02462 – Signals and data

Technical University of Denmark,
DTU Compute, Institut for Matematik og Computer Science.

Overview

Representations

Vectors as Feature Sets

We should distinguish between,

- the *subject* $s \in S$ of our study, from some *population* S.
- the *vector representation* $\boldsymbol{X}(s)$, which breaks down into *features* $X_k(s)$.

For a *survey*,

S is the *population* of interest, e.g. all citizens of voting age.

s is a *human subject* from the population.

X_q is a random variable representing the subjects *response to a single question*.

$\boldsymbol{X}$ is the random vector containing all features, representing the *full survey response* for a single subject.

$$\boldsymbol{X}_{\text{survey}}(s_{\text{person}}) = \begin{pmatrix} X_{\text{income}}(s_{\text{person}}) \\ X_{\text{education}}(s_{\text{person}}) \\ \vdots \\ X_{\text{age}}(s_{\text{person}}) \end{pmatrix}$$

Note that even an enormous survey would still be a very *incomplete representation* of a subject as complex as a human.

Data Matrix

- A *dataset* consists of N vector observations $\{\boldsymbol{x}_n\}_{n=1}^N$.
- each *observation* (or *datapoint*) $\boldsymbol{x}_n$ is a D-dimensional vector.
- each *dimension* x_{nd} of an observation is a *feature*.

Finally we can collect all of the datapoints and features into a *data matrix*,

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{x}_1^\top \\ \vdots \\ \boldsymbol{x}_N^\top \end{pmatrix} = \begin{pmatrix} x_{11} & \dots & x_{1D} \\ \vdots & \ddots & \vdots \\ x_{N1} & \dots & x_{ND} \end{pmatrix}, \qquad \text{(data matrix)}$$

so each row is an *observation* and each column is a *feature*.

The Generative Story

Last week we looked at how we could randomly *generate data* following a recipe,

1. We speculate that each data point x was a random sample from a *probability distribution* $p(x|\theta)$.
2. We pick a random *parameter* θ.
3. Finally, we use θ to draw samples $x_1, \ldots, x_N$ from $p(x|\theta)$.

Generative Models

"All models are wrong, but some are useful." — George Box

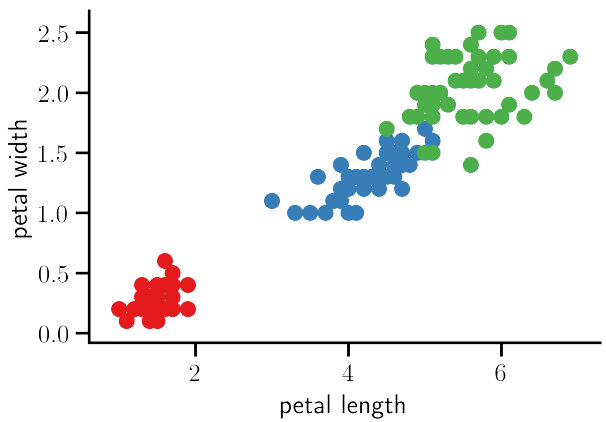
For real-world data we do not *know* how the data is generated.

Choose a *generative model* $p(x|\theta)$.

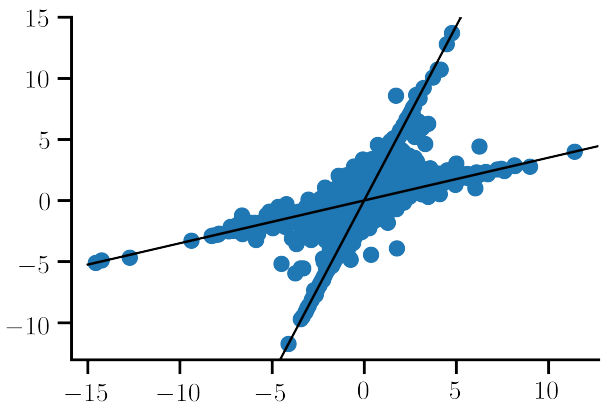
Pretend data is sampled from $p(x|\theta)$.

Learn by finding the θ that best describes the data.

How to choose $p(x|\theta)$? Pick a model with the *structure* you are interested in.



(a) Do you want to find *clusters*? Pick a model that samples from clusters.



(b) Do you want to find *subspaces*? Pick a model that samples from subspaces.

A Normal Example

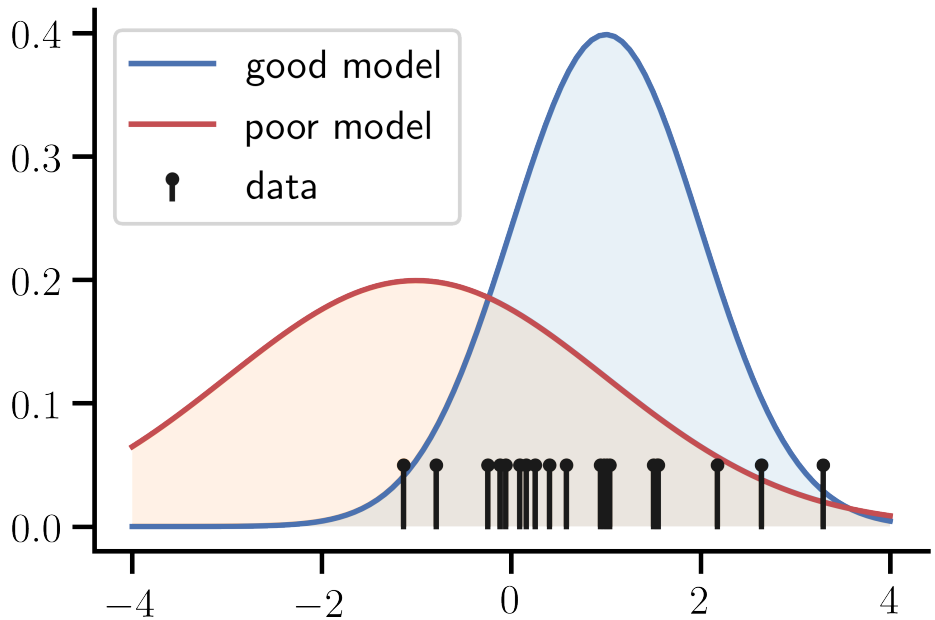
Problem *You have collected test scores t_n for a new intelligence test. How could you learn about the mean and the spread of the test scores?*

Following the recipe from before,

Choose a normal distribution $\mathcal{N}(\mu, \sigma^2)$ as knowing μ and σ^2 tells us about the data's shape.

Pretend that $x_n \sim \mathcal{N}(\mu, \sigma^2)$ which is reasonable as normals occur frequently in nature.

Learn by finding the best μ and σ^2, which tells us about the *mean* and *variance* of the data.



Learning about Your Data

We saw that by finding the best parameters $\theta = \{\mu, \sigma^2\}$ for the normal, we learned something about the data.

Parameters = Structure

In generative models learning the *parameters* θ tells us about the data,

Linear Regression $\theta = \{w_0, w_1\}$ told us about the line's slope and offset.

Cluster Models θ can encode the shape and location of clusters.

Subspace Models θ can encode the position and orientation of the subspaces.

So what is the "best parameter"?

- this is what *inference* from last week tried to answer.
- we pick the θ that has the highest chance of *generating* the data — the most probable parameter.

Maximum Likelihood

Posterior

Using Bayes' theorem, we can find the probability of the parameters θ given the data $\mathcal{D}$,

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}}$$

- If we assume *all models are equally likely*, or we have *no prior information*, we can use a prior that gives all models equal weight:

$$p(\boldsymbol{\theta}) \propto 1 \qquad \text{(no information)}$$

- This makes the posterior proportional to the likelihood,

$$p(\boldsymbol{\theta}|\mathcal{D}) = \frac{p(\mathcal{D}|\boldsymbol{\theta}) \cdot 1}{p(\mathcal{D})} \propto p(\mathcal{D}|\boldsymbol{\theta})$$

Maximum Likelihood

- finding the best model of your data requires solving an optimization problem.

Most Probable Parameter

If we get a dataset $x_1, \ldots, x_N$ how do we then find the θ^*?

- We look for the *most probable parameter*.

In practice, we can solve for the *maximum likelihood estimator*,

$$\theta^* \underbrace{\approx}_{\text{approximates}} \theta_{\text{ML}} = \underbrace{\arg\max_{\theta}}_{\text{find best } \theta} \prod_{n=1}^{N} p(x_n|\theta) = \arg\min_{\theta} \underbrace{-\sum_{n=1}^{N} \ln p(x_n|\theta)}_{\text{negative log-likelihood}}$$

Finding the Most Probable Normal

If data is generated from a normal distribution $x_n \sim \mathcal{N}(\mu, \sigma^2)$ then the negative log-likelihood is

$$-\ln L(\mu, \sigma^2) = -\ln \prod_{n=1}^{N} \underbrace{\mathcal{N}(x_n; \mu, \sigma^2)}_{\text{single likelihood}} = \frac{1}{2} \sum_{n=1}^{N} \left(\frac{(x_n - \mu)^2}{\sigma^2} + \ln 2\pi\sigma^2 \right).$$

To find the maximum likelihood solution we need to solve

$$\frac{\partial(-\ln L)}{\partial \mu} = 0$$
$$\frac{\partial(-\ln L)}{\partial \sigma^2} = 0$$

Finding the Most Probable Normal — the Mean Parameter

Likelihood

$$-\ln L(\mu, \sigma^2) = \frac{1}{2} \sum_{n=1}^{N} \left(\frac{(x_n - \mu)^2}{\sigma^2} + \ln \sigma^2 + \ln 2\pi \right).$$

To find the best μ, we take the derivative and set it to 0,

$$\frac{\partial(-\ln L)}{\partial \mu} = \frac{1}{2} \sum_{n=1}^{N} \left(\frac{1}{\sigma^2} \frac{\partial (x_n - \mu)^2}{\partial \mu} + 0 \right) = \frac{1}{2\sigma^2} \left(N\mu - \sum_{n=1}^{N} x_n \right) = 0$$

Solving for μ then gives us the maximum likelihood estimator μ_{ML}

$$\mu_{\text{ML}} = \frac{1}{N} \sum_{n=1}^{N} x_n$$

This is just the *sample mean*!

Finding the Most Probable Normal — the Variance Parameter

Likelihood

$$-\ln L(\mu, \sigma^2) = \frac{1}{2}\sum_{n=1}^{N}\left(\frac{(x_n-\mu)^2}{\sigma^2} + \ln\sigma^2 + \ln 2\pi\right).$$

To find the best σ^2, we take the derivative and set it to 0,

$$\frac{\partial(-\ln L)}{\partial\sigma^2} = \frac{1}{2}\sum_{n=1}^{N}\left((x_n-\mu)^2\frac{\partial 1/\sigma^2}{\partial\sigma^2} + \frac{\partial\ln\sigma^2}{\partial\sigma^2}\right) = \frac{1}{2}\left(-\sum_{n=1}^{N}\frac{(x_n-\mu)^2}{(\sigma^2)^2} + \frac{N}{\sigma^2}\right) = 0$$

Now we need to plug in μ_{ML} from before, and then we can solve to find the maximum likelihood variance parameter σ^2_{ML}

$$\sigma^2_{\text{ML}} = \frac{1}{N}\sum_{n=1}^{N}(x_n-\mu_{\text{ML}})^2$$

This is just the *sample variance*!

Standardization

Remember that if $Z \sim \mathcal{N}(0,1)$, then we can shift and scale to get any other normal distribution

$$X = \mu + \sigma Z \Rightarrow Y \sim \mathcal{N}(\mu, \sigma^2).$$

What if we do this in reverse?

Z-score

For data x_n we can use our best approximation $\mathcal{N}(\mu_{\text{ML}}, \sigma_{\text{ML}})$ to standardize the variable,

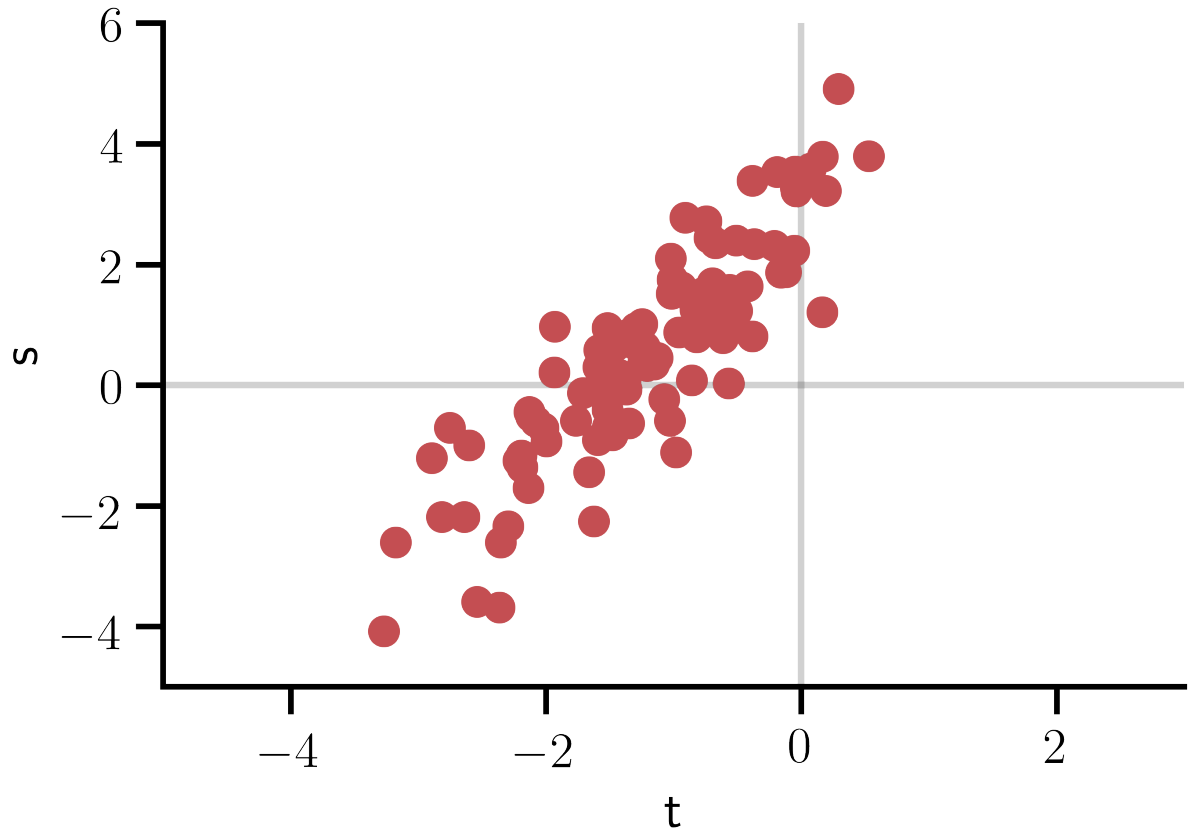
$$z_n = \frac{x_n - \mu_{\text{ML}}}{\sigma_{\text{ML}}}.$$

Why is the Z-score z_n interesting?

- $\frac{1}{N}\sum_n z_n = 0$ and $\frac{1}{N}\sum_n z_n^2 = 1$, so it has has mean/variance like $\mathcal{N}(0,1)$.
- If we know z_n and the parameters, we can reconstruct x_n — *z_n carries all the information!*
- If y_n has units, z_n is *unit-free* — we can compare variables measured in different ways.

High Dimensional Setting

Problem *You add a new test score s_n to your intelligence test data t_n from before. What can you now say about (t_n, s_n)?*



What if we take our *generative model* to be a *multivariate normal* $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$?

The Most Probable... Matrix?

$$\underbrace{-\ln L}_{\substack{\text{negative}\\ \text{log-likelihood}}} = \sum_{n=1}^{N} \ln \underbrace{p(\boldsymbol{x}_n|\boldsymbol{\mu}, \boldsymbol{\Sigma})}_{\text{single likelihood}} = -\sum_{n=1}^{N} \frac{1}{2}(\boldsymbol{x}_n - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}) + \frac{D}{2}\ln 2\pi + \ln|\boldsymbol{\Sigma}|.$$

Then we need to solve the two equations

$$\frac{\partial(-\ln L)}{\partial \boldsymbol{\mu}} = \begin{pmatrix} \frac{\partial(-\ln L)}{\partial \mu_1} \\ \vdots \\ \frac{\partial(-\ln L)}{\partial \mu_D} \end{pmatrix} = \mathbf{0} \quad , \quad \frac{\partial(-\ln L)}{\partial \boldsymbol{\Sigma}} = \begin{pmatrix} \frac{\partial(-\ln L)}{\partial \Sigma_{11}} & \cdots & \frac{\partial(-\ln L)}{\partial \Sigma_{1D}} \\ \vdots & \ddots & \vdots \\ \frac{\partial(-\ln L)}{\partial \Sigma_{1D}} & \cdots & \frac{\partial(-\ln L)}{\partial \Sigma_{DD}} \end{pmatrix} = \mathbf{0}$$

But these now require derivatives in vectors and matrices! You will learn about vector derivatives in your mathematics course, so we will reveal that

$$\boldsymbol{\mu}_{\text{ML}} = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{x}_n \qquad \text{(maximum likelihood for } \mu\text{)}$$

which is — again — the *sample mean*. But what about $\boldsymbol{\Sigma}$?

A Soft Introduction to Matrix Calculus I

Likelihood

$$-\ln L(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \frac{1}{2}(\boldsymbol{x}_n - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}) + \frac{D}{2}\ln 2\pi + \ln|\boldsymbol{\Sigma}|.$$

You likely know rules for ordinary derivatives, but it turns out that there are similar rules for matrix derivatives!

$$\frac{\mathrm{d}}{\mathrm{d}\sigma}\frac{v^2}{\sigma} = -\frac{v^2}{\sigma^2} \qquad \text{(scalar version)}$$

$$\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\Sigma}}\boldsymbol{v}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{v} = -\boldsymbol{\Sigma}^{-1}\boldsymbol{v}\boldsymbol{v}^\top\boldsymbol{\Sigma}^{-1} \qquad \text{(matrix version)}$$

If you squint, the formula look very *similar*.
We can use this new *rule* to calculate the derivative of the first term,

$$\frac{1}{2}\sum_{n=1}^{N}\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\Sigma}}(\boldsymbol{x}_n - \boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}) = -\frac{1}{2}\sum_{n=1}^{N}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu})(\boldsymbol{x}_n - \boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}.$$

Pretty straightforward (if you know the right rule...)

A Soft Introduction to Matrix Calculus II

Likelihood

$$-\ln L(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = -\sum_{n=1}^{N} \frac{1}{2}(\boldsymbol{x}_n - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}) + \frac{D}{2}\ln 2\pi + \ln|\boldsymbol{\Sigma}|.$$

This leaves the second term $\ln|\boldsymbol{\Sigma}|$, but again there is a helpful rule.

$$\frac{\mathrm{d}}{\mathrm{d}\sigma}\ln\sigma = \frac{1}{\sigma} \qquad \text{(scalar version)}$$

$$\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\Sigma}}\ln|\boldsymbol{\Sigma}| = \boldsymbol{\Sigma}^{-1} \qquad \text{(matrix version)}$$

Putting it all together we have that,

$$\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\Sigma}}(-\ln L) = \frac{N}{2}\boldsymbol{\Sigma}^{-1} - \frac{1}{2}\sum_{n=1}^{N}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu})(\boldsymbol{x}_n - \boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1} = 0.$$

Then we just need to solve: multiply by $\boldsymbol{\Sigma}$ on both sides, and plug in $\boldsymbol{\mu}_{\mathrm{ML}}$

$$\boldsymbol{\Sigma}_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{x}_n - \boldsymbol{\mu}_{\mathrm{ML}})(\boldsymbol{x}_n - \boldsymbol{\mu}_{\mathrm{ML}})^\top$$

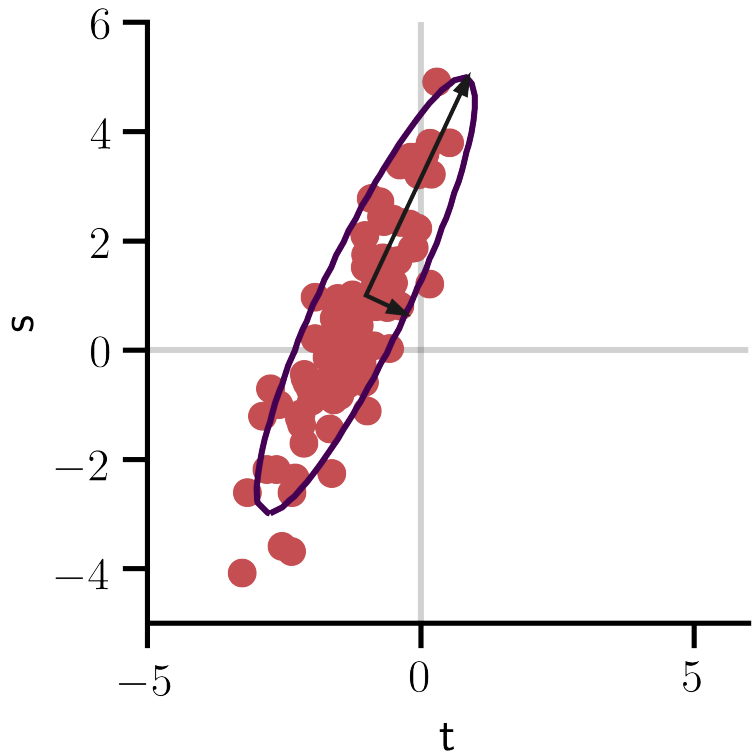
which is the *sample covariance matrix*!

The Sample Covariance Matrix and Standardization

If we look at the (i, j)'th element of $\mathbf{\Sigma}_{\text{ML}}$ we can see how it relates to variance

$$[\mathbf{\Sigma}_{\text{ML}}]_{ij} = \frac{1}{N}\sum_{n=1}^{N}(x_{ni} - [\mu_{\text{ML}}]_i)(x_{nj} - [\mu_{\text{ML}}]_j) \approx \mathbb{E}[(X_i - \mathbb{E}[X_i])(X_j - \mathbb{E}[X_j])]$$

Can we compute a standardization z_n which is distributed like a standard normal $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$?



Idea: project the data unto the *eigenbasis*

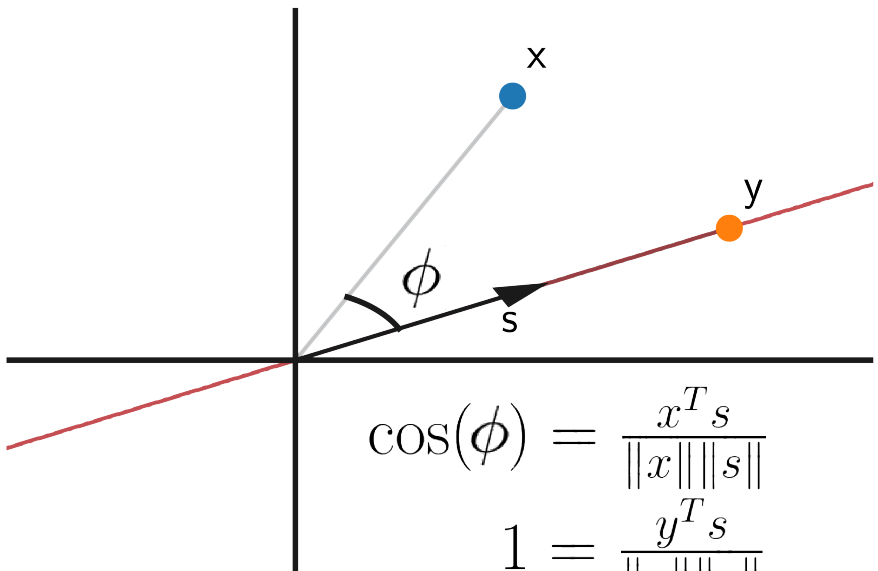
Angles between Vectors

- The *angle* θ between $\boldsymbol{x}$ and $\boldsymbol{y}$ is

$$\cos(\phi) = \frac{\boldsymbol{x}^\top \boldsymbol{y}}{\|\boldsymbol{x}\|\|\boldsymbol{y}\|}.$$

- the *norm* (or *length*) of a vector is,

$$\|\boldsymbol{x}\| = \sqrt{\sum_{d=1}^{D} x_d^2}$$



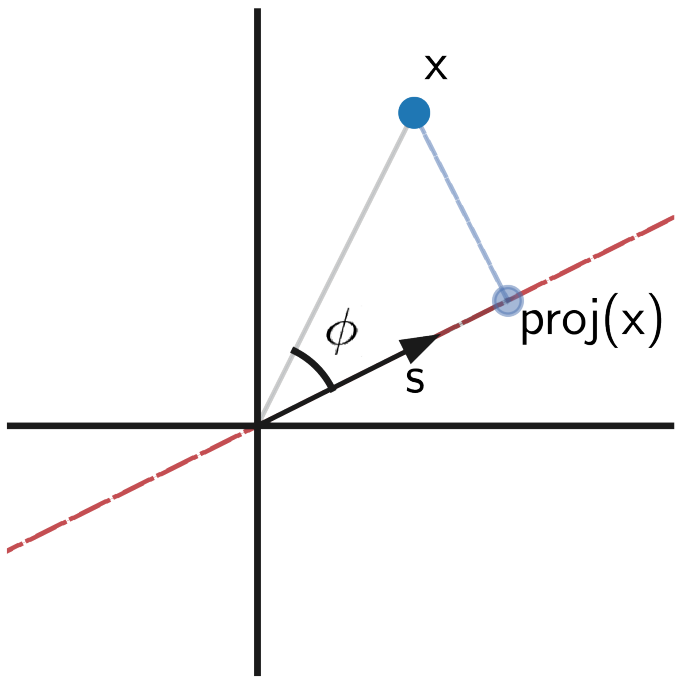
$$\cos(\phi) = \frac{x^T s}{\|x\|\|s\|}$$

$$1 = \frac{y^T s}{\|y\|\|s\|}$$

Projections

- We saw before that subspaces were spanned by *factors*.
- If we want to know how much of a datapoint is described by a factor, we can *project* $\boldsymbol{x}$ onto $\boldsymbol{v}$.

$$\text{proj}_v(\boldsymbol{x}) = \underbrace{\frac{\boldsymbol{v}}{\|\boldsymbol{v}\|}}_{\substack{\text{unit} \\ \text{factor}}} \cdot \underbrace{\|\boldsymbol{x}\|}_{\substack{\text{original} \\ \text{length}}} \cdot \underbrace{\cos(\phi)}_{\substack{\text{fraction parallel} \\ \text{to factor}}}$$



Projection and Basis Change

If $\boldsymbol{v}$ is a unit vector $\|\boldsymbol{v}\| = 1$ then projection is even easier,

$$\text{proj}_v(\boldsymbol{x}) = \boldsymbol{v}\|\boldsymbol{x}\| \cos(\phi) = \boldsymbol{v}(\boldsymbol{v}^\top \boldsymbol{x}).$$

- think of $\boldsymbol{v}^\top \boldsymbol{x}$ as the *"coordinate"* of $\boldsymbol{x}$ along $\boldsymbol{v}$.

So if we have a *basis matrix* $U = \begin{pmatrix} u_1 & u_2 & \dots & u_D \end{pmatrix}$ where $\|u_d\| = 1$

$$U^\top \boldsymbol{x} = \begin{pmatrix} u_1^\top \boldsymbol{x} \\ u_2^\top \boldsymbol{x} \\ \vdots \\ u_D^\top \boldsymbol{x} \end{pmatrix}$$

So $U^\top \boldsymbol{x}$ is the vector of *coordinates* in the new basis U!

Decorrelation

We can use an old rule to show what happens in the projection,

Transformation of Multivariate Normals from Week 3

If $X \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ then

$$Y = AX + b \Rightarrow Y \sim \mathcal{N}(A\boldsymbol{\mu} + b, A\boldsymbol{\Sigma}A^\top)$$

If $X \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ we can first *center* the data like we did with the Z-score:

$$\hat{X} = X - \boldsymbol{\mu} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$$

And then use the eigendecomposition of $\boldsymbol{\Sigma}$,

$$\boldsymbol{\Sigma} = USU^\top$$

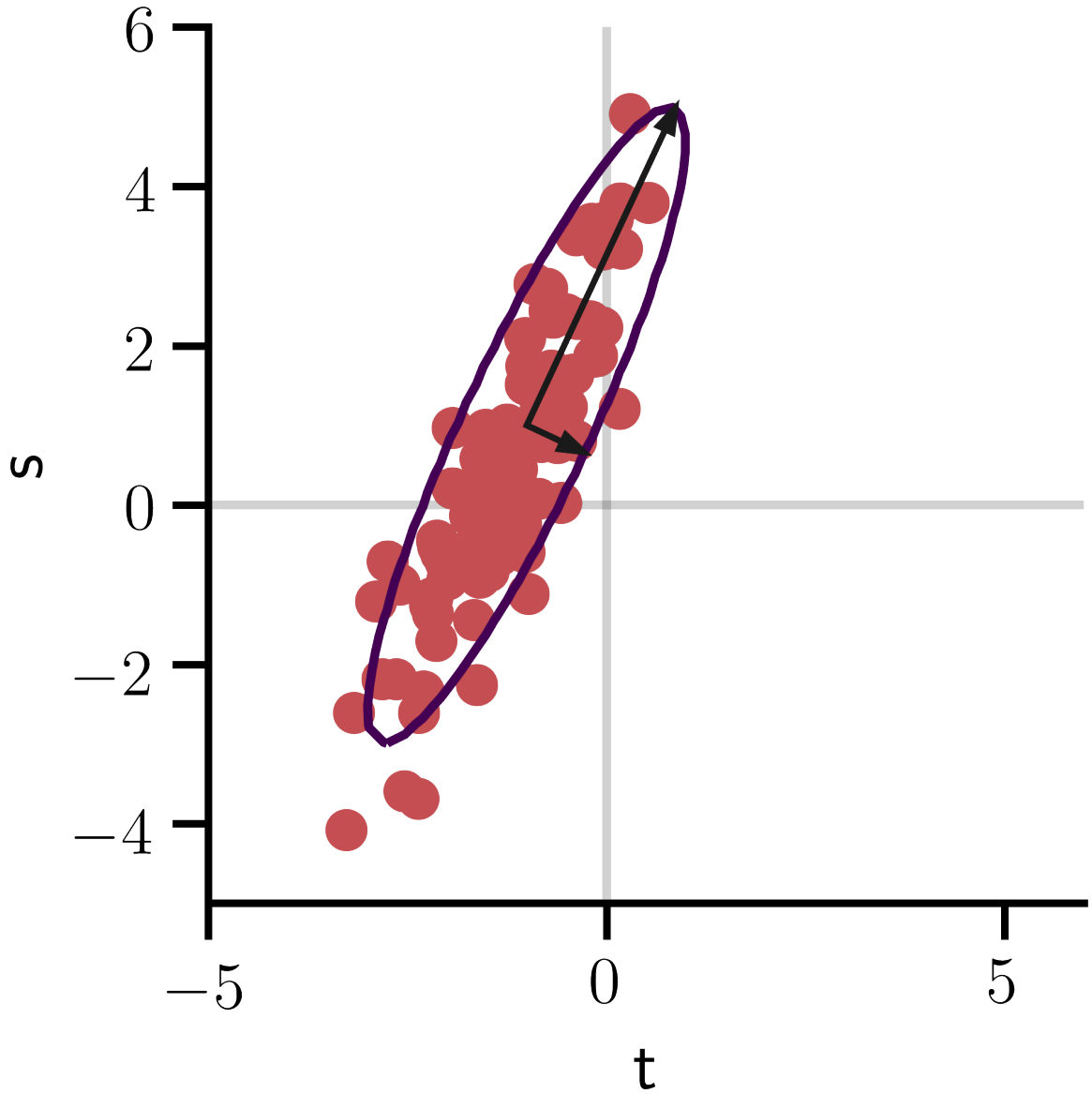
to project $\hat{X}$ onto the eigenbasis as $Z = U^\top \hat{X}$. What does the rule give us? The new covariance turns out to be

$$\underbrace{\boldsymbol{\Sigma}_Z}_{\text{new } \Sigma} = \underbrace{U^\top \boldsymbol{\Sigma} U}_{\text{our rule}} = \underbrace{U^\top U}_{U^\top U = I} SU^\top U = S.$$

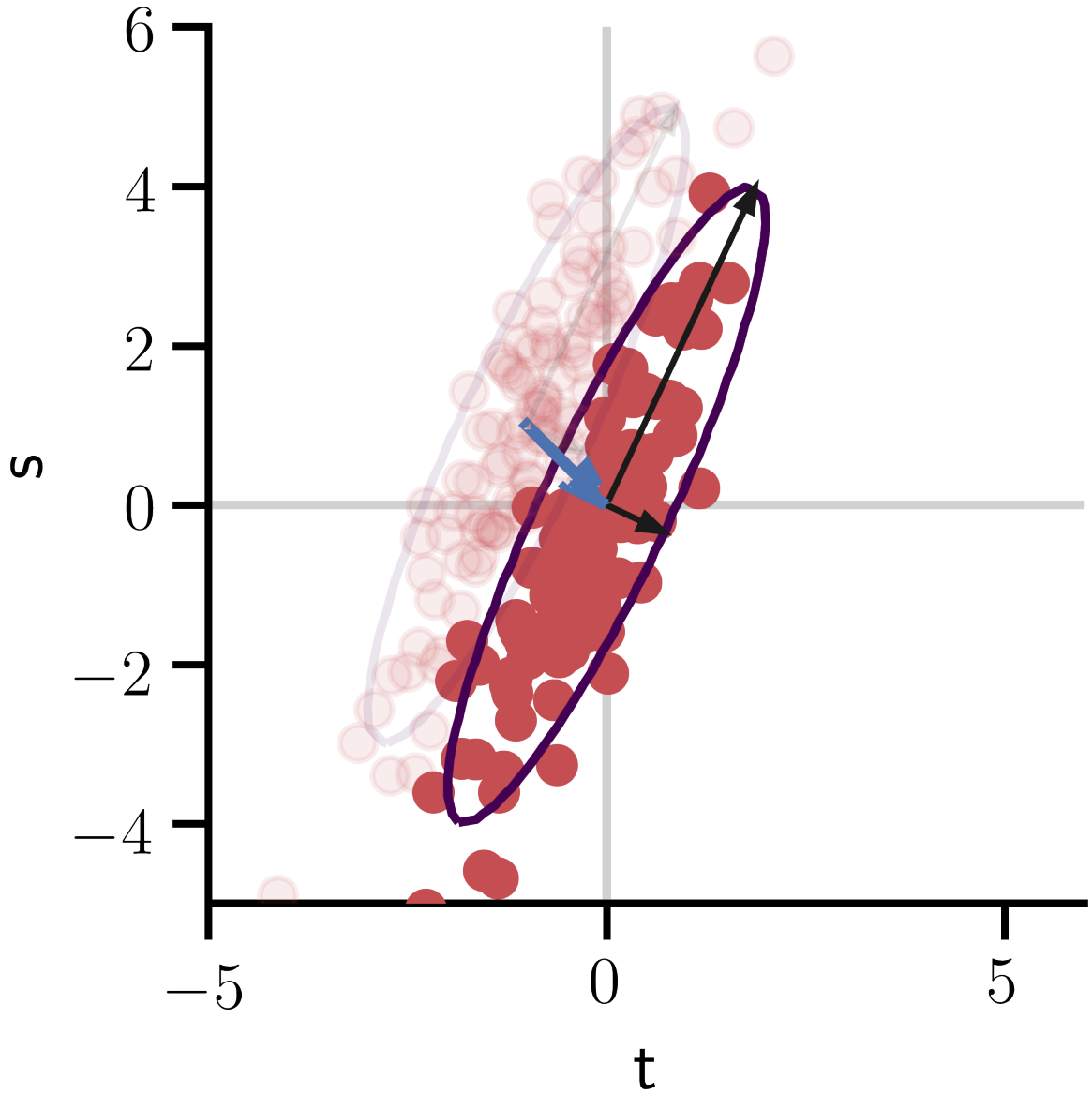
which is *diagonal* implying that the result is *uncorrelated*!

$$z_n = U_{\text{ML}}^\top (x_n - \boldsymbol{\mu}_{\text{ML}}). \qquad \text{(decorrelated representation)}$$

Decorrelation in Practice



Decorrelation in Practice



Decorrelation in Practice