High-Dimensional Linear Regression and the Bet on Sparsity

Pierre-Alexandre Mattei

http://pamattei.github.io – @pamattei
pierre-alexandre.mattei@inria.fr

MSc Data Science

Betting on sparsity for high-dimensional data

- Ubiquity and curses of high-dimensional data
- Betting on sparsity through Bayesian model uncertainty

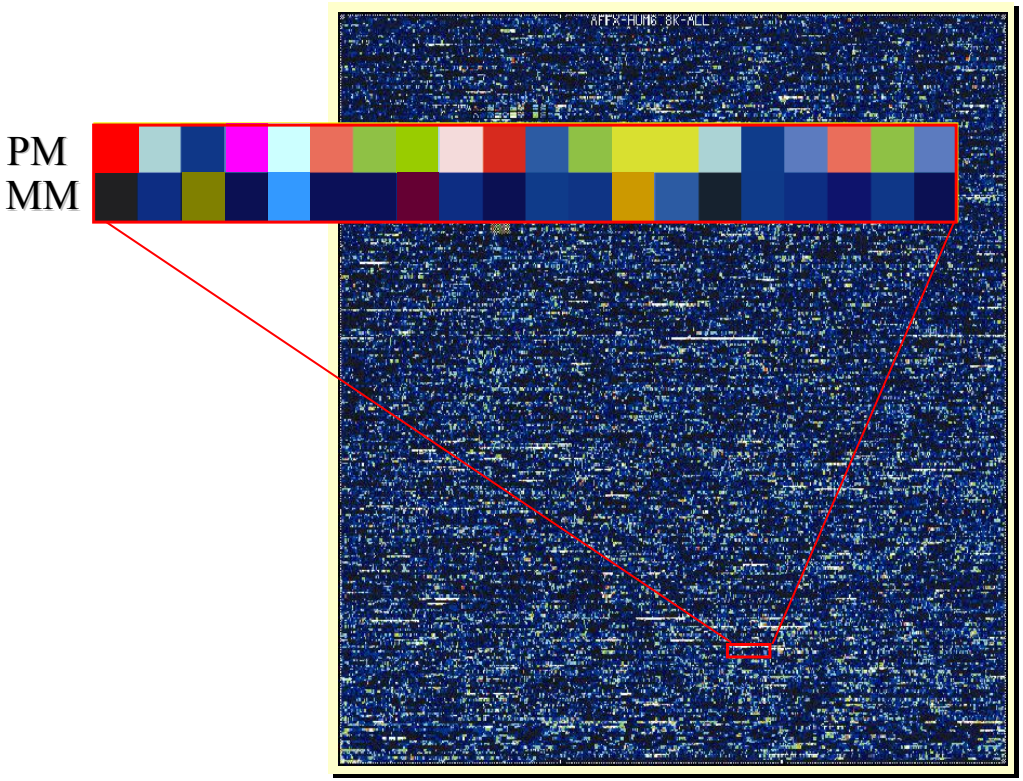
High-dimensional linear regression

High-dimensional data are ubiquitous

Traditional statistical paradigm:

- large number n of observations (patients, voters...),
- small number p of variables (medical measurements, answers in a survey...).

Modern (big) data: DNA microarray $n \approx 100, p \approx 10\,000$.

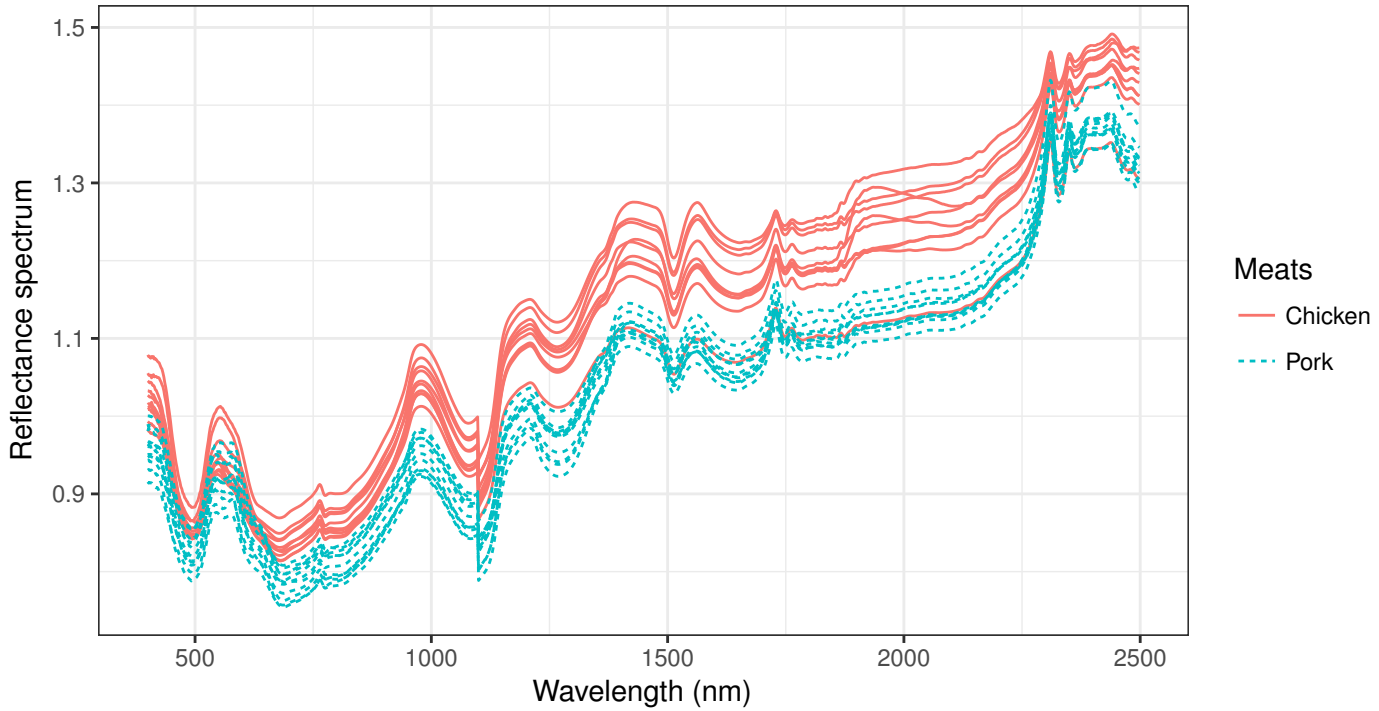


High-dimensional data are ubiquitous

Traditional statistical paradigm:

- large number n of observations (patients, voters...),
- small number p of variables (medical measurements, answers in a survey...).

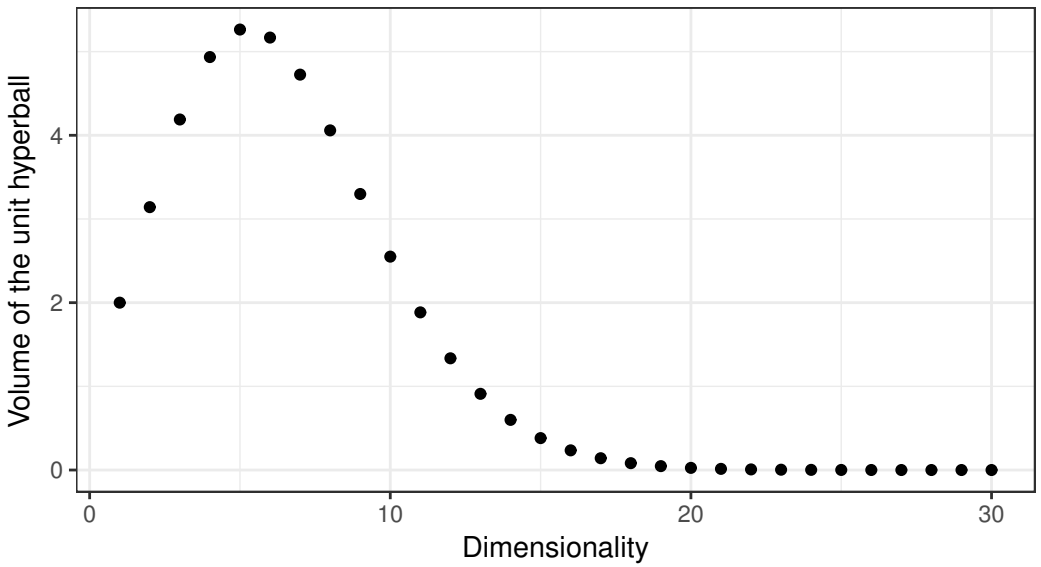
Modern (big) data: NMR spectra $n \approx 100, p \approx 1\,000$.



High-dimensional data are cursed

High-dimensional datasets are collections of points in high-dimensional spaces...
...and the geometry of high-dimensional spaces is rather peculiar.

High-dimensional Euclidean (hyper)balls are essentially empty!



High-dimensional data are cursed

Since Gauss's and Legendre's least squares ($\approx$1810), most of classical statistics rely on Euclidean distances, which do not behave nicely in high-dimensions.

"All this [the problems related to high-dimensional geometry] *may be subsumed under the heading "the curse of dimensionality". Since this is a curse (...) there is no need to feel discouraged about the possibility of obtaining significant results despite it."*

Richard Bellman (1957)

Betting on sparsity

Parametric statistical models assume that the observed data $\mathbf{X} \in \mathbb{R}^{n \times p}$ comes from a density in a parametrized family $(p(\cdot|\boldsymbol{\theta}))_{\boldsymbol{\theta} \in \Theta}$.

The dimension of Θ usually grows with the dimensionality p of the data, which is another challenge of high-dimensional inference!

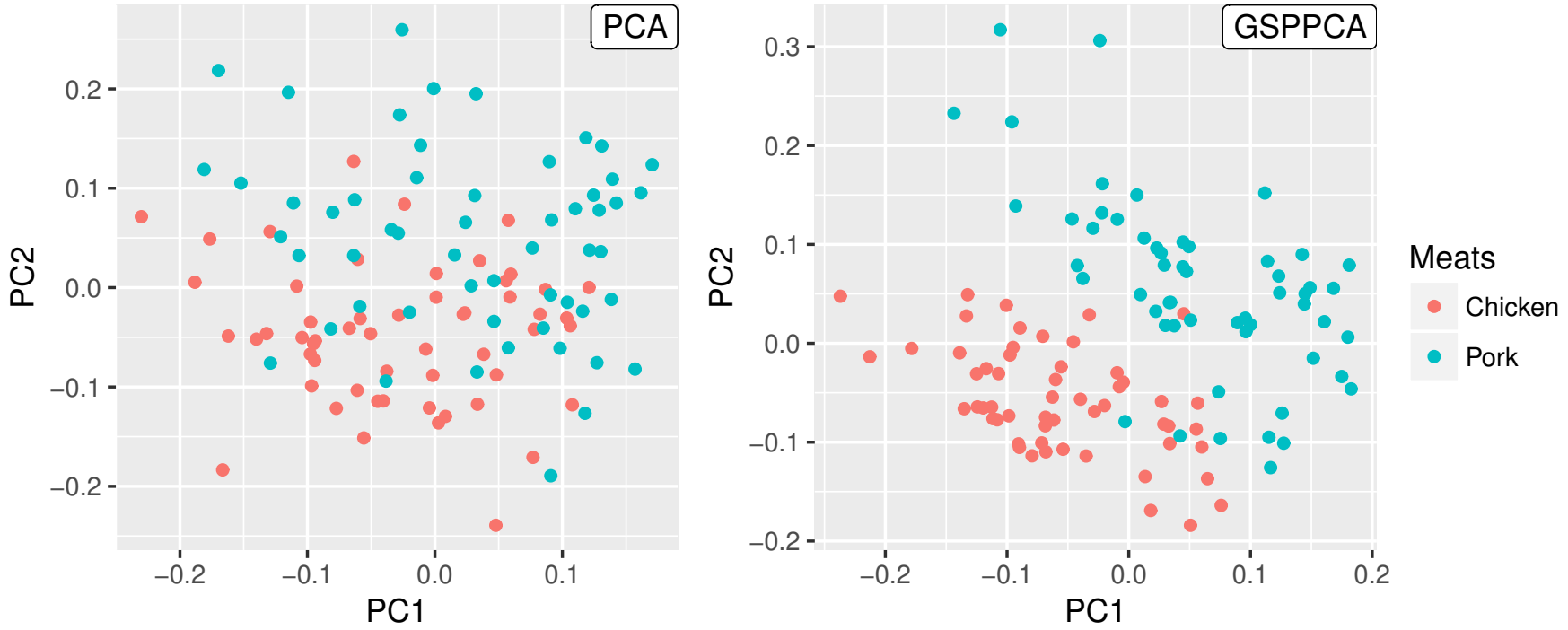
But most statistical/geometrical problems tend to disappear if we assume that $\boldsymbol{\theta}$ has few nonzero coefficients. We say that $\boldsymbol{\theta}$ is q-sparse, with $q \ll p$.

"This has been termed the "Bet on sparsity" principle: Use a procedure that does well in sparse problems, since no procedure does well in dense problems."

Hastie, Tibshirani & Wainwright
Statistical Learning with Sparsity: the Lasso and Generalizations
(CRC Press 2015)

Visualizing data via PCA and the bet on sparsity

PCA aims at summarizing high-dimensional data using only two transformed variables. Without betting on sparsity, the results are much less interpretable.



Bouveyron, Latouche, and Mattei (EJS 2018)

Betting on sparsity via likelihood penalization

A natural way to find a sparse parameter is to maximize a penalized version of the likelihood

$$\hat{\boldsymbol{\theta}} \in \text{argmax}_{\boldsymbol{\theta} \in \Theta} \log p(\mathbf{X}|\boldsymbol{\theta}) - \lambda ||\boldsymbol{\theta}||_0.$$

This combinatorial problem lacks scalability, and is often replaced by

$$\hat{\boldsymbol{\theta}} \in \text{argmax}_{\boldsymbol{\theta} \in \Theta} \log p(\mathbf{X}|\boldsymbol{\theta}) - \lambda ||\boldsymbol{\theta}||_1.$$

Such optimization problems – often called lasso problems, following Tibshirani ('96) – are highly scalable but hard to calibrate.

A good recent reference for these approaches is the freely available online book by Hastie, Tibshirani & Wainwright, *Statistical Learning with Sparsity: the Lasso and Generalizations* (CRC Press 2015).

Betting on sparsity as Bayesian model uncertainty

- The Bayesian bet on sparsity is a particular instance of Bayesian model uncertainty. All possible sparsity patterns can be viewed as competing statistical models, over which we spread prior beliefs.
- The idea of spreading prior belief between models and computing consequently their posterior probabilities was independently developed by Harold Jeffreys & Dorothy Wrinch ($\approx$ 1920), and by Jack Good & Alan Turing ($\approx$ 1942).
- It embodies the fact that simpler models are a priori pretty likely to be useful, a philosophical principle often referred to as Occam's razor.

A good recent reference for these approaches is the book by Murphy, *Machine Learning: A Probabilistic Perspective* (MIT Press 2012). I've also written a review paper on the subject: *A Parsimonious Tour of Bayesian Model Uncertainty* (arXiv:1902.05539).

High-dimensional linear regression

Good old linear regression

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \mu\mathbf{1}_n + \boldsymbol{\varepsilon}$$

$\mathbf{Y} \in \mathbb{R}^n$ is a vector of n observed responses,
$\mathbf{X} \in \mathcal{M}_{n,p}$ is the design matrix with p input variables,
$\boldsymbol{\varepsilon}$ is a stochastic noise term with zero mean and finite variance,
$\mu \in \mathbb{R}$ is the intercept,
Goal: Predicting the value of the respondse given some new data.

Centering the data

We typically assume that the data have been centered, i.e. $\mathbf{Y}$ has zero mean. A nice thing about this is that it allows us to forget about the intercept (which can just be estimated by $\mu = 0$).
Of course, we can still "uncenter" our predictions at the end if we want (as long as we kept the original means).
Exercise: How do we "uncenter"?

Good old linear regression

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \mu\mathbf{1}_n + \boldsymbol{\varepsilon}$$

$\mathbf{Y} \in \mathbb{R}^n$ is a vector of n observed responses,
$\mathbf{X} \in \mathcal{M}_{n,p}$ is the design matrix with p input variables,
$\boldsymbol{\varepsilon}$ is a stochastic noise term with zero mean and finite variance,
$\mu \in \mathbb{R}$ is the intercept,
Goal: Predicting the value of the respondse given some new data.

Centering the data

We typically assume that the data have been centered, i.e. $\mathbf{Y}$ has zero mean. A nice thing about this is that it allows us to forget about the intercept (which can just be estimated by $\mu = 0$).
Of course, we can still "uncenter" our predictions at the end if we want (as long as we kept the original means).
Exercise: How do we "uncenter"?
We can just estimate μ by the empirical mean of $\mathbf{Y}$ and add that to our predictions.

Good old (centered) linear regression

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

If we have a new data point $\mathbf{x}_{\text{new}}$, how do we predict its response?

Good old (centered) linear regression

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

If we have a new data point $\mathbf{x}_{\text{new}}$, how do we predict its response?
We can just use

$$\hat{\mathbf{y}}_{\text{new}} = \mathbb{E}[y|\mathbf{x}_{\text{new}}] = \mathbf{x}_{\text{new}}^T\boldsymbol{\beta}.$$

Good old (centered) linear regression

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

If we have a new data point $\mathbf{x}_{\text{new}}$, how do we predict its response?
We can just use

$$\hat{\mathbf{y}}_{\text{new}} = \mathbb{E}[y|\mathbf{x}_{\text{new}}] = \mathbf{x}_{\text{new}}^T\boldsymbol{\beta}.$$

Note that, for this to be true, we don't need the noise to be Gaussian (we just needed it to have zero mean).

Good old (centered) linear regression

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

If we have a new data point $\mathbf{x}_{\text{new}}$, how do we predict its response?
We can just use

$$\hat{\mathbf{y}}_{\text{new}} = \mathbb{E}[y|\mathbf{x}_{\text{new}}] = \mathbf{x}_{\text{new}}^T\boldsymbol{\beta}.$$

Note that, for this to be true, we don't need the noise to be Gaussian (we just needed it to have zero mean).
Important consequence: If we mainly want to do prediction, we're mostly interested in estimating β (and the noise term is not too important to estimate).

OLS for linear regression

Since Legendre anbd Gauss ($\approx$ 1805), the traditional approach to linear regression is through ordinary least squares

$$\hat{\boldsymbol{\beta}}_{\text{OLS}} \in \text{argmin}_{\boldsymbol{\beta} \in \mathbb{R}^p} ||\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}||_2^2,$$

which may be interpreted as doing maximum likelihood (exercise: under what assumptions?).

OLS for linear regression

Since Legendre anbd Gauss ($\approx$ 1805), the traditional approach to linear regression is through ordinary least squares

$$\hat{\boldsymbol{\beta}}_{\text{OLS}} \in \text{argmin}_{\boldsymbol{\beta} \in \mathbb{R}^p} ||\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}||_2^2,$$

which may be interpreted as doing maximum likelihood (exercise: under what assumptions?). If the matrix $\mathbf{X}^T\mathbf{X}$ is invertible, then the problem admits a unique solution:

$$\hat{\boldsymbol{\beta}}_{\text{OLS}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}.$$

OLS for linear regression

Since Legendre anbd Gauss ($\approx$ 1805), the traditional approach to linear regression is through ordinary least squares

$$\hat{\beta}_{\text{OLS}} \in \text{argmin}_{\beta \in \mathbb{R}^p} ||\mathbf{Y} - \mathbf{X}\beta||_2^2,$$

which may be interpreted as doing maximum likelihood (exercise: under what assumptions?). If the matrix $\mathbf{X}^T\mathbf{X}$ is invertible, then the problem admits a unique solution:

$$\hat{\beta}_{\text{OLS}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}.$$

In this course, we'll be concerned in cases where the number of features p is very large. Exercise: What might break in $\hat{\beta}_{\text{OLS}}$ when p is big? In particular bigger than n?

OLS fails in high dimensions

We have $\text{rank}(\mathbf{X}^T\mathbf{X}) \leq n$, so if $p > n$, the matrix $\mathbf{X}^T\mathbf{X}$ is not invertible!
There is an infinite number of minimisers of the squared error...

OLS fails in high dimensions

We have $\text{rank}(\mathbf{X}^T\mathbf{X}) \leq n$, so if $p > n$, the matrix $\mathbf{X}^T\mathbf{X}$ is not invertible!
There is an infinite number of minimisers of the squared error...
Exercise: How could we fix that?

- choose one among all the minimisers (which one?)
- regularise the problem (ridge regression, lasso...)

Choosing one minimiser

A straightforward way of fixing

$$\hat{\beta}_{\text{OLS}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y},$$

when $\mathbf{X}^T\mathbf{X}$ is not invertible is to remplace the inverse by the Moore-Penrose pseudo-inverse $(\mathbf{X}^T\mathbf{X})^\dagger$ (a generalisation of the inverse that exist even when the matrix is not invertible). This is equivalent to finding the minimiser of the squared loss that has the smallest 2-norm.
Exercise: What do you think are advantages, drawbacks?

Choosing one minimiser

A straightforward way of fixing

$$\hat{\beta}_{\text{OLS}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y},$$

when $\mathbf{X}^T\mathbf{X}$ is not invertible is to remplace the inverse by the Moore-Penrose pseudo-inverse $(\mathbf{X}^T\mathbf{X})^\dagger$ (a generalisation of the inverse that exist even when the matrix is not invertible). This is equivalent to finding the minimiser of the squared loss that has the smallest 2-norm.
Exercise: What do you think are advantages, drawbacks?

- easy to compute, no hyperparameter
- will overfit a lot

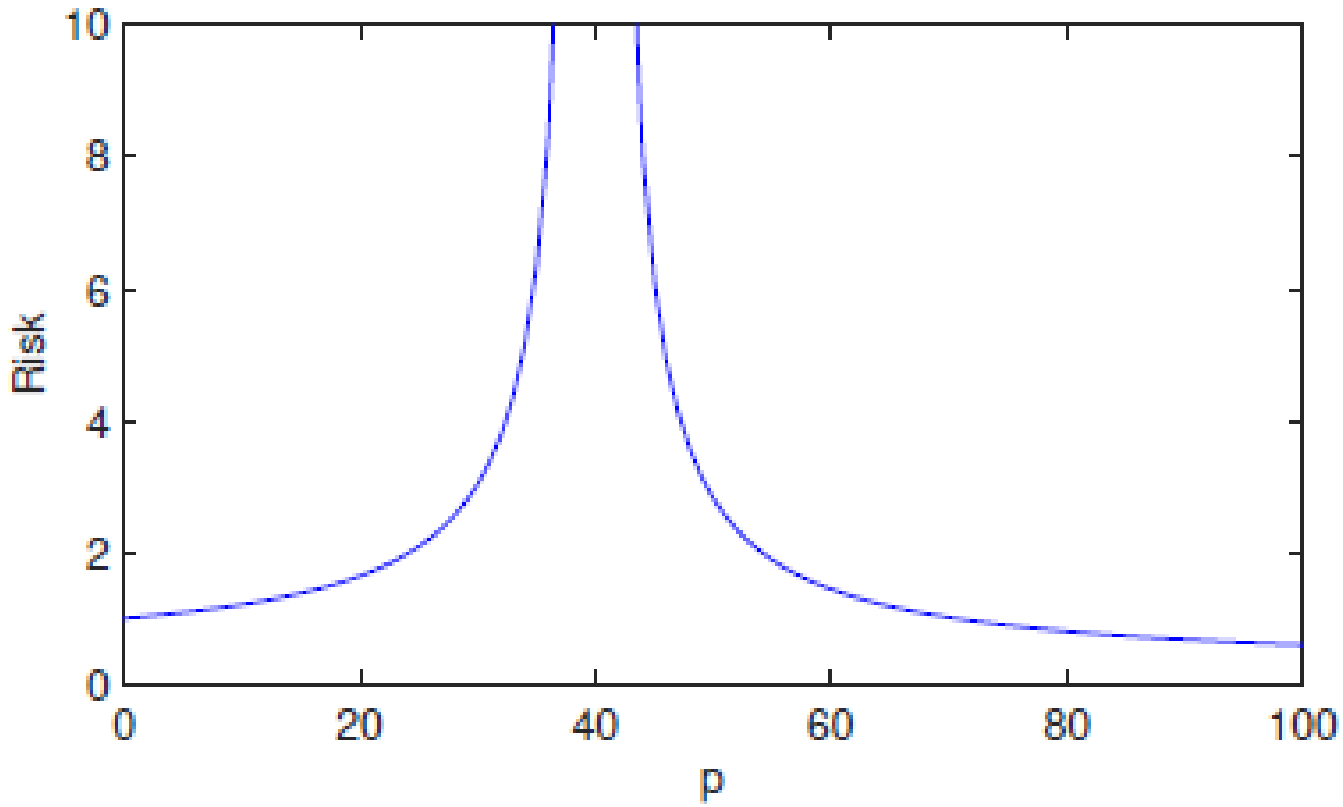
Is crazy overfitting really that bad in that case?

Very recently, people have started to notice that, sometimes, when you really overfeat a lot, you can mysteriously generalise better. This is called the double descent phenomenon (cf talk by Frank Nielsen at the SophIA summit). The concept was introduced by Belkin, Hsu, Ma, and Mandal in the paper *Reconciling modern machine learning practice and the bias-variance trade-off* (PNAS, 2019).

Is crazy overfitting really that bad in that case?

Very recently, people have started to notice that, sometimes, when you really overfeat a lot, you can mysteriously generalise better. This is called the double descent phenomenon (cf talk by Frank Nielsen at the SophIA summit). The concept was introduced by Belkin, Hsu, Ma, and Mandal in the paper *Reconciling modern machine learning practice and the bias-variance trade-off* (PNAS, 2019). Under some technical assumptions, there is a double descent phenomenon for the Moore-Penrose estimate!

Double descent of the Moore-Penrose error



Test error of the Moore-Penrose estimate as a function of the number of features, taken from Belkin, Hsu, and Xu (2019, arXiv:1903.07571). Here $n = 40$.

Ridge regression

A straightforward way of making $\mathbf{X}^T\mathbf{X}$ "more invertible" is by adding it a small diagonal matrix $\lambda\mathbf{I}_p$. The matrix $\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_p$ will then be invertible as long as $\lambda > 0$.
This leads to the ridge regression estimate

$$\hat{\boldsymbol{\beta}}_{\text{Ridge},\boldsymbol{\lambda}} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_p)^{-1}\mathbf{X}^T\mathbf{Y}$$

,

This is equivalent to doing ℓ_2 regularisation:

$$\hat{\boldsymbol{\beta}}_{\text{Ridge},\boldsymbol{\lambda}} = \text{argmin}_{\boldsymbol{\beta}\in\mathbb{R}^p}||\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}||_2^2 + \lambda||\boldsymbol{\beta}||_2^2,$$

for which there is a unique minimiser when $\lambda > 0$.
Exercise: What do you think are advantages, drawbacks?

Ridge regression

Advantages

- unique solution
- regularisation, so less overfitting
- easy to compute
- it can be shown that there exists some λ^* for which ridge beats OLS, even in low dimensions

Drawbacks

- we need to choose λ
- non-sparse, non-interpretable