathcal{H} shatters no set of five points.

Proof.

Take a rectangle enclosing all five points.

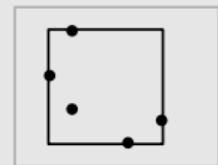
Shrink it until each side is stopped by a point.

For each side designate one point as stopper. It's ok if a point is a stopper for two edges. In case, several points stop an edge simultaneously choose one of them.

There are at most four stoppers.

Any rectangle containing the stoppers must contain all five points.

So $A \cap h$ can never be the set of the selected stoppers. □



Lemma

Consider $\Omega = \mathbb{R}$ and \mathcal{H} consists of all intervals. Then

$$\text{VCdim}(\mathcal{H}) = 2.$$

Lemma

Consider $\Omega = \mathbb{R}$ and \mathcal{H} consists of pairs of intervals. Then

$$\text{VCdim}(\mathcal{H}) = 4.$$

Lemma

Consider $\Omega = \mathbb{R}$ and \mathcal{H} consists of all m -sets. Then

$$\text{VCdim}(\mathcal{H}) = m.$$

Lemma

Consider $\Omega = \mathbb{R}$ and \mathcal{H} consists of all finite sets. Then

$$\text{VCdim}(\mathcal{H}) = \infty.$$

Lemma

Consider $\Omega = \mathbb{R}^2$ and \mathcal{H} consists of all polygons. Then

$$\text{VCdim}(\mathcal{H}) = \infty.$$

Lemma

Consider $\Omega = \mathbb{R}^d$ and \mathcal{H} consists of all affine halfspaces (ie. affine separators). Then

$$\text{VCdim}(\mathcal{H}) = d + 1.$$

Proof. . . .

□

This is connected with the fact that the XOR —described by four points in the plane, selecting a pair of opposite corners— cannot be described by halfplanes.

$\text{VCdim}(\mathcal{H}) \geq d + 1$: \mathcal{H} shatters $A = \{|0\rangle, |e_0\rangle, \dots, |e_{d-1}\rangle\}$ where $|e_i\rangle$ denotes the i -th standard unit vector.

$\text{VCdim}(\mathcal{H}) < d + 2$: Follows from Radon's theorem.

Theorem (Radon)

Any set $S \subseteq \mathbb{R}^d$ with $\#S \geq d + 2$ can be partitioned into two disjoint subsets A and B such that

$$\text{convex}(A) \cap \text{convex}(B) \neq \emptyset.$$

Proof.

Wlog. $\#S = d + 2$. [Otherwise drop some points.]

Let $T \in \mathbb{R}^{d \times (d+2)}$ be a matrix with columns being the points of S . Add an extra row of ones: $U \in \mathbb{R}^{(d+1) \times (d+2)}$. The columns of U must be linearly dependent, so take $|x\rangle \neq 0$ with $U|x\rangle = 0$.

Wlog. $x_0, \dots, x_{s-1} \geq 0$ and $x_s, \dots, x_{d+1} < 0$.

[Otherwise reorder S appropriately.]

Let A be the first s points in S and $B = S \setminus A$.

Normalize $|x\rangle$ so that $\sum_{i < s} |x_i| = 1$.

Now, $\sum_{i < s} |x_i| |U_{\cdot,i}\rangle = \sum_{s \leq i} |x_i| |U_{\cdot,i}\rangle$. The last row says $\sum_{i < s} |x_i| 1 = \sum_{s \leq i} |x_i| 1$. So $\sum_{i < s} |x_i| = 1$ and $\sum_{s \leq i} |x_i| = 1$. And thus dropping the last row yields

$$\sum_{i < s} |x_i| |T_{\cdot,i}\rangle = \sum_{s \leq i} |x_i| |T_{\cdot,i}\rangle,$$

an equality of convex combinations, and thus a point in the convex hull of A and in the convex hull of B . \square

Lemma

Consider $\Omega = \mathbb{R}^d$ and \mathcal{H} consists of all balls $B_r(|x_0\rangle) := \{|x\rangle \in \mathbb{R}^d \mid \| |x\rangle - |x_0\rangle \|_2 \leq r\}$.
Then

$$\text{VCdim}(\mathcal{H}) = d + 1.$$

Proof. . . .



Proof.

$\text{VCdim}(\mathcal{H}) \geq d + 1$: \mathcal{H} shatters $A = \{|0\rangle, |e_0\rangle, \dots, |e_{d-1}\rangle\}$ where $|e_i\rangle$ denotes the i -th standard unit vector.

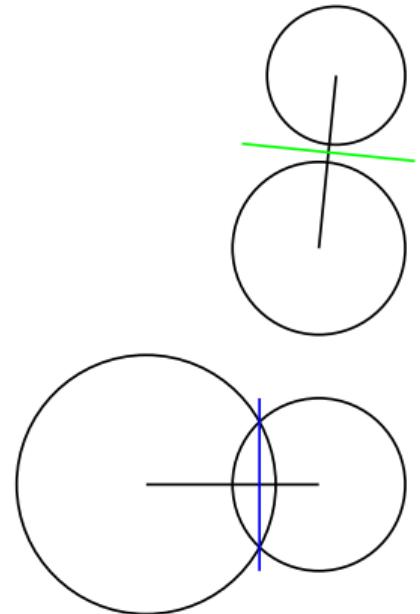
For any a -subset U of $A \setminus \{|0\rangle\}$ let $|c\rangle = \sum U$. Notice that

- ▶ $\text{dist}(|c\rangle, |v\rangle) = \sqrt{a-1}$ for $|v\rangle \in U$,
- ▶ $\text{dist}(|c\rangle, |0\rangle) = \sqrt{a}$ and
- ▶ $\text{dist}(|c\rangle, |v\rangle) = \sqrt{a+1}$ for $|v\rangle \in A \setminus \{|0\rangle\} \setminus U$.

Thus $U = A \cap B(|c\rangle, \frac{\sqrt{a-1}+\sqrt{a}}{2})$, $U \cup \{|0\rangle\} = A \cap B(|c\rangle, \frac{\sqrt{a+1}+\sqrt{a}}{2})$.

$\text{VCdim}(\mathcal{H}) < d + 2$: Take any set A shattered by \mathcal{H} and any partition $A = A_1 \uplus A_2$. Assume $A_1 = A \cap B_1$ and $A_2 = A \cap B_2$ for two balls $B_i \in \mathcal{H}$. Clearly, $A \cap B_1 \cap B_2 = \emptyset$. Now, find a hyperplane perpendicular to the line between the centers of B_1 and B_2 such that $B_1 \setminus B_2$ is on one side and $B_2 \setminus B_1$ on the other. Thus halfspaces shatter A .

And then by the previous then $\#A < d + 2$.



□

Some VC-dimensions

Ω	\mathcal{H}	$\text{VCdim}(\mathcal{H})$
Ω	some finite set	$\leq \log_2 \#\mathcal{H}$
\mathbb{R}	intervals	2
\mathbb{R}	pairs of intervals	4
\mathbb{R}	m -sets	m
\mathbb{R}	finite sets	∞
\mathbb{R}^2	axis parallel rectangles	4
\mathbb{R}^2	polygons	∞
\mathbb{R}^d	halfspaces	$d + 1$
\mathbb{R}^d	spheres	$d + 1$

More examples can be found in Figure 1.2 of

- ▶ Michael M. Wolf (2018). Mathematical Foundations of Supervised Learning (Version of June 6, 2018).

Shatter function

Let's ask the more detailed question:

Question

Given a set system (Ω, \mathcal{H}) . What is the largest number of subsets of an n -set A that can be expressed as a shard $A \cap h$ with $h \in \mathcal{H}$?

Shatter function

Definition

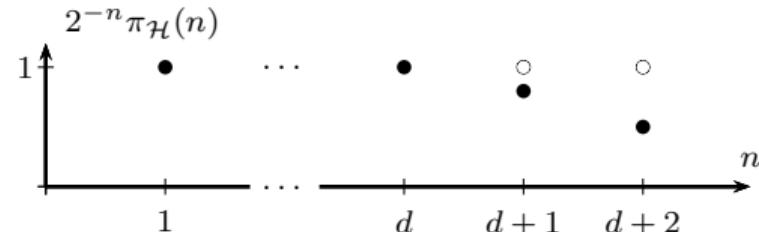
The shatter function $\pi_{\mathcal{H}}$ is given by

$$\pi_{\mathcal{H}}(n) := \max \left\{ \underbrace{\#\{A \cap h \mid h \in \mathcal{H}\}}_{=: \pi_{\mathcal{H}}(A)} \middle| A \in \binom{\Omega}{n} \right\}.$$

Clearly,

- ▶ $\pi_{\mathcal{H}}(n) = 2^n$ for $n \leq \text{VCdim}(\mathcal{H})$ and
- ▶ $\pi_{\mathcal{H}}(n) < 2^n$ for $n > \text{VCdim}(\mathcal{H})$.

Can we say more in the latter case?



Define

$$\binom{n}{\leq d} := \binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{d}.$$

Note $\binom{n}{\leq d} \leq n^d + 1$.

Lemma (Sauer 1972, Shelah 1972, ...)

For any set system (Ω, \mathcal{H}) of VC-dimension at most d and for each n we have

$$\pi_{\mathcal{H}}(n) \leq \binom{n}{\leq d}.$$

Proof. . . .

□

Proof.

First, note for $n, d \geq 1$

$$\binom{n}{\leq d} = \binom{n-1}{\leq d-1} + \binom{n-1}{\leq d}.$$

Either the first element of Ω is part of the $\leq d$ -set or it is not. That partitions and...

Having noted this we proceed by induction.

Base (n, d) with $n \leq d$ or $d = 0$.

Step (n, d) using $(n-1, d-1)$ and $(n-1, d)$, assuming the lemma holds for instances with smaller n .

Consider $n \leq d$. Then $\binom{n}{\leq d} = 2^n$ and $\pi_{\mathcal{H}}(n) = 2^n$. The lemma holds with equality.

Consider $d = 0$. Then $\#\mathcal{H} \leq 1$. Otherwise, pick $h_0, h_1 \in \mathcal{H}$ and $A = \{a\}$ with $a \in h_0 \Delta h_1$. Then \mathcal{H} shatters A and $\text{VCdim}(\mathcal{H}) \geq 1$. Contradiction. So $\pi_{\mathcal{H}}(n) = 1 = \binom{n}{\leq 0}$.

Consider the general case, namely $n > d > 0$ since the rest is in the base case. Our aim is to prove that $\pi_{\mathcal{H}}(n) \leq \binom{n}{\leq d}$.

Take an n -set A with $\pi_{\mathcal{H}}(n) = \pi_{\mathcal{H}}(A)$. Wlog. $A = \Omega$. [Otherwise cut away anything outside A , namely put $\tilde{X} = A$, $\tilde{\mathcal{H}} = \{A \cap h \mid h \in \mathcal{H}\}$. Then $\pi_{\mathcal{H}}(n) = \pi_{\mathcal{H}}(A) = \pi_{\tilde{\mathcal{H}}}(A)$. So it suffices to prove $\pi_{\tilde{\mathcal{H}}}(A) \leq \binom{n}{\leq d}$.] Actually, now $\pi_{\mathcal{H}}(A) = \#\mathcal{H}$.

Fix some $u \in A$ and put

$$\begin{aligned}\mathcal{H}_0 &:= \{h \in \mathcal{H} \mid u \in h\}, \\ \mathcal{H}_1 &:= \{h \in \mathcal{H} \mid u \notin h, h \cup \{u\} \notin \mathcal{H}\}, \\ \mathcal{H}_2 &:= \{h \in \mathcal{H} \mid u \notin h, h \cup \{u\} \in \mathcal{H}\}, \\ \mathcal{H}_3 &:= \{h \in \mathcal{H} \mid u \in h, h \setminus \{u\} \in \mathcal{H}\}.\end{aligned}$$

Notice that the extra condition in \mathcal{H}_1 grants that $\mathcal{H}_0 \bigcirclearrowleft \{\{u\}\}$ and \mathcal{H}_1 are disjoint. Also note that $\mathcal{H}_2 = \mathcal{H}_3 \bigcirclearrowleft \{\{u\}\}$. Let

$$\begin{aligned}\tilde{\mathcal{H}}_1 &:= (\mathcal{H}_0 \bigcirclearrowleft \{\{u\}\}) \uplus \mathcal{H}_1, \\ \tilde{\mathcal{H}}_2 &:= \mathcal{H}_2 = \mathcal{H}_3 \bigcirclearrowleft \{\{u\}\}.\end{aligned}$$

It suffices to show

1. $\text{VCdim}(\tilde{\mathcal{H}}_1) \leq d$,
2. $\text{VCdim}(\tilde{\mathcal{H}}_2) \leq d - 1$,
3. $\pi_{\mathcal{H}}(A) \leq \pi_{\tilde{\mathcal{H}}_1}(A \setminus \{u\}) + \pi_{\tilde{\mathcal{H}}_2}(A \setminus \{u\})$.

Because by induction hypothesis then

$$\begin{aligned}\pi_{\mathcal{H}}(n) &= \pi_{\mathcal{H}}(A) \\ &\leq \pi_{\tilde{\mathcal{H}}_1}(A \setminus \{u\}) + \pi_{\tilde{\mathcal{H}}_2}(A \setminus \{u\}) \\ &\leq \binom{n-1}{\leq d} + \binom{n-1}{\leq d-1} = \binom{n}{\leq d}\end{aligned}$$

and we are done.

Ad 1: Suppose $\tilde{\mathcal{H}}_1$ shatters a $(d+1)$ -set $B \subseteq A \setminus \{u\}$. Then $\mathcal{H}_0 \cup \mathcal{H}_1$ also shatters B and so would \mathcal{H} contradicting the assumption $\text{VCdim}(\mathcal{H}) \leq d$.

Ad 2: Suppose $\tilde{\mathcal{H}}_2$ shatters a d -set $B \subseteq A \setminus \{u\}$. Then \mathcal{H} shatters $B \uplus \{u\}$, since for each $h \in \tilde{\mathcal{H}}_2$ both h and $h \uplus \{u\}$ are in \mathcal{H} . But that again contradicts the assumption $\text{VCdim}(\mathcal{H}) \leq d$.

Ad 3: First, notice

$$\pi_{\mathcal{H}}(A) \leq \pi_{\mathcal{H}_0 \uplus \mathcal{H}_1}(A) + \pi_{\mathcal{H}_2}(A).$$

This follows since \mathcal{H} is the (disjoint) union of $\mathcal{H}_0 \uplus \mathcal{H}_1$ and \mathcal{H}_2 .

Further,

$$\pi_{\mathcal{H}_0 \uplus \mathcal{H}_1}(A) = \pi_{\tilde{\mathcal{H}}_1}(A \setminus \{u\}).$$

As each side is just the size of the class, this is due to the fact that $\mathcal{H}_0 \bigcirclearrowleft \{\{u\}\}$ is disjoint from \mathcal{H}_1 . Finally,

$$\pi_{\mathcal{H}_2}(A) = \pi_{\tilde{\mathcal{H}}_2}(A \setminus \{u\}).$$

Just notice that none of the sets in \mathcal{H}_2 contains u . That's all, folks! \square

Some shatter functions

Ω	\mathcal{H}	$\text{VCdim}(\mathcal{H})$	$\pi_{\mathcal{H}}(n)$
Ω	some finite set	$\leq \log_2 \#\mathcal{H}$	$\leq \#\mathcal{H} \in \Theta(1)$!
\mathbb{R}	intervals	2	$\binom{n+1}{2} + 1$
\mathbb{R}	pairs of intervals	4	$\binom{n+1}{4} + \binom{n+1}{2} + 1$
\mathbb{R}	m -sets	m	$\binom{n}{\leq m} \in \Theta(n^m)$
\mathbb{R}	finite sets	∞	2^n
\mathbb{R}^2	axis parallel rectangles	4	$\Theta(n^4)$
\mathbb{R}^2	polygons	∞	2^n
\mathbb{R}^d	halfspaces	$d + 1$	$\Theta(n^d)$!
\mathbb{R}^d	balls	$d + 1$	$\Theta(n^{d+1})$

Combinations of concepts

What can we say about, say, the hypothesis class consisting of intersections of two linear separators?

Definition

Let (Ω, \mathcal{H}_0) and (Ω, \mathcal{H}_1) be two set systems on the same underlying set Ω . Define the intersection system $(\Omega, \mathcal{H}_0 \odot \mathcal{H}_1)$ where

$$\mathcal{H}_0 \odot \mathcal{H}_1 = \{h_0 \cap h_1 \mid h_0 \in \mathcal{H}_0, h_1 \in \mathcal{H}_1\}.$$

Example: Apply this to $\Omega = \mathbb{R}^d$, $\mathcal{H}_0 = \mathcal{H}_1$ being all halfspaces. This corresponds to taking the Boolean AND of the output of two threshold gates and is the most basic neural net besides a single gate.

Lemma

Let (Ω, \mathcal{H}_0) and (Ω, \mathcal{H}_1) be two set systems on the same underlying set Ω . Then

$$\pi_{\mathcal{H}_0 \odot \mathcal{H}_1}(n) \leq \pi_{\mathcal{H}_0}(n) \cdot \pi_{\mathcal{H}_1}(n).$$

Proof. . . .



Proof.

Just observe

$$A \cap (h_0 \cap h_1) = (A \cap h_0) \cap (A \cap h_1).$$

For an n -set A there are up to $\pi_{\mathcal{H}_0}(n) \cdot \pi_{\mathcal{H}_1}(n)$ pairs $(A \cap h_0, A \cap h_1)$ and that bounds the number of possible intersections. \square

Lemma

Let (Ω, \mathcal{H}_0) and (Ω, \mathcal{H}_1) be two set systems on the same underlying set Ω . Then

$$\pi_{\mathcal{H}_0 \odot \mathcal{H}_1}(n) \leq \pi_{\mathcal{H}_0}(n) \cdot \pi_{\mathcal{H}_1}(n).$$

Proof. . . .



Speculations

Notice that for $f(n) \in (1 + o(1)) \cdot n^v$ we have $\frac{\ln f(n)}{\ln n} \in v + \frac{o(1)}{\ln n}$ and so $v = \lim_{n \rightarrow \infty} \frac{\ln f(n)}{\ln n}$.

Similarly, from the Sauer-Shelah lemma we can infer

$$\overline{\lim}_{n \rightarrow \infty} \frac{\ln \pi_{\mathcal{H}}(n)}{\ln n} \leq \text{VCdim}(\mathcal{H}).$$

~~Assuming equality~~, the previous lemma thus ~~would~~ imply

~~$$\text{VCdim}(\mathcal{H}_0 \odot \mathcal{H}_1) \leq \text{VCdim}(\mathcal{H}_0) + \text{VCdim}(\mathcal{H}_1).$$~~

But, alas, 'assuming equality' is too simplistic: for example, for \mathcal{H} finite the VC-dimension d can be arbitrarily large and we have $\pi_{\mathcal{H}}(n) \leq \#\mathcal{H}$ which cannot be $\Omega(n^d)$.

Instead of a simple AND we may use an arbitrary Boolean function $f: \{0, 1\}^k \rightarrow \{0, 1\}$ for combining concepts and define

$$\text{COMB}_f(\mathcal{H}) := \{\text{comb}_f(h_0, \dots, h_{k-1}) \mid \forall i: h_i \in \mathcal{H}\}$$

with $\text{comb}_f(h_0, \dots, h_{k-1}) = \{x \in \Omega \mid f(x \in h_0, \dots, x \in h_{k-1}) = 1\}$.
You may think of a depth-two neural network.

Lemma

For each integer k , any k -ary Boolean function f and any hypothesis class \mathcal{H} we have

$$\pi_{\text{COMB}_f(\mathcal{H})}(n) \leq \pi_{\mathcal{H}}(n)^k.$$

Theorem

If concept class \mathcal{H} has VC-dimension d then for any integer k and any k -ary Boolean function f , the class $\text{COMB}_f(\mathcal{H})$ has VC-dimension $\mathcal{O}(kd \log_2(kd))$.

Proof. . . .



Remark

This is a little weaker than the speculated kd bound.

Proof.

Let $n = \text{VCdim}(\text{COMB}_f(\mathcal{H}))$. Then there must be a set A of n points shattered by $\text{COMB}_f(\mathcal{H})$.

Denoting $d = \text{VCdim}(\mathcal{H})$ the Sauer-Shelah lemma implies $\pi_{\mathcal{H}}(n) \leq n^d + 1$. We can drop the '+1' unless $d = 1$. So, for simplicity, we only consider $d \neq 1$.

Each set in $\text{COMB}_f(\mathcal{H})$ is determined by k sets in \mathcal{H} , and there are at most n^{kd} different k -tuples of such sets relevant to a given set A . [To describe a shard $A \cap \text{comb}_f(h_0, \dots, h_{k-1}) = \{x \in A \mid f(x \in h_0, \dots, x \in h_{k-1}) = 1\}$ you only need to know $A \cap h_0, \dots, A \cap h_{k-1}$.]

Since A is shattered we must have $2^n \leq n^{kd}$. Or, equivalently,

$$n \leq kd \log_2 n.$$

Thus $n \leq kd \log_2(n) \leq kd \log_2(kd \log_2 n) = kd(\log_2(kd) + \log_2 \log_2 n)$. Last, $\log_2 n \leq \frac{n}{\log_2 n} \leq kd$ for $n \geq 16$. This yields the desired estimate.

The omitted case $d = 1$ is similar. [From $\pi_{\mathcal{H}}(n) \leq n + 1$ we obtain $n \leq 2k \log_2(k + 1)$. This is of same order.] □

Theorem

If concept class \mathcal{H} has VC-dimension d then for any integer k and any k -ary Boolean function f , the class $\text{COMB}_f(\mathcal{H})$ has VC-dimension $\mathcal{O}(kd \log_2(kd))$.

Proof. . . .



Remark

This is a little weaker than the speculated kd bound.

Theorem (The key theorem)

Let (Ω, \mathcal{H}) be a set system, D a probability distribution over Ω , $\delta \in]0, \frac{1}{8}[$, $\varepsilon > 0$ and let n be an integer satisfying

$$n \geq \frac{2}{\varepsilon} \log_2 \frac{2\pi_{\mathcal{H}}(2n)}{\delta}.$$

Let S_1 consists of n points drawn from D . With probability greater than or equal to $1 - \delta$, every set in \mathcal{H} of probability mass greater than ε intersects S_1 .

Proof. . .



The following method is called “double sampling” or the “ghost sample method”.

Proof.

Draw a second set S_2 of n points from D and consider the events

$$A \quad \exists h \in \mathcal{H}: \text{prob}(h) \geq \varepsilon \wedge \#(S_1 \cap h) = 0.$$

$$B \quad \exists h \in \mathcal{H}: \text{prob}(h) \geq \varepsilon \wedge \#(S_1 \cap h) = 0 \wedge \#(S_2 \cap h) \geq \frac{\varepsilon}{2}n.$$

By Chebyshev we find $\text{prob}(B|A) \geq 1 - \frac{4}{\varepsilon n} \geq \frac{1}{2}$. [Put $X = \#(S_2 \cap h) = \sum_i I_{S_2 i \in h}$. (We ignore collisions ...) Notice $\mathbb{E}X = n \cdot \text{prob}(h)$ and $\text{var } X = n \cdot \text{prob}(h) \cdot (1 - \text{prob}(h)) = \mathbb{E}X \cdot (1 - \text{prob}(h))$. Conditioned under A fix its h and so $\text{prob}(h) \geq \varepsilon$. By Chebyshev we obtain

$$\begin{aligned} \text{prob}\left(X \geq \frac{\varepsilon}{2}n\right) &\leq \text{prob}\left(|X - \mathbb{E}X| \geq \frac{1}{2}\mathbb{E}X\right) \\ &\leq \frac{4 \text{var } X}{(\mathbb{E}X)^2} = \frac{4(1 - \text{prob}(h))}{\mathbb{E}X} \\ &\leq \frac{4}{n\varepsilon} \leq \frac{1}{2} \end{aligned}$$

whenever $n \geq \frac{8}{\varepsilon}$. The latter is granted by the condition on n if only $\frac{2\pi_{\mathcal{H}}(2n)}{\delta} \geq 2^4$ or $\pi_{\mathcal{H}}(2n) \geq 2^3\delta$. This follows if $\mathcal{H} \neq \emptyset$ and $\delta < \frac{1}{8}$, say.] Consequently,

$$\begin{aligned} \text{prob}(B) &\geq \text{prob}(A \cap B) = \text{prob}(B|A)\text{prob}(A) \\ &\geq \frac{1}{2}\text{prob}(A) \end{aligned}$$

and to prove $\text{prob}(A) \leq \delta$ it suffices to prove $\text{prob}(B) \leq \frac{\delta}{2}$.

For the latter, we consider a second way of picking S_1 and S_2 : Draw a random set S_3 of $2n$ points from D and in a second step partition S_3 into two equal pieces S_1 and S_2 .

Now, consider the point in time after S_3 has been drawn but before it has been randomly partitioned into S_1 and S_2 . We know $\pi_{\mathcal{H}}(S_3) \leq \pi_{\mathcal{H}}(2n)$. Thus it suffices to show that for each given $h' \in \{S_3 \cap h \mid h \in \mathcal{H}\}$ the probability over the random partition of S_3 into S_1 and S_2 that $\#(S_1 \cap h') = 0$ and $\#(S_2 \cap h') \geq \frac{\varepsilon}{2}n$ is at most $\frac{\delta}{2\pi_{\mathcal{H}}(2n)}$.

► Case $\#h' < \frac{\varepsilon}{2}n$. Then $\#(S_2 \cap h') \geq \frac{\varepsilon}{2}n$ is impossible and the probability is zero.

► Case $\#h' \geq \frac{\varepsilon}{2}n$. Now the probability that none of the points in h' fall into S_1 , that is, all points of h' fall into the ghost sample S_2 , is at most $2^{-\frac{\varepsilon}{2}n}$. [Assume the points of h' are distributed first.

The probability that the first goes to S_2 is $\frac{1}{2}$. For the second it is $\frac{n-1}{2n-1} < \frac{1}{2}$] Using the bound on n this probability is at most $\frac{\delta}{2\pi_{\mathcal{H}}(2n)}$ as desired:

$$2^{-\frac{\varepsilon}{2}n} \leq 2^{-\log_2 \frac{2\pi_{\mathcal{H}}(2n)}{\delta}} = \frac{\delta}{2\pi_{\mathcal{H}}(2n)}.$$

Thus $\text{prob}(B) \leq \frac{\delta}{2}$ and $\text{prob}(A) \leq \delta$. □



Theorem (The key theorem)

Let (Ω, \mathcal{H}) be a set system, D a probability distribution over Ω , $\delta \in]0, \frac{1}{8}[$, $\varepsilon > 0$ and let n be an integer satisfying

$$n \geq \frac{2}{\varepsilon} \log_2 \frac{2\pi_{\mathcal{H}}(2n)}{\delta}.$$

Let S_1 consists of n points drawn from D . With probability greater than or equal to $1 - \delta$, every set in \mathcal{H} of probability mass greater than ε intersects S_1 .

Proof. . .



Theorem (Sample bound)

For any class \mathcal{H} and distribution D , if a training sample S is drawn from D of size

$$n \geq \frac{2}{\varepsilon} \log_2 \frac{2\pi_{\mathcal{H}}(2n)}{\delta}$$

then with probability greater than or equal to $1 - \delta$, every $h \in \mathcal{H}$ with $\text{err}_D(h) \geq \varepsilon$ has $\text{err}_S(h) > 0$. Equivalently, every $h \in \mathcal{H}$ with $\text{err}_S(h) = 0$ has $\text{err}_D(h) < \varepsilon$.

Proof. . . .



Proof.

Let c^* be the target concept and put $\mathcal{H}' = \mathcal{H} \setminus \{c^*\} = \{h \Delta c^* \mid h \in \mathcal{H}\}$. Note that \mathcal{H}' and \mathcal{H} have the same VC-dimension and shatter function. [We have $A \cap (h \Delta c^*) = (A \cap h) \Delta (A \cap c^*)$. Then,

$$\begin{aligned}\pi_{\mathcal{H}'}(A) &= \# \{A \cap (h \Delta c^*) \mid h \in \mathcal{H}\} \\ &= \# \{(A \cap h) \Delta (A \cap c^*) \mid h \in \mathcal{H}\} \\ &\stackrel{!}{=} \# \{A \cap h \mid h \in \mathcal{H}\} \quad = \pi_{\mathcal{H}}(A).\end{aligned}$$

And so the shatter functions coincide and thus also the VC-dimension.] Now, draw a training sample S from D of size n . By the key theorem with probability at least $1 - \delta$, for each $h' = h \Delta c^* \in \mathcal{H}'$ with $\text{prob}(h') \geq \varepsilon$ we have $\#(S \cap (h \Delta c^*)) > 0$. In other words, if $\text{err}_D(h) \geq \varepsilon$ then $\text{err}_S(h) > 0$. □

Theorem (Sample bound)

For any class \mathcal{H} and distribution D , if a training sample S is drawn from D of size

$$n \geq \frac{2}{\varepsilon} \log_2 \frac{2\pi_{\mathcal{H}}(2n)}{\delta}$$

then with probability greater than or equal to $1 - \delta$, every $h \in \mathcal{H}$ with $\text{err}_D(h) \geq \varepsilon$ has $\text{err}_S(h) > 0$. Equivalently, every $h \in \mathcal{H}$ with $\text{err}_S(h) = 0$ has $\text{err}_D(h) < \varepsilon$.

Proof. . . .



A combination of the techniques is possible and yields:

Theorem (Growth function uniform convergence)

For any class \mathcal{H} and distribution D , if a training sample S is drawn from D of size

$$n \geq \frac{8}{\varepsilon^2} \ln \frac{2\pi_{\mathcal{H}}(2n)}{\delta}$$

then with probability greater than or equal to $1 - \delta$, every $h \in \mathcal{H}$ will have
 $|\text{err}_D(h) - \text{err}_S(h)| \leq \varepsilon$.

We can write the previous theorems in terms of VC-dimension. The sample bound theorem yields

Corollary

For any class \mathcal{H} and distribution D , a training sample S drawn from D of size

$$\Omega\left(\frac{2}{\varepsilon}\left(\text{VCdim}(\mathcal{H}) \log \frac{1}{\varepsilon} + \log \frac{1}{\delta}\right)\right)$$

is sufficient to ensure that with probability greater than or equal to $1 - \delta$, every $h \in \mathcal{H}$ with $\text{err}_D(h) \geq \varepsilon$ has $\text{err}_S(h) > 0$. Equivalently, every $h \in \mathcal{H}$ with $\text{err}_S(h) = 0$ has $\text{err}_D(h) < \varepsilon$.

Machine learning:

VC-dimension and generalizing

Theorem	true error with prob. $\geq 1 - \delta$	requirements
First sample bound	$\leq \varepsilon$	$n \geq \frac{1}{\varepsilon} \ln \frac{\#\mathcal{H}}{\delta}$ and $\text{err}_S(h) = 0$
Uniform convergence	$< \text{err}_S(h) + \varepsilon$	$n \geq \frac{1}{2\varepsilon^2} \ln \frac{2\#\mathcal{H}}{\delta}$
Sample bound	$\leq \varepsilon$	$n \geq \frac{2}{\varepsilon} \log_2 \frac{2\pi_{\mathcal{H}}(2n)}{\delta}$ and $\text{err}_S(h) = 0$
Growth function uniform convergence	$\leq \text{err}_S(h) + \varepsilon$	$n \geq \frac{8}{\varepsilon^2} \ln \frac{2\pi_{\mathcal{H}}(2n)}{\delta}$
Sample bound corollary	$\leq \varepsilon$	$n \geq \Omega\left(\frac{2}{\varepsilon} (\text{VCdim}(\mathcal{H}) \log \frac{1}{\varepsilon} + \log \frac{1}{\delta})\right)$ and $\text{err}_S(h) = 0$

Consequences for learning

Want:

predict a binary classification based on training data.

Namely, find a hypothesis h from some concept class \mathcal{H} that performs as good as possible.

- ▶ The parameters are:
 - ▶ the concept class \mathcal{H} ,
 - ▶ the distribution D ,
 - ▶ the amount $\#S$ of training data,
 - ▶ the training error err_S (aka. empirical risk),
 - ▶ the true error err_D (aka. risk).
- ▶ The more training data the better.
- ▶ The smaller the VC-dimension or the growth of \mathcal{H} the better.
- ▶ Also notice that the given guarantees only hold with probability $1 - \delta$ and the quality depends on δ :
 - ▶ As δ approaches 0, the quality decreases.
 - ▶ Large δ , ie. close to 1, diminishes the guarantees.

The previous results tell us how to achieve good answers:

Various other measures apart from VC-dimension are studied to fasten the ties.

For example:

- ▶ Empirical Rademacher complexity of \mathcal{H} is defined as

$$R_S(\mathcal{H}) = \mathbb{E} \left(\max \left\{ \frac{1}{n} \sum_{x \in S} \sigma_x h(x) \mid h \in \mathcal{H} \right\} \mid \sigma \xrightarrow{\text{IID}} \{+1, -1\}^S \right).$$

This allows results like our uniform convergence corollary with

$$\text{err}_D(h) - \text{err}_S(h) \leq R_S(\mathcal{H}) + 3 \sqrt{\frac{\ln \frac{2}{\delta}}{2 \# S}}.$$

More about this you find in statistical learning theory.

A remark about language

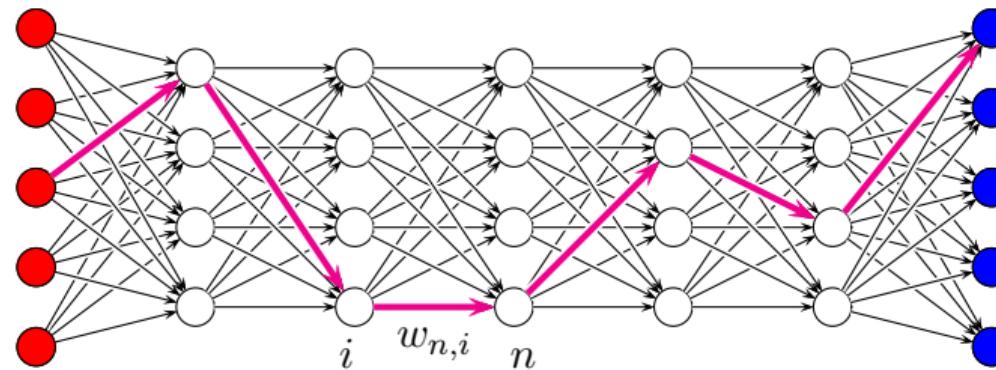
- ▶ Training error = sample error = empirical error $\hat{=}$ empirical risk.
- ▶ True error $\hat{=}$ risk = true risk.

Machine learning:

*Deep learning

- ▶ Supervised learning: Training data can be used including answers. We are looking for answers to new data items.
- ▶ Unsupervised learning: Only data is given. We are looking for answers.

A neural network



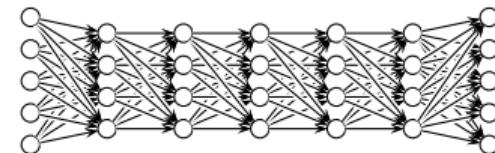
- ▶ Each ‘wire’ carries a value, usually in \mathbb{R} .
- ▶ Each node, ie. **neuron**, takes a weighted sum of its inputs, applies a threshold function t and outputs the result: $y_n \leftarrow f_n(y) = t(\sum_i w_{n,i} y_i)$.
- ▶ The network computes the concatenated function f_w .

How to train it?

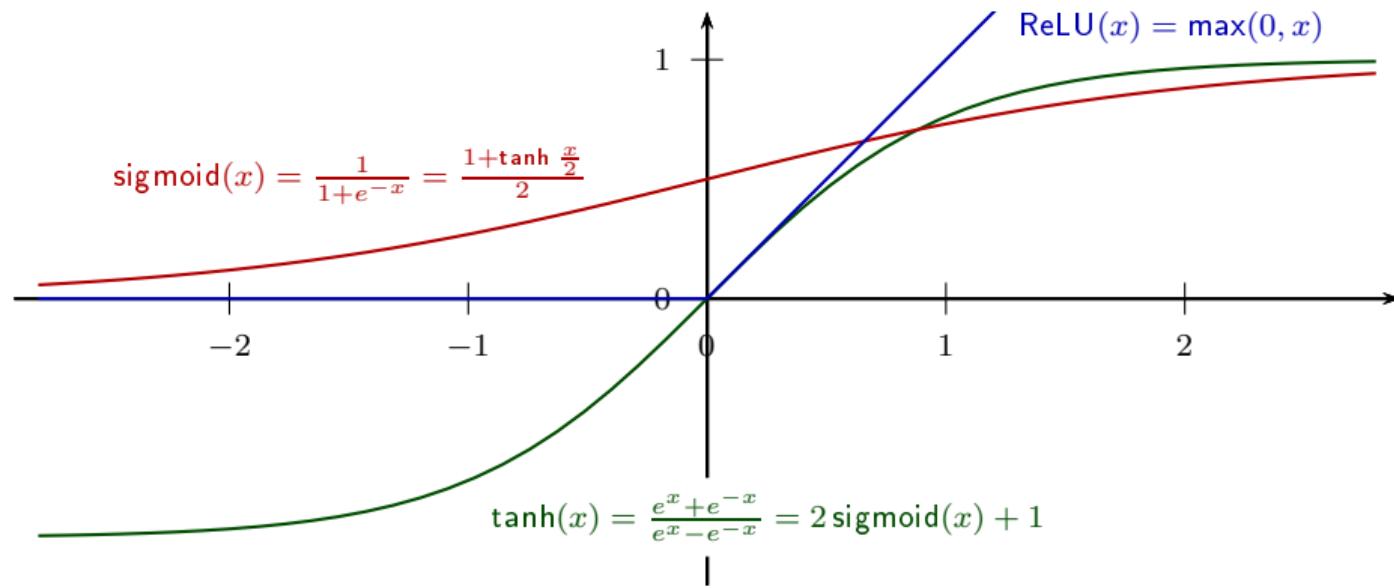
- ▶ The output function f_w depends on the weights w .
- ▶ Use training samples $x \in S$ with labels $c^*(x)$.
- ▶ Minimize the training error, eg. the sum of squared deviations

$$\text{err}_S(w) = \sum_{x \in S} (f_w(x) - c^*(x))^2.$$

- ▶ For minimizing it is advantageous if the total function f_w is smooth wrt. w .



Threshold functions



Question

How to find good weights?

- ▶ Want to minimize the training error:

$$\text{err}_S(w).$$

It is a function in many variables, namely the unknown weights w and the known structure of the neural network and the known training set.

- ▶ Optimization can be done by gradient descent if the function is sufficiently **smooth**. Recall:

The gradient $\nabla \text{err}_S(w) = \left[\frac{\partial \text{err}(w)}{\partial w_z} \right]_z$ points in direction of steepest ascent.

Thus in direction $-\nabla \text{err}_S(w)$ the function gets smaller.

But how far to move?

There are numerous strategies, but often depending on features of the function that we cannot assess. Often experimental heuristics are used...

Machine learning:

*Deep learning: Gradient descent

For example: Nesterov's Accelerated Gradient

Update according to

$$v \leftarrow \gamma v + \alpha \nabla \text{err}(w),$$

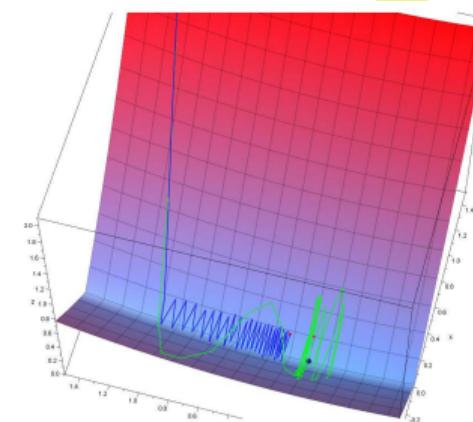
$$w \leftarrow w - v.$$

- ▶ v : velocity vector.
- ▶ α : learning rate.
- ▶ γ : typically 0.5 until learning stabilizes and then increased to 0.9.

Problems

Gradient descent ($\gamma = 0, \alpha = 0.2$).
Nesterov's... ($\gamma = 0.8, \alpha = 0.2$).

- ▶ Ravines...



- ▶ Local minima...

Machine learning:

*Deep learning: Stochastic gradient descent

The error function is actually the sum of many simple functions:

$$\text{err}_S(w) = \sum_{x \in S} \text{err}_x(w)$$

We might have, say,

- ▶ 100 000 images,
- ▶ 1 000 000 weights.

Alternative

- ▶ At each iteration pick one image $x \in S$.
- ▶ Improve $\text{err}_x(w)$.

In practice, increase the number of images in an iteration from 1, to 50, then 200, finally all.

Machine learning:

*Deep learning: Stochastic gradient descent

Imagine gradient descent starting at around 1 000

click here

Hopcroft et al. (2018), Figure 5.7

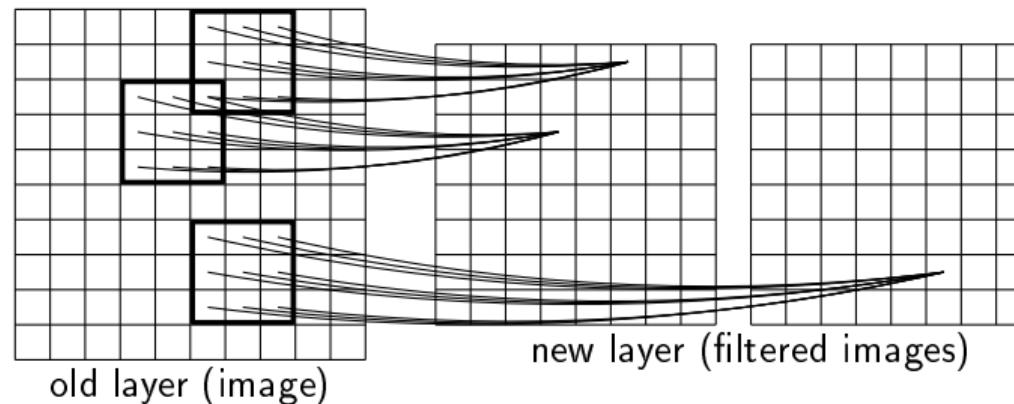
click here

Hopcroft et al. (2018), Figure 5.8

Adapt the network

You *may* fully connect layers. But the **more weights** are used the **more difficult to train** and the **more dangerous to overfit data**.

Fewer weights are needed when adapting to the problem structure. For images **convolution** is helpful:

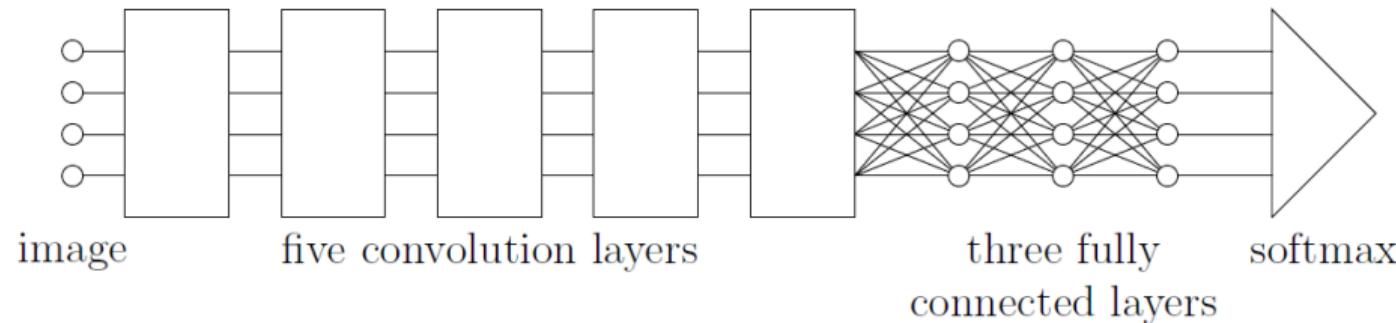


For this double 3×3 -convolution we only need 18 weights rather than 12800.

Machine learning:

*Deep learning: Tweaks

More sophisticated networks adapt more sophisticatedly...



Hopcroft et al. (2018), Figure 5.5

Reverse engineering questions

- ▶ Recreating an image from an activation vector?

Given the input values of later neurons what can we say about the networks input?
→ use gradient descent to find out.

- ▶ Style transfer?

For example, find an image with similar first level activation than a style image but close to a given image.

- ▶ Fooling.

Construct a small variation of —say— a cat image that a given net will classify as an automobile. Use gradient descent to find out best tiny mods —say— just up to ± 1 in colors.

- ▶ What does a neural network tell about its training data?

- ▶ ...

Generative Adversarial Networks (GANs)

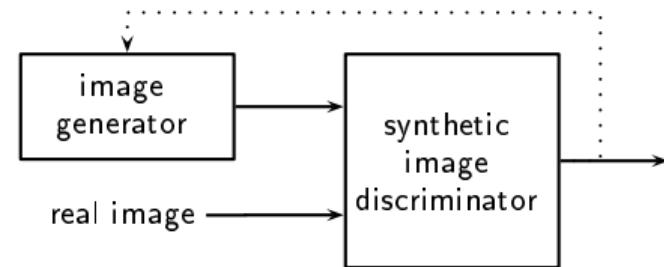
Question

How can we create synthetic real images?

Ethical point: **Do we want that?** ...

One discussed solution uses several interconnected neural networks:

- ▶ One first trains an image discriminator. It distinguishes between real images and synthetic ones.
- ▶ Then one trains an image generator to generate images that the discriminator believes are real images.
- ▶ Alternate the training between the two networks to further improve the discriminator and the generator.

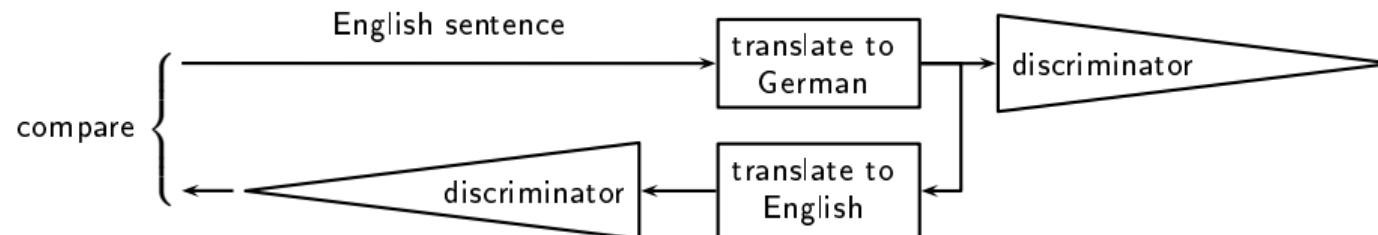


Generative Adversarial Networks (GANs)

Question

How to translate English to German?

Same idea: use a discriminator that distinguishes well-formed German opposed to synthetic.



Machine learning:

*Online learning

- ▶ So far **batch learning**,
 - ▶ ie. entire training set S available from the beginning.
- ▶ In **online learning** at each time t two events occur:
 1. The algorithm is given a further example $x_t \in \Omega$ and shall predict its label: ℓ_t .
 2. Then the algorithm is told the true label $c^*(x_t)$ and is charged for a mistake if $c^*(x_t) \neq \ell_t$.

Goal: Make as few mistakes as possible in total.

Learning disjunctions

- ▶ $\Omega = \{0, 1\}^d$.
- ▶ $\mathcal{H} = \left\{ \bigvee_{i \in y} v_i \mid y \subseteq \{0, \dots, d-1\} \right\}$:
set of all possible ORs of v_0, \dots, v_{d-1} , eg. $h = v_2 \vee v_3 \vee v_5$.

Strategy

- ▶ Start with the 'all' hypothesis $h = v_0 \vee \dots \vee v_{d-1}$.
- ▶ On each new example x predict acc.to the present h .
- ▶ On a mistake, remove the variables set to 1 in the example x .

We make at most d mistakes and this is actually optimal for deterministic strategies.

The halving algorithm

if you do not care for efficiency...

Strategy

- ▶ Maintain a version space $\mathcal{V} \subseteq \mathcal{H}$, start with $\mathcal{V} \leftarrow \mathcal{H}$.
- ▶ On a new example, predict acc.to the **majority** result among the hypotheses \mathcal{V} .
- ▶ On a mistake, discard the wrong hypotheses from \mathcal{V} .

On each mistake this strategy reduces the version space \mathcal{V} to at most half.

Thus the total number of mistakes is at most $\log_2 \#\mathcal{H}$.

Recall: The perceptron algorithm (batch version)

1. $|w\rangle \leftarrow 0$.
2. **While** there exists $|x_i\rangle$ with $\langle w| \cdot \ell_i |x_i\rangle \leq 0$ **do** 3–3
3. $|w\rangle \leftarrow |w\rangle + \ell_i |x_i\rangle$.

This is based on the following intuition:

- ▶ When updating $|w\rangle$ we have

$$(\langle w| + \ell_i \langle x_i|) \cdot \ell_i |x_i\rangle = \langle w| \cdot \ell_i |x_i\rangle + \||x_i\rangle\|_2^2$$

larger than $\langle w| \cdot \ell_i |x_i\rangle$.

- ▶ Of course, what is good for $|x_i\rangle$ needs not be good for other $|x_j\rangle$.
- ▶ It turns out that this approach **does work** and **the better the wider the corridor**.

The perceptron algorithm (online version)

1. $|w\rangle \leftarrow 0$.
2. **For** $t = 0, \dots$ **do** 3–4
3. Given an example $|x_t\rangle$ predict $\ell_t \leftarrow \text{sgn}(\langle w | x \rangle)$.
4. If the prediction was mistaken, update $|w\rangle \leftarrow |w\rangle + c^*(x_t) |x_t\rangle$.

Theorem

On any sequence of examples $|x_0\rangle, |x_1\rangle, \dots$, if there exists a vector $|w^\rangle$ such that $\langle w^* | \cdot c^*(x_t) |x_t\rangle \geq 1$ then the perceptron algorithm makes at most $r^2 \|\langle w^* | \|^2$ mistakes where $r = \max_t \| |x_t\rangle \|_2$. □*

Machine learning:

*Online learning

*Extension for perceptron in an imperfect scenario

- ▶ What if there is no perfect $|w^*\rangle$?
- ▶ ...as typical in real life...

Define the *total hinge-loss* of $|w^*\rangle$ as

≤ 0 for perfect $|w^*\rangle$

$$L_{\text{hinge}}(|w^*\rangle, S) = \sum_{t < \#S} \max(0, \overbrace{1 - \langle w^* | c^*(x_t) | x_t \rangle}^{\leq 0 \text{ for perfect } |w^*\rangle}).$$

Theorem

On any sequence of examples $S = |x_0\rangle, |x_1\rangle, \dots$ the perceptron algorithm makes at most

$$\min \left\{ r^2 \| |w^*\rangle \|_2^2 + 2L_{\text{hinge}}(|w^*\rangle, S) \mid |w^*\rangle \right\}$$

mistakes where $r = \max_t \| |x_t\rangle \|_2$.



Online to batch conversion

Suppose we are given an online learning algorithm \mathcal{A} making few errors.

Question

Can we transform it into a batch learning algorithm with good true error?

Machine learning:

*Online learning: Online to batch conversion

1. Pick a sample S of size $\frac{m}{\varepsilon}$ from D . sample. Then
2. Pick $t \leftarrow \{1, \dots, \#S\}$.
3. For each of the first t samples in S call the online learning algorithm \mathcal{A} , yielding the hypothesis h_t .

Consider the indicator random variable I_j telling whether \mathcal{A} is mistaken on the j -th

$$\begin{aligned}\mathbb{E}[\text{err}_D(h_t)] &= \mathbb{E}[I_t] = \frac{1}{\#S} \sum_{1 \leq j \leq \#S} \mathbb{E}[I_j] \\ &= \frac{1}{\#S} \mathbb{E}\left[\underbrace{\sum_{1 \leq j \leq \#S} I_j}_{\leq m}\right] \leq \frac{m}{\#S} = \varepsilon.\end{aligned}$$

Theorem (Online to batch via random stopping)

If an online algorithm \mathcal{A} with mistake-bound m is run on a sample S of size $\frac{m}{\varepsilon}$ and stopped at a random time between 1 and $\#S$, the expected error of the hypothesis h produced satisfies $\mathbb{E}[\text{err}_D(h)] \leq \varepsilon$. □

Machine learning:

*Online learning: Online to batch conversion

1. Run \mathcal{A} to get an initial hypothesis h_1 .
2. **For** $t = 0, \dots$ **do** 3–5
3. Draw a set of $n_t = \frac{1}{\varepsilon} \log \frac{1}{\delta_t}$ random examples and test whether h_t gets all of them correct.
4. If not, run \mathcal{A} with one of those examples to obtain h_{t+1} .
5. If so, quit.

We define δ_i slowly decreasing with i such that $\sum_{i \geq 0} \delta_i \leq \delta$. (Eg. $\delta_i = \frac{\delta}{(i+2)^2}$.) When the algorithm quits, if $\text{err}_D(h_t) \geq \varepsilon$ the chance that h_t passes all test is at most $(1 - \varepsilon)^{n_t} \leq \delta_t$. Thus in total the probability for a bad answer is at most $\sum_{i \geq 0} \delta_i \leq \delta$.

Theorem (Online to Batch via Controlled Testing)

Let \mathcal{A} be an online learning algorithm with mistake-bound m . Then this procedure will halt after $\mathcal{O}\left(\frac{m}{\delta} \log \frac{m}{\delta}\right)$ examples and with probability at least $1 - \delta$ will produce a hypothesis of error at most ε . □

*Expert advice

Suppose you have a bunch of expert rules, like

- ▶ ‘Red sky at night, shepherd’s delight. Red sky in the morning, shepherd’s warning.’
- ▶ ‘A red sky —in the morning or evening— is a result of high pressure air in the atmosphere trapping particles of dust or soot.’
- ▶ ‘If the groundhog sees its shadow on Groundhog Day (2 February), six weeks of winter remain.’
- ▶ ...

Question

Can we use such rules to find a good predictor?

Machine learning:

*Online learning: *Expert advice

- ▶ Suppose we have access to n sleeping experts h_0, \dots, h_{n-1} . Each either stays silent (ie. he sleeps) or makes a prediction.
- ▶ S_i : samples on which h_i makes a prediction.
- ▶ $\text{mistakes}(\mathcal{A}, S)$: number of mistakes an algorithm \mathcal{A} makes on a sequence of examples.
- ▶ Construct \mathcal{A} with $E(\text{mistakes}(\mathcal{A}, S_i))$ not much larger than the expert's error $\text{mistakes}(h_i, S_i)$.

Machine learning:

*Online learning: *Expert advice

Let \mathcal{H}_x denote the set of experts that make a prediction on x .

Combining sleeping experts algorithm

Input: $\varepsilon \in]0, 1[$

1. Initialize each expert h_i with a weight $w_i \leftarrow 1$.
2. **For** each sample x **do** 3–8
3. **Make prediction:** Let $w_x = \sum_{h_i \in \mathcal{H}_x} w_i$. Choose $h_i \leftarrow \mathcal{H}_x$ with probability $p_{ix} = \frac{w_i}{w_x}$ and predict $h_i(x)$.
4. **Receive feedback and update weights:**
5. **For** $h_j \in \mathcal{H}_x$: **if** $h_j(x)$ was incorrect **then** let $m_{jx} \leftarrow 1$ **else** let $m_{jx} \leftarrow 0$.
6. **For** $h_i \in \mathcal{H}_x$ **do** 7–8
7. $r_{ix} \leftarrow \frac{\sum_{h_j \in \mathcal{H}_x} p_{jx} m_{jx}}{1+\varepsilon} - m_{ix}$.
8. Update $w_i \leftarrow w_i(1 + \varepsilon)^{r_{ix}}$.

Machine learning:

*Online learning: *Expert advice

Let \mathcal{H}_x denote the set of experts that make a prediction on x .

Theorem

For any set of n sleeping experts h_0, \dots, h_{n-1} and for any sequence of examples S , the combining sleeping experts algorithm \mathcal{A} satisfies for all i

$$\mathbb{E}(\text{mistakes}(\mathcal{A}, S_i)) \leq (1 + \varepsilon) \cdot \text{mistakes}(h_i, S_i) + \mathcal{O}\left(\frac{\log n}{\varepsilon}\right)$$

where $S_i = \{x \in S \mid h_i \in \mathcal{H}_x\}$.



In other words: the algorithm is at most a little worse than each expert on its area. But it covers a possibly much larger set of situations.

Machine learning:

*Online learning: *Boosting

Task

Given a weak learner construct a strong learner.

Idea: Put more emphasis on wrongly classified samples.

Boosting algorithm

Input: Sample S .

1. Initialize weights $w_i \leftarrow 1$ for each $x_i \in S$, $w \leftarrow (w_i)_i$.
2. **For** $t = 1, 2, \dots, t_0$ **do** 3–4
3. Call the weak learner on the weighted sample (S, w) and obtain hypothesis h_t .
4. Multiply the weight of each sample that was misclassified by h_t by $\alpha = \frac{\frac{1}{2} + \gamma}{\frac{1}{2} - \gamma}$.
5. Output the classifier $\text{MAJ}(h_1, \dots, h_{t_0})$.

Result: t_0 large enough \Rightarrow training error zero.

Machine learning:

*Further current directions

- ▶ Semi-supervised learning.
Eg. few labelled images, but also myriads of unlabelled ones!
- ▶ Active learning.
The algorithm decides which data points it wants labelled.
- ▶ Multi-task learning.
- ▶ ...

Mind this

Machine learning is a garbage in garbage out process if you have no idea about what to predict or what features maybe useful. It will only be useful if you fit a model using meaningful features, meaningful outcome and this is what 80 percentage of work you will involve in reality, namely feature engineering.

Ernest Chan, 2015

Section overview

Organizational

Introduction

High-dimensional space

Gaussians in high dimensions

Eigenvalues and eigenvectors

Best-fit subspaces and SVD

Power method for SVD

Applications of SVD

Machine learning

***Clustering**

Introduction

k -means clustering

k -center clustering

Spectral clustering

Approximation stability

High-density clusters

*Kernel methods

*Sparse cuts & recursive clustering

*Spectral clustering applied to social networks

Summary / Outro

1. First you need to represent your data.

- ▶ Frequently, each item is given as a **vector in \mathbb{R}^d** .
Example: in a “bag of words” representation each entry counts how often a certain word occurs in the described document.
- ▶ An item may be a **vertex in a graph** with edges weighted by some measure of similarity. Typically symmetric and with triangle inequality, so a metric.
Example: Protein sequences. Distance may be the cost of transforming one into the other.

2. Before clustering you have to ask a question!

For a set of photos of individuals, say, you may cluster them according to the depicted person or according to the facial expression.

Form of cluster

► Center-based clusters

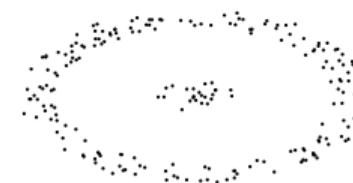
Clustering defined by k centers c_j , $j \in \mathbb{N}_{<k}$.
A data point belongs to a cluster C_j iff c_j is
the closest center to it. This partitions

$$A = \biguplus_{j \in \mathbb{N}_{<k}} C_j.$$



► High-density clusters

Idea: The density 'in' a cluster is high
compared to the density between clusters.



The 'ring' and the 'disc' seem to be the
clusters. But this cannot be center-based.

► k -center clustering. Minimize

$$\Phi_{k\text{-center}}(\mathcal{C}) = \max_{j \in \mathbb{N}_{<k}} \max_{a \in C_j} d(a, c_j).$$

► k -median clustering. Minimize

$$\Phi_{k\text{-median}}(\mathcal{C}) = \sum_{j \in \mathbb{N}_{<k}} \sum_{a \in C_j} d(a, c_j).$$

► k -means clustering. Minimize

$$\Phi_{k\text{-means}}(\mathcal{C}) = \sum_{j \in \mathbb{N}_{<k}} \sum_{a \in C_j} d(a, c_j)^2.$$

Naïve approaches and on

- ▶ If we require that cluster centers are data points:
Enumerate all $\binom{n}{k}$ possible sets and minimize Φ over them. Though slow, this is polynomial time wrt. n provided k is bounded.
Yet, for k -means the best cluster centers are centroids, usually not data points.
- ▶ Otherwise, when k is part of the input:
Then all the mentioned optimizitation problems are \mathcal{NP} -hard.

Thus all the following algorithms will involve

- ▶ some form of approximation or
- ▶ some additional assumption or
- ▶ both.

Spectral clustering

Idea

1. *Find the space V spanned by the top k right singular vectors of the matrix whose rows are the data points.*
2. *Project to V and cluster in this projection.*

Reasons

- ▶ Dimension reduction.
- ▶ Almost no loss in accuracy under mild assumptions about the data's distribution.

*Clustering:

k -means clustering

- ▶ Data points in \mathbb{R}^d .
- ▶ Minimize

$$\Phi_{k\text{-means}}(\mathcal{C}) = \sum_{j \in \mathbb{N}_{< k}} \sum_{a \in C_j} \|a - c_j\|^2$$

over clusterings $\mathcal{C} = (C_j)$ with centers (c_j) .

*Clustering:

k -means clustering

Motivation

Suppose: data generated according an equal mixture of k spherical well-separated Gaussians centered at μ_i with variance 1, ie. density

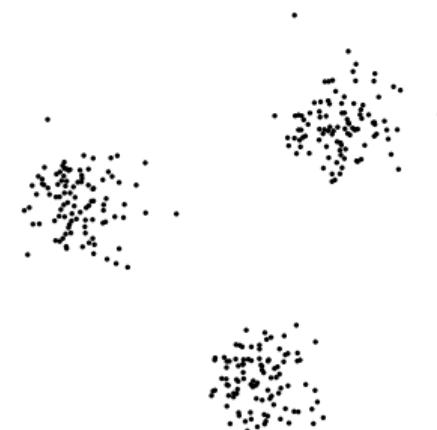
$$D(x) = \frac{1}{k} \sum_{i < k} \frac{1}{(2\pi)^{\frac{d}{2}}} e^{-\|x - \mu_i\|_2^2}.$$

Denote $\mu(x) = \underset{\mu \in \{\mu_i | i < k\}}{\operatorname{argmin}} \|x - \mu\|$ the center nearest to x .

Then the density to pick $(x^{(j)})_{j < n}$ is roughly

$$\prod_{j < n} D(x^{(j)}) \approx \frac{1}{(2\pi)^{\frac{nd}{2}} k^n} e^{-\underbrace{\sum_{j < n} \|x^{(j)} - \mu(x^{(j)})\|^2}_{=\Phi_{k\text{-means}}}}$$

which is maximized iff $\Phi_{k\text{-means}}$ is minimized.



*Clustering:

k -means clustering

Structural properties

Suppose the job is done, $A = \biguplus C_j$.

Question

What are the best centers?

*Clustering:

k -means clustering

Lemma (Centroid lemma)

Let A be a set of n points. The sum of their squared distances to any point x equals the sum of the squared distances to their centroid plus n times the squared distance from x to the centroid. That is,

$$\sum_{a \in A} \|a - x\|^2 = \sum_{a \in A} \|a_i - c\|^2 + n \|c - x\|^2$$

where $c = \frac{1}{n} \sum_{a \in A} a$ is the centroid of the set of points.

In particular, the sum of their squared distances to a point x is minimized iff x is the centroid $\frac{1}{n} \sum A$.

Proof. . . .

□

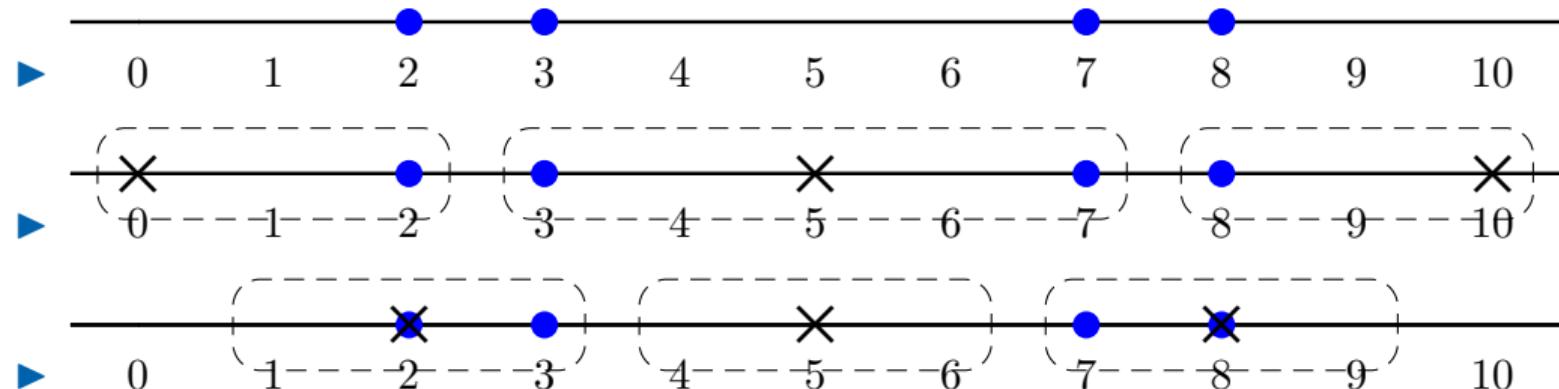
Lloyd's algorithm

1. Start with some set of k centers.
 2. **While** $\Phi_{k\text{-means}}$ shrinking considerably **do** 3–4
 3. Cluster each point with the center nearest to it.
 4. Replace the centers with the set of centroids.
-
- ▶ By the previous lemma each iteration must improve $\Phi_{k\text{-means}}$.
 - ▶ Thus the above process does converge.
 - ▶ ... sometimes to a local minimum only.
 - ▶ It may happen that some cluster becomes empty.

*Clustering:

k -means clustering

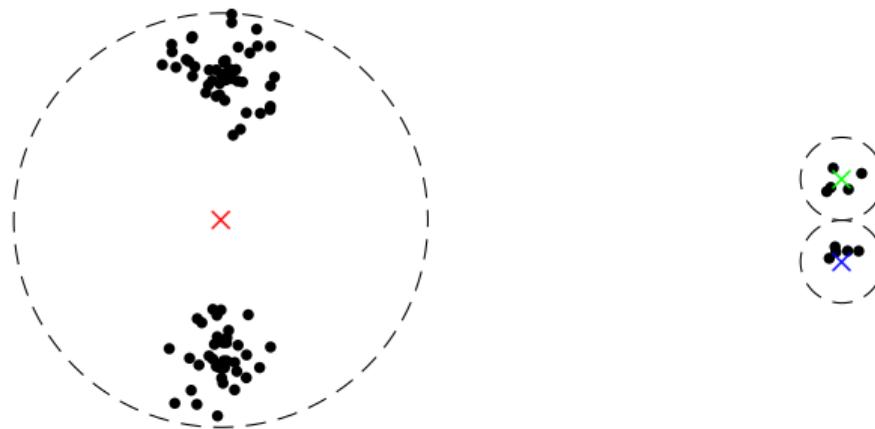
Example



*Clustering:

k -means clustering

Example



Here, Lloyd's algorithm finds a local but non-global minimum...

Farthest traversal

Initial centers can substantially influence the quality of the result.

Idea

1. *Pick first center c_0 from data points.*
2. **For** $i = 2, \dots, k$ **do** 3–3
3. *Pick c_i with largest sum of squared distances.*

Still, few outliers can do considerable harm.

Probabilistic farthest traversal

A probabilistic variant can mostly heal the harm done by a few outliers.

Idea

1. *Pick first center c_0 from data points.*
2. **For** $i = 2, \dots, k$ **do** 3–3
3. *Pick c_i randomly from data points weighted with sum of squared distances.*

*Clustering:

k -means clustering

A further approach —eg. after spectral clustering— uses Lloyd's algorithm to postprocess the result of any other approximation algorithm for the k -means problem.

Ward's algorithm

Idea

Define $\text{cost}(C) := \sum_{a \in C} d(a, c)^2$ for a cluster C with centroid c .

1. Initialize a clustering \mathcal{C} with each data point in its own cluster.
2. **While** $\#\mathcal{C} > k$ **do** 3–3
3. Merge two clusters C and C' minimizing $\text{cost}(C \cup C') - \text{cost}(C) - \text{cost}(C')$.

This is a greedy heuristic.

*Clustering:

k -means clustering

... on the line

Theorem

... can be done in polynomial time.

Namely, by dynamic programming (clever brute force):

1. Assume $A = (a_0, \dots, a_{n-1})$ is sorted.
2. **For** $i = 1, \dots, n$ **do** 10–10
3. **For** $k' = 1, \dots, k$ **do** 4–10
4. $\text{bestPhi} \leftarrow \infty$.
5. **For** $j = i - 1, \dots, 1$ **do** 6–9

6. $\Phi \leftarrow \Phi_{1\text{-means}}(a[j..i-1]) + \underbrace{\Phi_{(k'-1)\text{-means}}(a[0..j-1])}_{\text{Known!}}$.
7. **If** $\Phi < \text{bestPhi}$ **then**
8. $\text{bestPhi} \leftarrow \Phi$,
9. $C^{(i,k')} \leftarrow C^{(j,k'-1)} \cup [j..i]$.
10. $\Phi_{k'\text{-means}}(a[0..i-1]) \leftarrow \text{bestPhi}$.
11. **Return** $\Phi_{k\text{-means}}(a)$.

Runtime: $\mathcal{O}(kn^2)$.

*Clustering:

k -center clustering

- ▶ Data points in \mathbb{R}^d .
- ▶ Minimize

$$\Phi_{k\text{-center}}(\mathcal{C}) = \max_{j \in \mathbb{N}_{<k}} \max_{a \in C_j} \|a - c_j\|$$

over clusterings $\mathcal{C} = (C_j)$ with centers (c_j) .

Definition

The radius $\Phi_{k\text{-center}}(\mathcal{C})$ of a clustering \mathcal{C} is the maximum distance of any point to its cluster center.

In other words: there is a k -clustering of radius r iff there are k balls of radius r which cover all points.

Farthest traversal k -center clustering

Idea

1. Pick first center c_0 from data points.
2. For $i = 2, \dots, k$ do 3–3
3. Pick c_i with largest sum of squared distances.

Theorem

If there is a k -clustering of radius $\frac{r}{2}$, then the above algorithm finds a k -clustering with radius at most r .

Proof. . . .



Task

Achieve low errors only (rather than no errors).

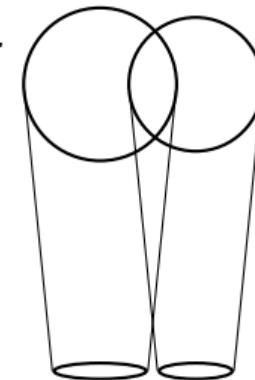
*Clustering:

Spectral clustering

- ▶ Data points in \mathbb{R}^d , stored as rows in an $n \times d$ matrix A .
- ▶ Minimize $\Phi_{k\text{-means}}(\mathcal{C}) = \sum_{j \in \mathbb{N}_{<k}} \sum_{a \in C} \|a - c_j\|^2$.

Idea

1. Find the space V spanned by the top k right singular vectors of A .
2. Project data points into V .
3. Cluster the projected points.



- ▶ Seen: Perfect for a mixture of spherical Gaussians.
- ▶ Now: general data.
- ▶ Intuition: Projection brings data points closer to cluster centers.
Sounds mysterious? See earlier...

Notation

Let C be the matrix whose i -th row is the center of the cluster to which the i -th row of A belongs to.

Then the k -means objective

$$\Phi_{k\text{-means}}(A) = \sum_i \|A_{i\cdot} - C_{i\cdot}\|^2 = \|A - C\|_F^2$$

equals the squared Frobenius norm of $A - C$.

*Clustering:

Spectral clustering

We will see that the projection reduces the sum of squared distances

- ▶ from $\|A - C\|_F^2 = \sum_t \sigma_t^2(A)$
- ▶ to $8k$ times $\|A - C\|_2^2 = \sigma_0^2(A)$.
- ▶ Often $\|A - C\|_F^2 \gg k \cdot \|A - C\|_2^2$.

Theorem

Let A be an $n \times d$ matrix with the projection A_k of the rows of A to the subspace of the first k right singular vectors of A . For any matrix C of rank less than or equal to k

$$\|A_k - C\|_F^2 \leq 8k \cdot \|A - C\|_2^2.$$

Proof. . . .

□

Spectral clustering algorithm

Define $\sigma(\mathcal{C}) := \frac{1}{\sqrt{n}} \|A - C\|_2$, $\varepsilon(\mathcal{C}) := \min_j \frac{\#\mathcal{C}_j}{n}$.

1. Find the space V spanned by the top k right singular vectors of A .
2. Project data points into V to get A_k .
3. **For** k times **do** 4–4
4. Select a random unclustered row from A_k and form a cluster with all rows of A_k at distance less than $\frac{6k}{\varepsilon}\sigma(\mathcal{C})$.

Theorem

If in a k -clustering \mathcal{C} every cluster has **at least εn points** in it and every pair of centers is separated by **at least $\frac{15k}{\varepsilon}\sigma(\mathcal{C})$** then **with probability at least $1 - \varepsilon$** the spectral clustering algorithm finds a clustering \mathcal{C}' that differs from \mathcal{C} on **at most $\varepsilon^2 n$ points**.

Proof. . .



*Clustering:

Spectral clustering

Our variance for clustering is

$$\begin{aligned}\sigma^2(C) &= \frac{1}{n} \|A - C\|_2^2 \\ &= \frac{1}{n} \max \left\{ \|(A - C)v\|^2 \mid \|v\| = 1 \right\} \\ &= \max \left\{ \frac{1}{n} \sum_i \left((A_{i\cdot} - C_{i\cdot}) \cdot v \right)^2 \mid \|v\| = 1 \right\}\end{aligned}$$

- ▶ This is similar to $\text{var } X \sim \frac{1}{n} \sum_i (x_i - \mathbb{E} X)^2$, where $x_i \leftarrow \otimes X$ are n samples of X .
- ▶ The vector v just picks the direction where the result is maximal.

*Clustering:

Spectral clustering

Briefly, the theorem says:

If cluster centers in \mathcal{C} are separated by $\Omega(\sigma(\mathcal{C}))$ then the spectral clustering algorithm finds \mathcal{C}' which differs from \mathcal{C} only in a small fraction of data points.

This mimicks the 1-dimensional situation where six standard deviations of seperation between Gaussians were sufficient to separate them with high certainty.

*Clustering:

Approximation stability

Problem: Most of the time, finding an optimal clustering is \mathcal{NP} -hard.

Example

- ▶ Given: news articles with some clever distance measure $d(x, y)$.
Find ‘correct’ clustering: ‘ \mathcal{NP} -hard’.
- ▶ Given: news articles by $\|\cdot\|_1$ -normalized ‘bag of words’ vectors.
Minimizing k -means score: \mathcal{NP} -hard.
- ▶ Given: news articles by $\|\cdot\|_1$ -normalized ‘bag of words’ vectors.
Approximating k -means score within 10%: \mathcal{NP} -hard.

Task

Weaker goal: Approximate clustering, say within 10%, 90% of the time.

Formal model

Definition

Given a dataset A and a cost measure Φ , such as k -means or k -median.

- ▶ For two k -clustering $\mathcal{C} = \{C_i \mid i < k\}$, $\mathcal{C}' = \{C'_i \mid i < k\}$ we define their *distance*

$$\text{dist}(\mathcal{C}, \mathcal{C}') := \min_{\sigma \in S_k} \frac{1}{n} \sum_{i < k} \#(C_i \setminus C'_{\sigma(i)})$$

where σ runs over permutations of $\mathbb{N}_{<k}$.

- ▶ Let $\mathcal{C}^* = \operatorname{argmin}_{\mathcal{C}} \Phi(\mathcal{C})$ be an (or the) optimal clustering.
- ▶ A data set A satisfies **(c, ε)-approximation-stability** with respect to the objective Φ if each clustering with $\Phi(\mathcal{C}') \leq c\Phi(\mathcal{C}^*)$ satisfies $\text{dist}(\mathcal{C}', \mathcal{C}^*) \leq \varepsilon$.

We will focus on the objective k -median $\Phi_{k\text{-median}}$. $\Phi_{k\text{-median}}(\mathcal{C}) = \sum_{j \in \mathbb{N}_{<k}} \sum_{a \in C_j} d(a, c_j)$.

For a data set A with optimal clustering \mathcal{C}^* define

- ▶ $w(a)$ the distance of a to the center of its cluster in \mathcal{C}^* ,
- ▶ $w_2(a)$ the distance of a to the second-closest center of \mathcal{C}^* .

Put $w_{\text{avg}} = \frac{1}{n} \sum_{a \in A} w(a)$. Notice that $\Phi_{k\text{-means}}(A) = nw_{\text{avg}}$.

Lemma

Assume the data set A satisfies (c, ε) -approximation-stability with respect to $\Phi_{k\text{-median}}$ and each cluster in \mathcal{C}^* has size at least $2\varepsilon n$. Then

1. Fewer than εn points a have $w_2(a) - w(a) \leq \frac{c-1}{\varepsilon} w_{\text{avg}} = 5d_{\text{crit}}$.
2. At most $\frac{5\varepsilon}{c-1} n$ points a have $w(a) \geq \frac{c-1}{5\varepsilon} w_{\text{avg}} =: d_{\text{crit}}$.

Proof. . .

□

*Clustering:

Approximation stability

- ▶ Call the points satisfying one of the lemma's cases **bad points**.
- ▶ The lemma says that a (c, ε) -approximation-stable data set has at most $\varepsilon n + \frac{5\varepsilon}{c-1}n$ bad points.
- ▶ Put $d_{\text{crit}} := \frac{c-1}{5\varepsilon}w_{\text{avg}}$. Good points have at most this distance to their center and at least $5d_{\text{crit}}$ to any other center.

Idea

Create a graph G with the points from A as vertices and edges between any two points a, b with $d(a, b) < 2d_{\text{crit}}$.

- ▶ By the triangle inequality, good points within some cluster have distance at most $2d_{\text{crit}}$. All good points in a cluster will thus form a clique in G .
- ▶ Any edge between different clusters must be between two bad points.

*Clustering:

Approximation stability

Put $m = \varepsilon n + \frac{5\varepsilon}{c-1}n$, so there are at least m bad points. Assume that each cluster has at least $2m+1$ points.

- ▶ Thus each cluster has at least $m+1$ good points.
- ▶ Create a graph H as follows: connect two vertices a and b if a and b share at least $m+1$ neighbors (themselves included).
- ▶ Since good points within a cluster form a clique in G and each cluster has at least $m+1$ good points the good points form again a clique in H .
- ▶ Points in different clusters are not connected

in H . There are at most m bad points.

- ▶ Take the k largest components in H : \mathcal{C}' .
- ▶ These are subsets of clusters in \mathcal{C}^* and at most bad points remain, ie. no more than m .
- ▶ Assign the remaining points: for each point $a \in A \setminus \bigcup \mathcal{C}'$ find the cluster $C' \in \mathcal{C}'$ minimizing median $\{d(a, b) \mid b \in C'\}$. Add a to this C' .
- ▶ In the resulting clustering \mathcal{C} , the points from $\bigcup \mathcal{C}'$ and the points with $w_2(a) - w(a) > 2d_{\text{crit}}$ are correctly clustered.
- ▶ Only the at most εn points with $w_2(a) - w(a) < 2d_{\text{crit}}$ may be wrongly attributed.

Algorithm k -median stability

Input: data set A , approximation stability parameters (c, ε) and critical distance d_{crit} .

1. Put $m \leftarrow \varepsilon n + \frac{5\varepsilon}{c-1}n$.
2. Create a graph G with
 - ▶ a vertex for each data point in A and
 - ▶ an edge between vertices a and b iff $d(a, b) < 2d_{\text{crit}}$.
3. Create a graph H with
 - ▶ same vertices than G and
 - ▶ an edge between vertices a and b iff they share at least $m + 1$ neighbors, self included.
4. Let \mathcal{C}' consist of the k largest components in H .
5. Assign each point $a \in A \setminus \bigcup \mathcal{C}'$ to the cluster $C' \in \mathcal{C}'$ of smallest median distance median $\{d(a, b) \mid b \in C'\}$.

Theorem

Given a data set A with k -median optimal clustering \mathcal{C}^* . Assume:

- ▶ The data set A satisfies (c, ε) -approximation-stability with respect to the k -median objective.
- ▶ Each cluster in \mathcal{C}^* has size at least $2m + 1$ with $m = \varepsilon n + \frac{5\varepsilon}{c-1}n$.

Then the algorithm k -median stability finds a clustering \mathcal{C} such that $\text{dist}(\mathcal{C}, \mathcal{C}^*) \leq \varepsilon$. □

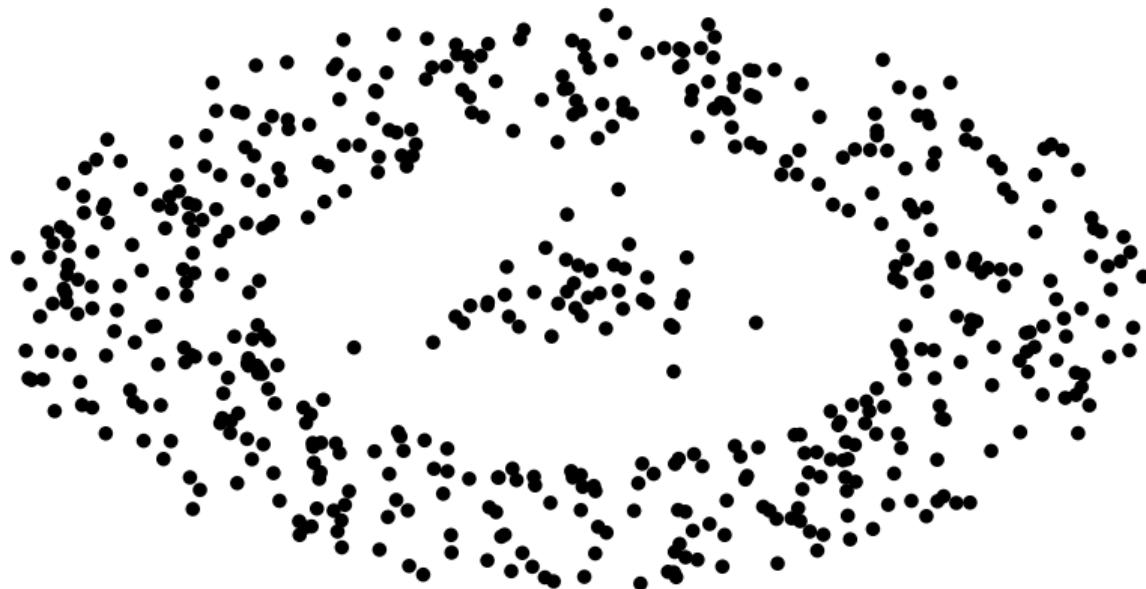
Bottom line

If A is nice then we can approximate the correct clustering.

*Clustering:

High-density clusters

Another task



Algorithm single linkage agglomeration

1. Initialize a clustering \mathcal{C} with single point clusters.
2. **While** $\#\mathcal{C} > k$ **do** 3–4
3. Find two clusters $C, C' \in \mathcal{C}$ minimizing
$$d_{\min}(C, C') = \min \{d(x, y) \mid x \in C, y \in C'\}.$$
4. Merge these two clusters into one.

Theorem

Suppose the desired k -clustering \mathcal{C}^* satisfies the property that there exists some distance σ such that

1. any two data points in different clusters have distance at least σ and
2. for any cluster C^* and any partition of C^* into two non-empty sets B and $C^* \setminus B$ there exist points on each side of the partition of distance less than σ .

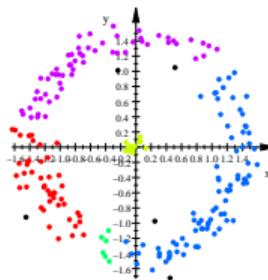
Then single-linkage agglomeration will correctly recover the clustering \mathcal{C}^* .

Proof. . . .

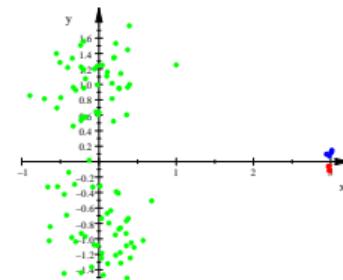
□

*Clustering:

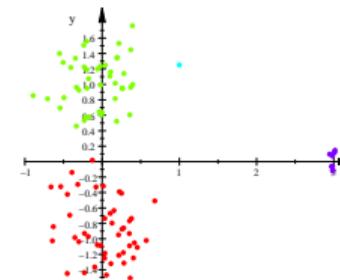
Some examples



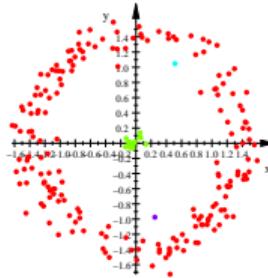
Single linkage $k = 10$



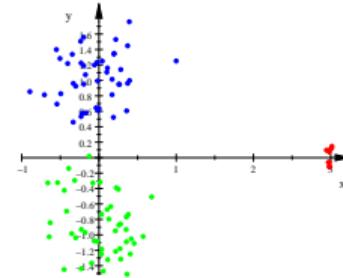
Lloyd's algorithm $k = 3$



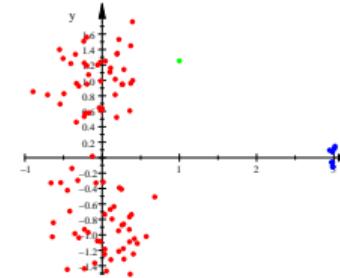
Single linkage $k = 4$



Single linkage $k = 4$



Ward's algorithm $k = 3$



Single linkage $k = 3$

Robust linkage

- ▶ Single linkage algorithm is fairly brittle.
- ▶ More robust: generalized Wishart's algorithm.

Input: Data set A , threshold t , parameter α .

1. **For** $r = 0 \dots \infty$ **do** 2–3
2. Construct a graph G_r with nodes $\{x \in A \mid \#(B(x, r) \cap A) \geq t\}$.
Add edge (x, y) iff $\|x - y\| \leq \alpha r$.
3. Let $\mathcal{C}(r)$ be the connected components of G_r .

- ▶ For $\alpha = 1$, $t = 2$ this is single linkage.
- ▶ Wishart suggested $\alpha = 1$ and larger t .
- ▶ Chaudhuri & Dasgupta (2010) prove that $\alpha = 2$ and $t = \Theta(d \log n)$ are sufficient to obtain reasonable success guarantees.

- ▶ Center-based clustering always means that any two clustered are linearly separated.
- ▶ Otherwise... kernels!
- ▶ Recall $k(|x\rangle, |y\rangle) = \langle \varphi(x) | \varphi(y) \rangle$ for some embedding $\varphi: \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$ into some possibly much higher dimensional space.
- ▶ Then we may consider the distance $d(|x\rangle, |y\rangle)$ as the distance after the embedding. Thus

$$\begin{aligned} d(|x\rangle, |y\rangle)^2 &= \| |\varphi(x)\rangle - |\varphi(y)\rangle \|_2^2 \\ &= \langle \varphi(x) | \varphi(x) \rangle + \langle \varphi(y) | \varphi(y) \rangle - 2 \langle \varphi(x) | \varphi(y) \rangle \\ &= k(|x\rangle, |x\rangle) + k(|y\rangle, |y\rangle) - 2k(|x\rangle, |y\rangle). \end{aligned}$$

And so we can express the distance merely based on the kernel!

- ▶ We can now run a center-based clustering algorithm on these new distances.
- ▶ A popular kernel function is the Gaussian kernel:

$$k(|x\rangle, |y\rangle) = e^{-\frac{1}{2\sigma^2} \||x\rangle - |y\rangle\|_2^2}.$$

It emphasizes closeness of points.

*Clustering:

*Sparse cuts & recursive clustering

Opposing agglomeration

Consider data as nodes in an undirected connected graph G and edges indicate similarity.

Idea

Start with all vertices in one cluster and split it up stepwise by sparse cuts.

This leads to recursive clustering...

*Clustering:

*Sparse cuts & recursive clustering

Given a graph $G = (V, E)$ mit n vertices and m edges, $E \subset \binom{V}{2}$.

- ▶ Define

$$d(S, T) := \sum_{x \in S} \sum_{y \in T} a_{xy},$$

where a_{xy} counts the number of edges from x to y .

- ▶ The sum $d(S)$ of the degrees $d(x) = d(\{x\}, V)$ of vertices in S is given by

$$d(S) := d(S, V) = \sum_{x \in S} d(x).$$

Notice that $d(S \uplus T) = d(S) + d(T)$.

- ▶ The conductance

$$\Phi(S, T) = \frac{d(S, T)}{\min(d(S), d(T))}$$

measures the relative strength of similarities between S and T .

Algorithm (Recursive clustering)

Input: Graph $G = (V, E)$, quality requirement ε .

1. Initialize a clustering $\mathcal{C} \leftarrow \{V\}$ with a single cluster.
2. **While** there is a splittable $W \in \mathcal{C}$ **do** 3–4
3. **If** W has a subset S with
 - ▶ $d(S) \leq \frac{1}{2}d(W)$ and
 - ▶ $\Phi(S, W \setminus S) \leq \varepsilon$**then**
4. Split W into two clusters S and $W \setminus S$.

*Clustering:

*Sparse cuts & recursive clustering

Theorem

At termination of Recursive Clustering, the total number of edges between vertices in different clusters is at most $\mathcal{O}(\varepsilon m \ln n)$.

Proof. . . .



*Clustering:

*Sparse cuts & recursive clustering

- ▶ The theorem does not yet lead to a fast algorithm as we would have to compute

$$\min_{S \subseteq W} \Phi(S, W \setminus S)$$

which is an \mathcal{NP} -hard problem.

- ▶ However, eigenvalues and eigenvectors can be used for an approximate answer.
- ▶ And they can be approximated fast.
- ▶ ...

*Clustering:

*Spectral clustering applied to social networks

Finding communities in social networks

... is different:

- ▶ Want: communities of tiny size, say 20–50 within 100 000 000 vertices.
- ▶ Typically, a person (node) is in a number of *overlapping communities*, ie. **not** disjoint clusters.
- ▶ Often various levels of structure.
A set of dominant communities may hide a set of weaker communities that are of interest.

Spectral clustering offers solutions here...

Section overview

Organizational

Introduction

High-dimensional space

Gaussians in high dimensions

Eigenvalues and eigenvectors

Best-fit subspaces and SVD

Power method for SVD

Applications of SVD

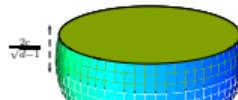
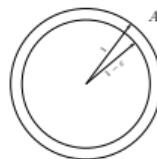
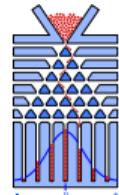
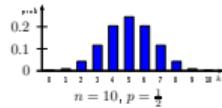
Machine learning

*Clustering

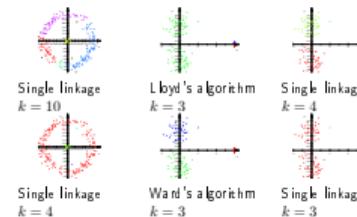
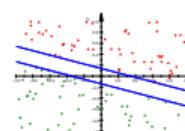
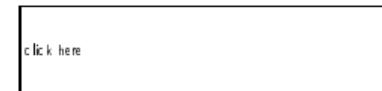
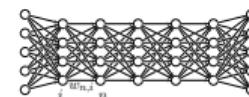
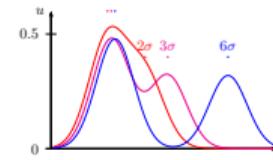
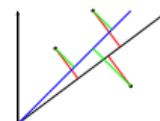
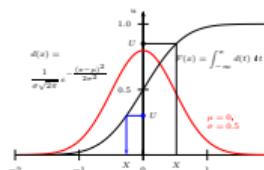
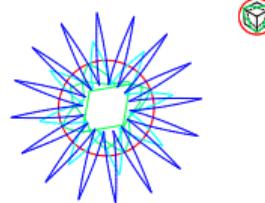
Summary / Outro

Summary / Outro

$\{\square, \blacksquare, \blacksquare, \blacksquare, \blacksquare, \blacksquare, \blacksquare\}$



$$A_{n \times d} = \begin{matrix} U_{n \times r} \\ D_{r \times r} \\ V^T_{r \times d} \end{matrix}$$



Summary / Outro

Organizational

Webpage & mailing list

Digital teaching

Time & place

Hand-in & exam

Literature

Introduction

High-dimensional space

Probabilities

The law of large numbers

Tail bounds

Geometry of high dimensions

Properties of the unit ball

Volume near the equator

Gaussians in high dimensions

Generating points uniformly at random
from a ball

Interludium: Inverse transform sampling

Gaussians

Fitting a spherical Gaussian to data

Separating Gaussians

Eigenvalues and eigenvectors

Basics

Symmetric matrices

Extremal properties of eigenvalues

Eigenvalues of the sum of two symmetric
matrices

Norms

Additional linear algebra

Best-fit subspaces and SVD

Introduction

Singular vectors

Singular value decomposition (SVD)

Best rank- k approximation

Left singular vectors

Power method for SVD

Applications of SVD

Centering data

Unmixing a mixture of spherical
Gaussians

Principal component analysis

Ranking documents and web pages

Machine learning

Introduction

The perceptron algorithm

Kernel functions and non-linearly separa-
ble data

Generalizing to new data

VC-dimension

VC-dimension and generalizing

*Deep learning

*Online learning

*Further current directions

*Clustering

Introduction

k -means clustering

k -center clustering

Spectral clustering

Approximation stability

High-density clusters

*Kernel methods

*Sparse cuts & recursive clustering

*Spectral clustering applied to social
networks

Summary / Outro