Example: Plug-in Estimators

A simple application of the empirical distribution are plug-in estimators.

Integral statistics

Many of the most common statistics can be written in the form

$$S[p] = \int_{\mathbf{X}} f(\mathbf{x}) p(\mathbf{x}) d\mathbf{x} \; .$$

Examples: Expectation of p (where $f(\mathbf{x}) = \mathbf{x}$), variance of p (where $f(\mathbf{x}) = (\mathbf{x} - \mu)^2$), etc.

Plug-in estimator

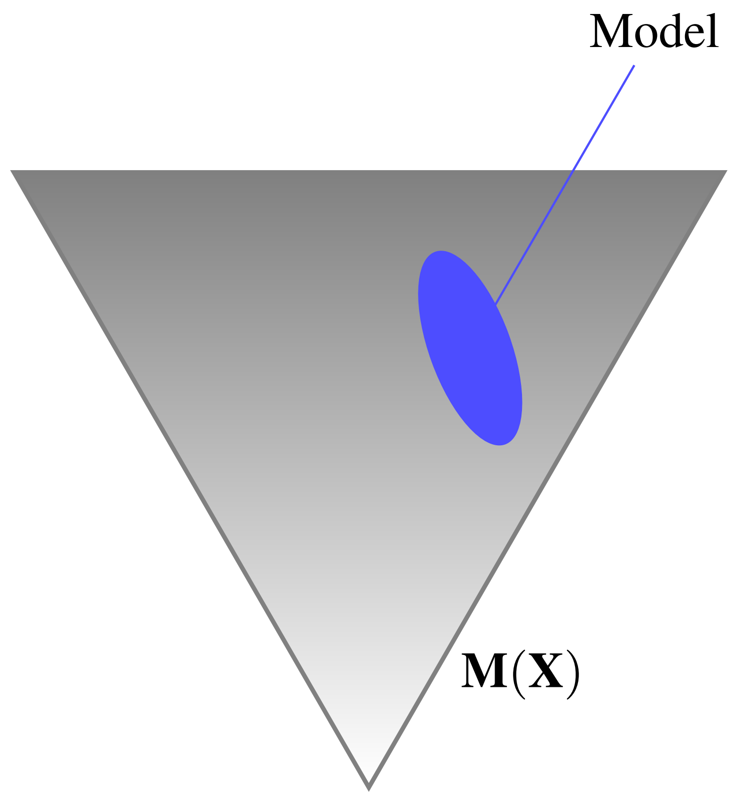
One way to estimate S from a sample $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ is to "plug in" the empirical distribution $\mathbb{F}_n$ for the true distribution p:

$$\hat{S} := \int_{\mathbf{X}} f(\mathbf{x}) \mathbb{F}_n(d\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} f(\mathbf{x}_i)$$

This estimator is called the **plug-in** estimator of S.

Not examinable.

STATISTICAL MODELS



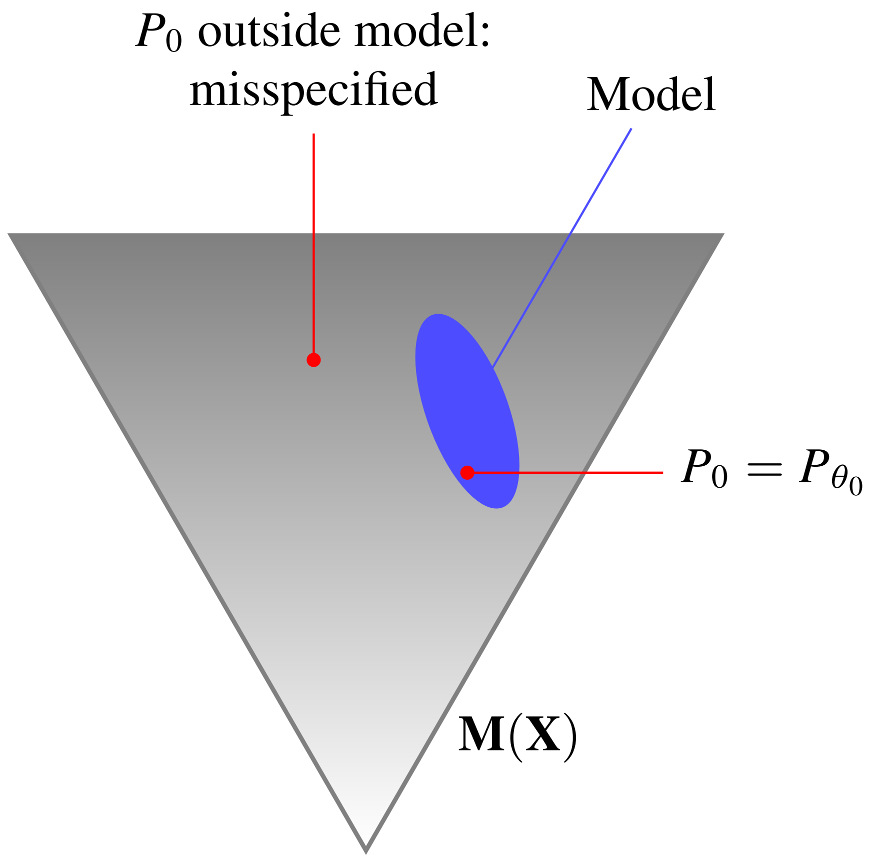
Recall that a statistical model with parameter space $\mathcal{T}$ is a set

$$\mathcal{P} = \{P_\theta | \theta \in \mathcal{T}\}$$

of distributions. In particular, a model is a subset of $\mathbf{M}(\mathbf{X})$.

Not examinable.

Model Misspecification



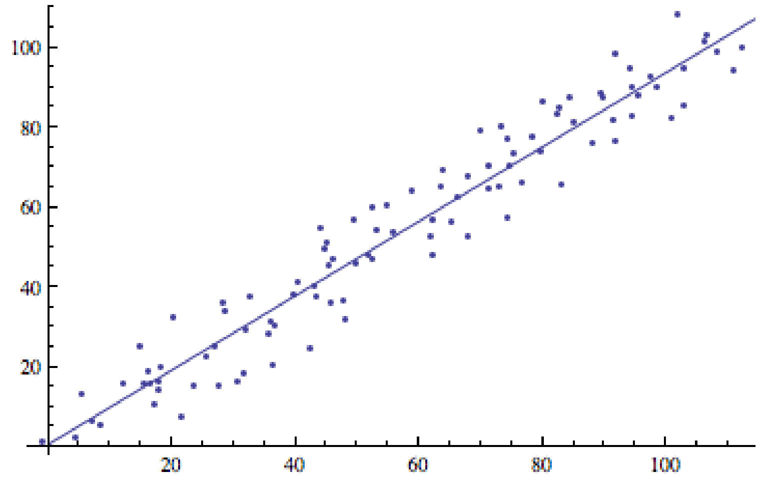
Suppose the observed data is generated by a "true" distribution P_0.

- We say that the model is **correctly specified** if $P_0 \in \mathcal{P}$.
- If $P_0 \notin \mathcal{P}$, the model is **misspecified**.

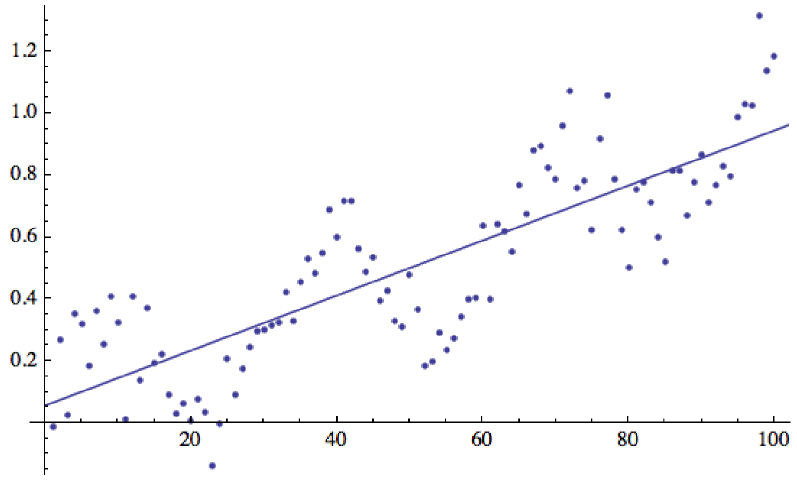
Not examinable.

Model Misspecification

Example: Regression



Correctly specified



Misspecified

Implications

- If the model is correctly specified, we can in principle find a parameter value $\theta \in \mathcal{T}$ which fully explains the data.
- Finding θ still requires a valid estimation procedure.
- In most cases, we can live with misspecification, provided that the approximation error can be controlled.

Not examinable.

Bias-Variance Trade-Off

Model complexity

If our only objective is to avoid misspecification, we should make the model (the subset $\mathcal{P}$ of $\mathbf{M}(\mathbf{X})$) as large as possible. A larger set $\mathcal{P}$ corresponds to a model that is more flexible.

Bias vs. Variance

- Misspecification means that, no matter how much data we observe, our estimated model never completely explains the data. This can be interpreted as a form of bias.
- To avoid misspecification, we can make the model more flexible.
- We have already seen how estimates in more flexible models tend to vary more between sample sets (higher variance).

Thus, we can decrease bias at the expense of increasing variance, and vice versa. This phenomenon is called the **bias-variance trade-off**.

Not examinable.

Measuring Model Complexity

In parametric models

- A fundamental measure of model complexity is the number of **degrees of freedom** (d.o.f.).
- This is roughly the dimension of the parameter space (= the number of independent scalar paramaters), provided the parametrization is chosen such that the entries of the parameter vector are reasonably independent of each other.
- For example, a Gaussian scale model on $\mathbb{R}^d$ (unknown mean, fixed variance) has d degrees of freedom, a Gaussian model with unknown mean and variance has $d + d(d-1)/2$.

Remark: Nonparametric models

- In nonparametric models (= infinite-dimensional parameter space), measuring model complexity is much harder.
- Tools to solve this problem are developed in two closely related research fields called Statistical Learning Theory (in Machine Learning) and Empirical Process Theory (in Statistics).

Not examinable.

Example: Regression with Additive Noise

Model

$$Y = f(X) + \varepsilon \qquad \text{with} \quad \mathbb{E}[\varepsilon] = 0 \quad \text{and} \quad \text{Var}[\varepsilon] = \sigma^2 \, .$$

We assume that $f \in \mathcal{F}$, where $\mathcal{F}$ is some class of functions. Linear regression is the special case where $\mathcal{F}$ is the set of linear functions.

Bias and Variance

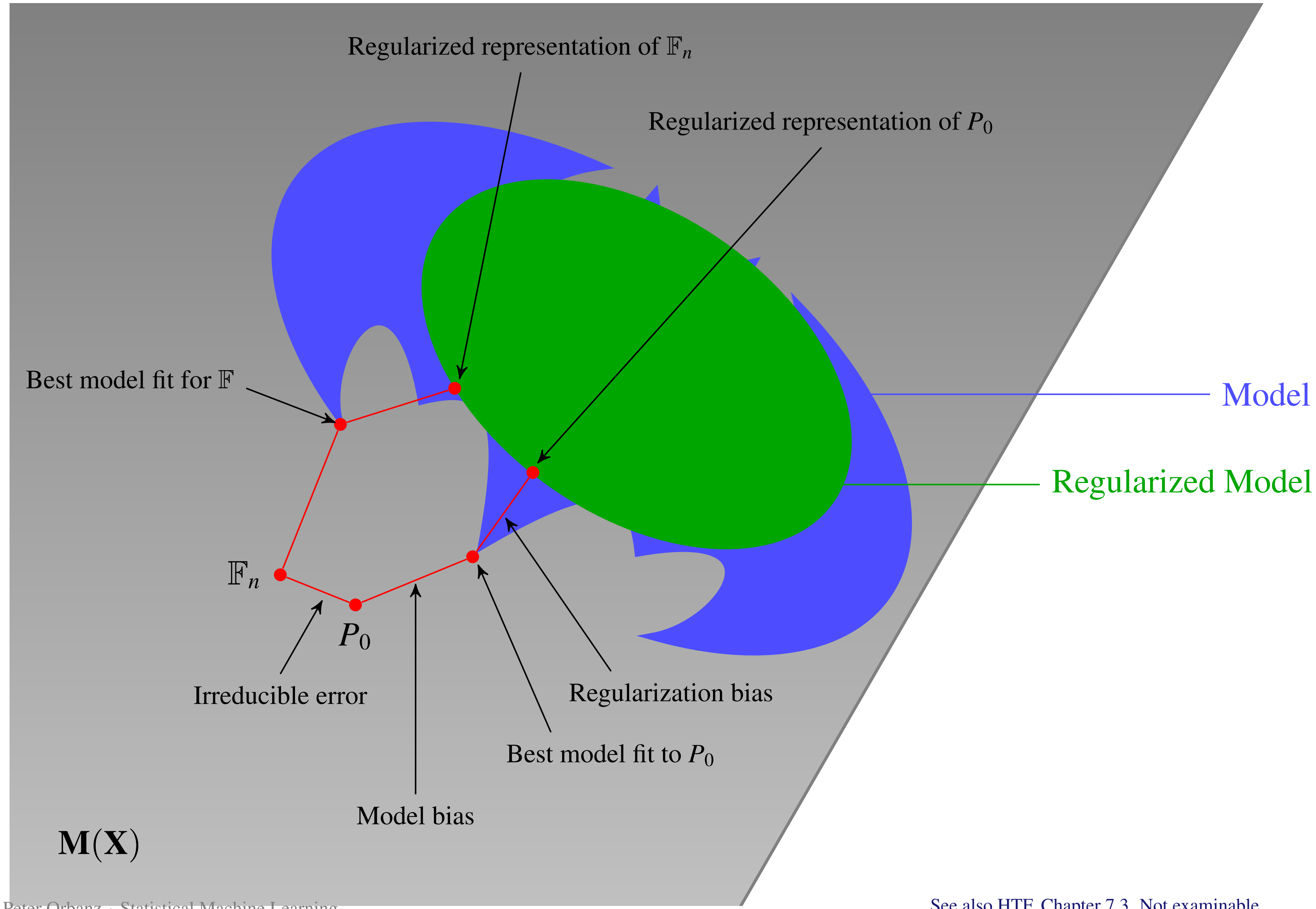
$$\begin{aligned}
\text{Prediction error}(x_0) &= \mathbb{E}[(Y - \hat{f}(x_0))^2 | X = x_0] \\
&= \sigma_\varepsilon^2 + (\mathbb{E}[\hat{f}(x_0)] - f(x_0))^2 + \mathbb{E}[\hat{f}(x_0) - \mathbb{E}[\hat{f}(x_0)]]^2 \\
&= \sigma_\varepsilon^2 + \text{Bias}(\hat{f}(x_0))^2 + \text{Var}[\hat{f}(x_0)] \\
&= \underbrace{\text{Irreducible error}}_{\text{This is due to } \mathbb{F}_n \neq P_0.} + \underbrace{\text{Bias}^2}_{\uparrow} + \underbrace{\text{Variance}}_{\uparrow}
\end{aligned}$$

Increases with model flexibility.

Decreases with model flexibility.

Not examinable.

Types of Errors



See also HTF, Chapter 7.3. Not examinable.

Specifically: Linear regression

Unregularized case

In linear least-squares regression, the varaince term is

$$\mathrm{Var}[\hat{f}(x_0)] = \|(\tilde{\mathbf{X}}^t\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}^t x_0\|\sigma_\varepsilon^2$$

Ridge regression

In linear least-squares regression, the variance term is

$$\mathrm{Var}[\hat{f}(x_0)] = \|(\tilde{\mathbf{X}}^t\tilde{\mathbf{X}} + \lambda\mathbb{I})^{-1}\tilde{\mathbf{X}}^t x_0\|\sigma_\varepsilon^2$$

This term is generally smaller than in the unregularized case, but the corresponding bias term is larger.

Not examinable.

SUMMARY

Model complexity

- Model choice has to trade off stability (low variance) vs flexibility (low bias).
- It can be beneficial (in terms of prediction error) to permit a bias if this decreases the variance.
- Bias and variance terms combine to form prediction error.

How does cross validation fit in?

- Cross validation estimates the prediction error.
- A high variance of estimates will typically be reflected in a high variance between estimates on different folds.

Not examinable.

Unsupervised Learning

Unsupervised Learning

In short

- Label information available $\rightarrow$ supervised learning (classification, regression)
- No label information available $\rightarrow$ **unsupervised learning**

Problem

- Try to find structure or patterns in data without knowing a correct solution.
- By choosing a model, we specify what kind of patterns we are looking for.

Examples of unsupervised learning problems

- Dimension reduction
- Clustering

Dimension Reduction

Dimension Reduction Problems

Setting

- Given: High-dimensional data $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^D$
- Look for: Low-dimensional projection of the data such that important structure in data is preserved

More precisely

- Find suitable linear subspace $V \subset \mathbb{R}^D$ with $\dim(V) =: d$ small.
- Compute projection $\mathbf{x}_j^v$ of each $\mathbf{x}_j$ onto V

Most common cases: $d \in \{2, 3\}$ for visualization.

PCA: Overview

Assumptions

1. Directions along which uncertainty in data is maximal are most interesting.
2. Uncertainty is measured by variance.

Method

- Compute empirical covariance matrix of the data.
- Compute its EValues $\lambda_1, \dots, \lambda_D$ and EVectors $\xi_1, \dots, \xi_D$.
- Choose the d largest EValues, say, $\lambda_{j_1}, \dots, \lambda_{j_d}$.
- Define subspace as $V := \text{span}\{\xi_{j_1}, \dots, \xi_{j_d}\}$
- Project data onto V: For each $\mathbf{x}_i$, compute $\mathbf{x}_i^v := \sum_{j=1}^{d} \langle \mathbf{x}_i, \xi_j \rangle \xi_j$

This algorithm is called **Principal Component Analysis (PCA)**.

PCA

Notation

- Empirical mean of data: $\hat{\mu}_n := \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i$
- Empirical variance of data (1 dimension): $\hat{\sigma}_n^2 := \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \hat{\mu}_n)^2$
- Empirical covariance of data (D dimensions): $\hat{\Sigma}_n := \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \hat{\mu}_n)(\mathbf{x}_i - \hat{\mu}_n)^t$

Recall outer product of vectors: Matrix $(\mathbf{x}\mathbf{x}^t)_{ij} := x_i x_j$

PCA Idea

Project data onto a direction $v \in \mathbb{R}^D$ such that the variance of the projected data is maximized.

PCA

Claim

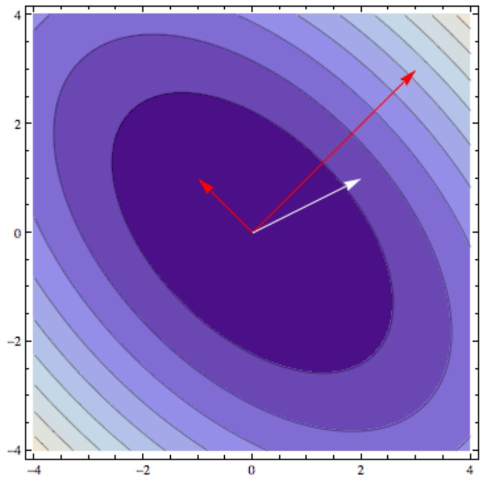
The variance of the projected data is given by $\langle \mathbf{v}, \hat{\Sigma}_n \mathbf{v} \rangle$.

Explanation

The projection of $\mathbf{x}_i$ onto $\mathbf{v}$ is $\langle \mathbf{x}_i, \mathbf{v} \rangle$. Substitute into empirical variance:

$$\frac{1}{n}\sum_{i=1}^{n}(\langle \mathbf{x}_i, \mathbf{v}\rangle - \langle \hat{\mu}_n, \mathbf{v}\rangle)^2 = \frac{1}{n}\sum_{i=1}^{n}\langle (\mathbf{x}_i - \hat{\mu}_n), \mathbf{v}\rangle^2$$

$$= \Big\langle \underbrace{\Big(\frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \hat{\mu}_n)(\mathbf{x}_i - \hat{\mu}_n)^t\Big)}_{=\hat{\Sigma}_n}\mathbf{v}, \mathbf{v} \Big\rangle$$



Red: Eigenvectors. White: $\mathbf{v}$.

Recall: quadratic forms

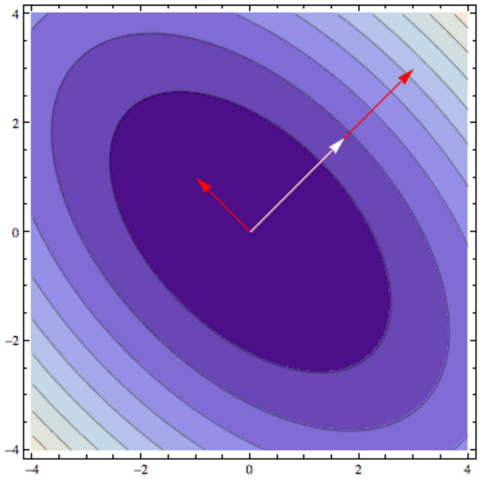
The variance along $\mathbf{v}$ is the value of the quadratic form defined by $\hat{\Sigma}_n$, evaluated at $\mathbf{v}$.

PCA

PCA as optimization problem

$$\begin{aligned} \max_{\mathbf{v}} \quad & \langle \mathbf{v}, \hat{\Sigma}_n \mathbf{v} \rangle \\ \text{s.t.} \quad & \|\mathbf{v}\|_2 = 1 \end{aligned}$$

The constraint $\|\mathbf{v}\|_2 = 1$ ensures that we maximize by adjusting the direction of $\mathbf{v}$; otherwise, we could make $\langle \mathbf{v}, \hat{\Sigma}_n \mathbf{v} \rangle$ arbitrarily large by scaling $\mathbf{v}$.



Optimization problem: Solution

We know from our discussion of quadratic forms:

$\langle \mathbf{v}, \hat{\Sigma}_n \mathbf{v} \rangle$ maximal $\Leftrightarrow$ $\mathbf{v}$ points in direction of $\xi_{\max}$

where $\xi_{\max}$ is the EVector associated with the largest EValue.

Projecting onto several dimensions

Projecting onto 2 dimensions

1. Project onto 1 dimension.
2. Remove that dimension from data, i.e. restrict data to the space orthogonal to $\mathbf{v}$.
3. Apply PCA on restricted space.

It is not hard to show that the result is the direction of the EVector associated with the second-largest eigenvalue.

Projecting onto d dimensions

By iterating the procedure above, we find that the optimal projection onto d dimensions corresponds to the d largest EValues.

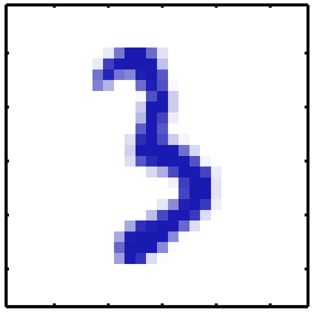
Summary

The PCA algorithm (=project on the d "largest" EVectors) can be justified as the projection which maximizes the variance of the projected data.

PCA: Example

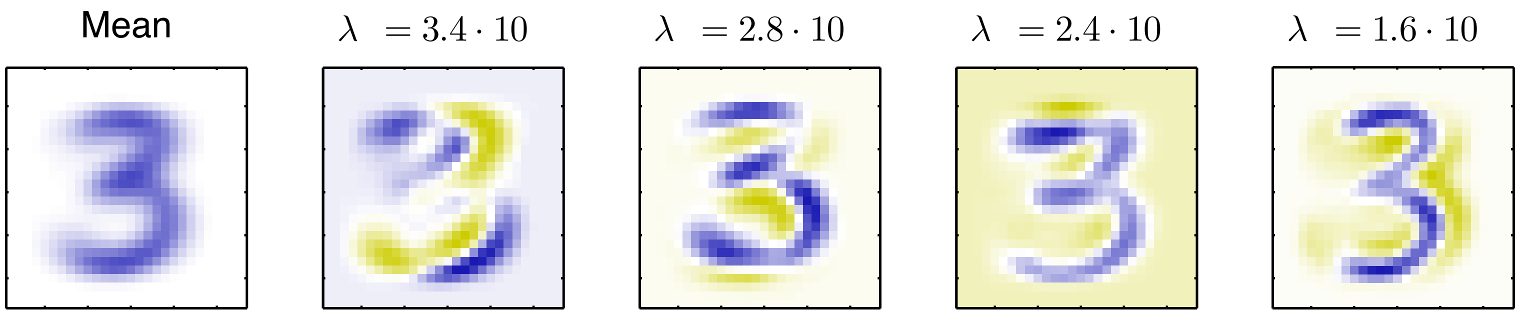
Again: Digit data

- Recall: $\mathbf{x}_i \in \mathbb{R}^{256}$
- Here: Images representing the number 3.



Eigenvectors

The mean $\hat{\mu}_n$ and the EVectors are also elements of $\mathbb{R}^{256}$, so we can plot them as images as well.


Mean
$\lambda \ = 3.4 \cdot 10$
$\lambda \ = 2.8 \cdot 10$
$\lambda \ = 2.4 \cdot 10$
$\lambda \ = 1.6 \cdot 10$

These are the EVectors for the four largest EValues.

Principal components

The first few eigenvectors are called **principal components**. They can be regarded as a summary of the main features of the data.

COMPRESSION

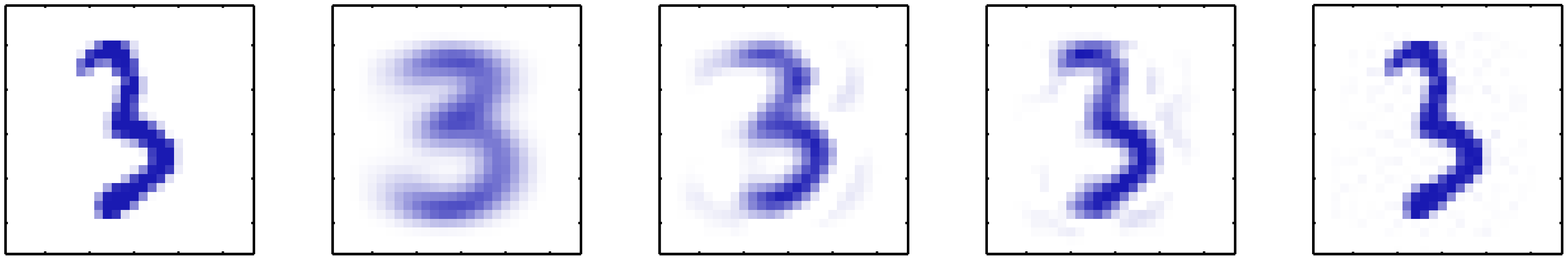
Using PCA as a compressor

- To store a digit, we have to store 256 floating point (FP) numbers.
- If we store its projection onto d eigenvectors, we have to store:
  1. The d complete eigenvectors = $d \cdot 256$ FP numbers.
  2. d FP numbers per image.
- For n large enough, i.e. if $n \cdot d + d \cdot 256 < n \cdot 256$, this results in compression of the data.

Lossy data compression

- From the compressed data, we cannot restore the data completely. Such compression methods are called **lossy compression**.
- Other examples: JPEG, MP3, etc.
- Compression methods which completely preserve the data are called **lossless**. (Example: ZIP compression for digital files.)

Compressing Digits with PCA



- The input image $\mathbf{x}$ is projected onto each eigenvector ξ_i to obtain a coefficient c_i.
- Then $\mathbf{x}$ can be represented as

$$\mathbf{x} = \sum_{j=1}^{D} c_i \xi_i$$

- A compressed version using d components is obtained as

$$\mathbf{x}^{(d)} = \sum_{j=1}^{d} c_i \xi_i$$

Since $\mathbf{x}^{(d)} \in \mathbb{R}^{256}$, we can plot it as an image. These are the images above.

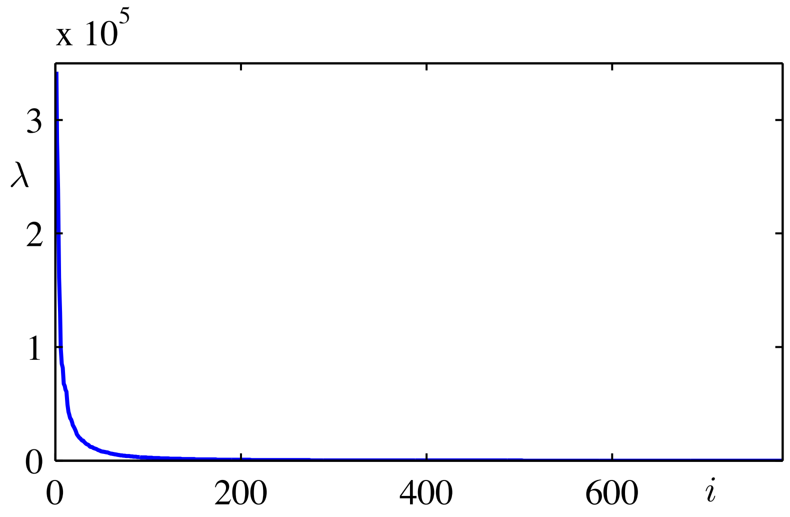
MODEL SELECTION

How many eigenvectors should we use?

- For visualization: Usually 2 or 3.
- For approximation or compression: We would like to minimize the approximation error, so we should try to keep all large EValues.

Eigenvalues in the digit problem

- Ideally, the curve of the size-ordered EValues shows a clear jump or bent at which we can truncate.
- Such a jump is called a **spectral gap**.


x 10^5
λ
i