Machine Learning
CS342

Lecture 7: Sparsity and Regularisation in Linear regression: The Lasso and Ridge Regression

Dr. Theo Damoulas
T.Damoulas@warwick.ac.uk

Office hours (CS 307)
Mon 16:00-17:00 Fri 16:00-17:00

Model Sparsity and Interpretation

Sparsity in ML/Stats: *When few attributes (or few observations) contribute significantly to the resulting model. Many of them are "switched off"*

Why?

- Statistical Interpretation:
  - These attributes are more predictive then others (DTs?)
  - Discover significant correlations (e.g. income and education level)
  - Easier to "read". DTs very successful as easy to read.

- Computational reasons:
  - Resulting model is sparse so a reduced representation, less computation and less storage
  - Certain sparse models will become sparse even during training. So both training and prediction time improved (Big Data setting)
  - *Sparsity as a way to fight overfitting! Preference for simpler models*

Model Sparsity and Interpretation

Lets go back to linear regression with multiple attributes (multivariate)

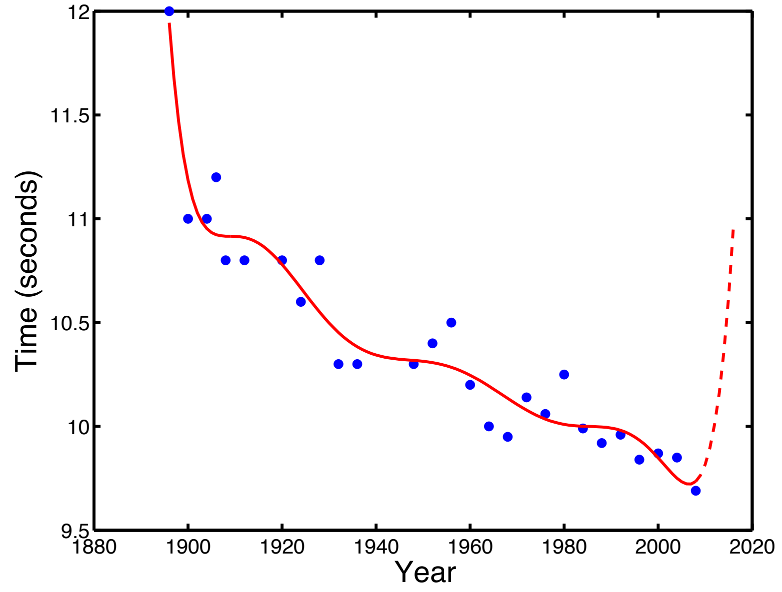
e.g. 2 attributes (2-D input space):

$$\hat{t}_n = \hat{w}_0 + \hat{w}_1 x_{n1} + \hat{w}_2 x_{n2} = \mathbf{x}_n \hat{\mathbf{w}}$$

e.g. 1 attribute but 4th order polynomial expansion (4-D input space):

$$\hat{t}_n = \hat{w}_0 + \hat{w}_1 x_{n1} + \hat{w}_2 x_{n1}^2 + \hat{w}_3 x_{n1}^3 + \hat{w}_4 x_{n1}^4 = \mathbf{x}_n \hat{\mathbf{w}}$$

$$\mathbf{X} = \begin{bmatrix} x_1^0 & x_1^1 & x_1^2 & \cdots & x_1^K \\ x_2^0 & x_2^1 & x_2^2 & \cdots & x_2^K \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ x_N^0 & x_N^1 & x_N^2 & \cdots & x_N^K \end{bmatrix}$$



Model Sparsity and Interpretation

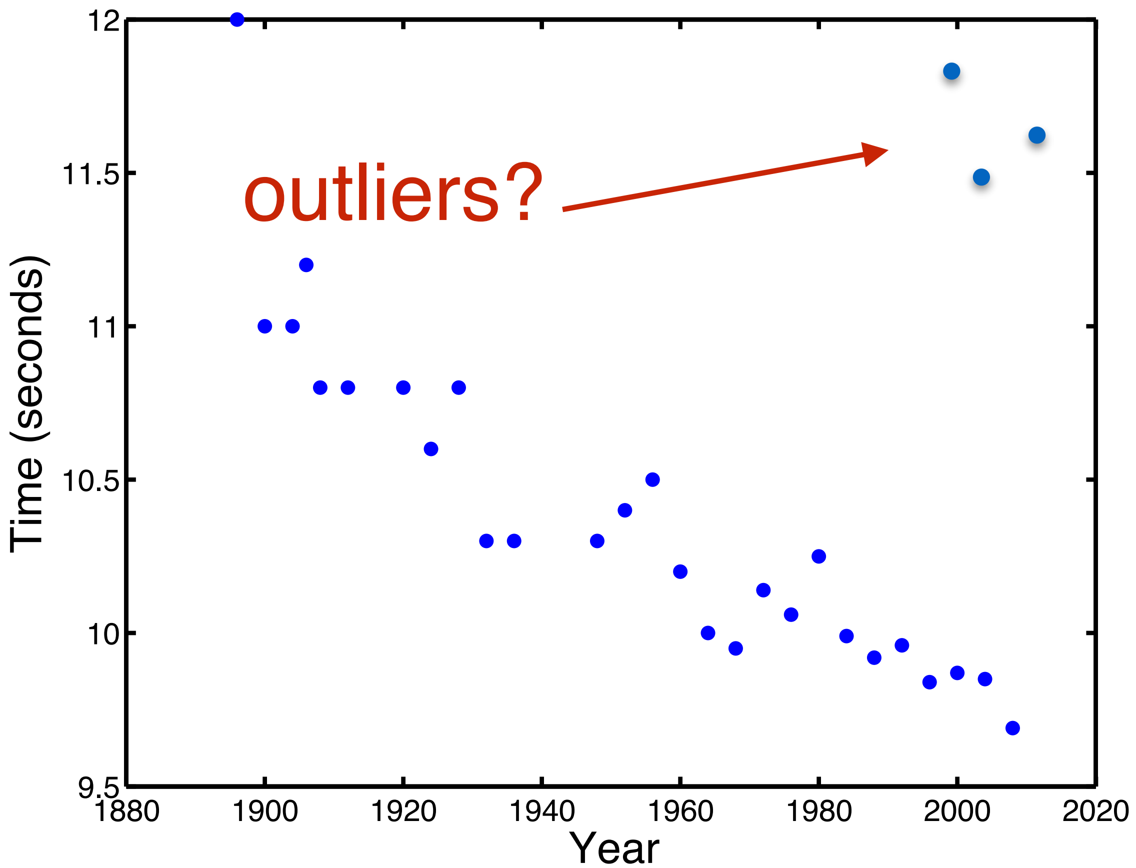
OLS solution

$$\widehat{\mathbf{w}} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{t}$$

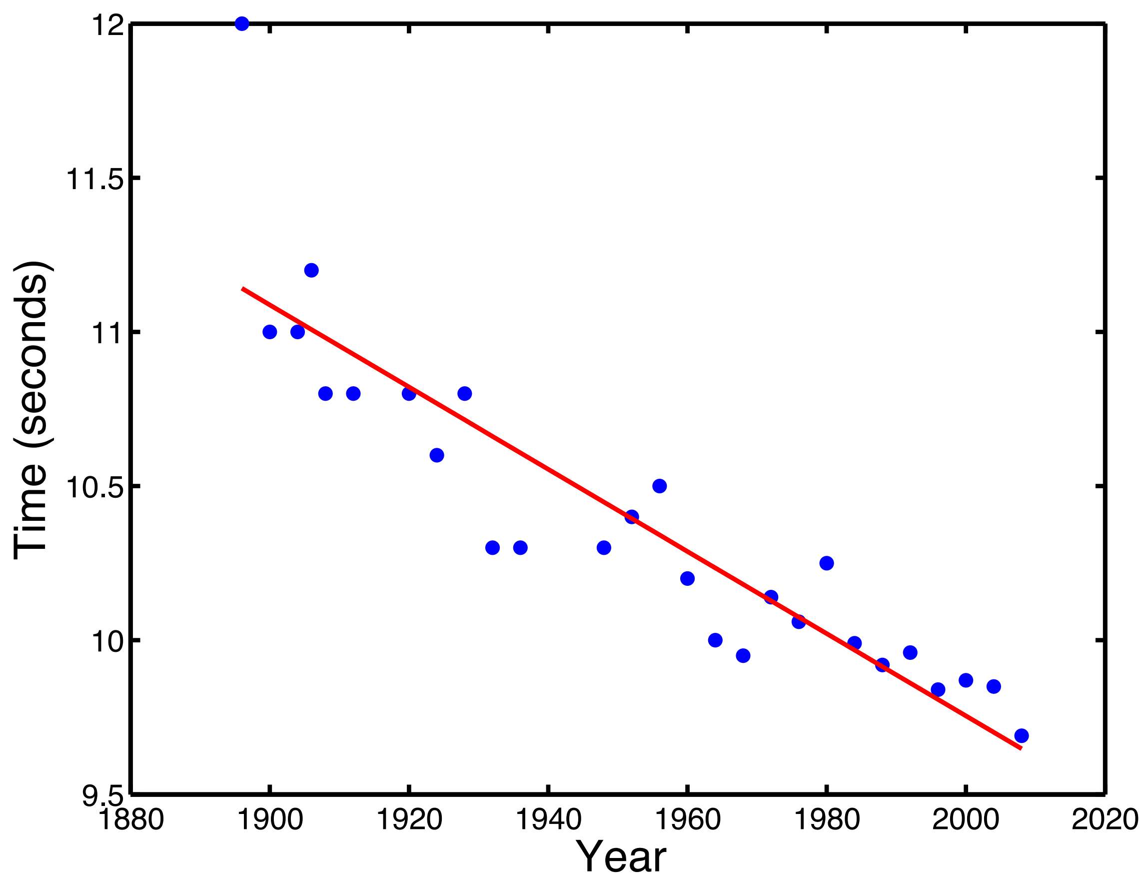
Our parameters can grow very large in magnitude - complex model

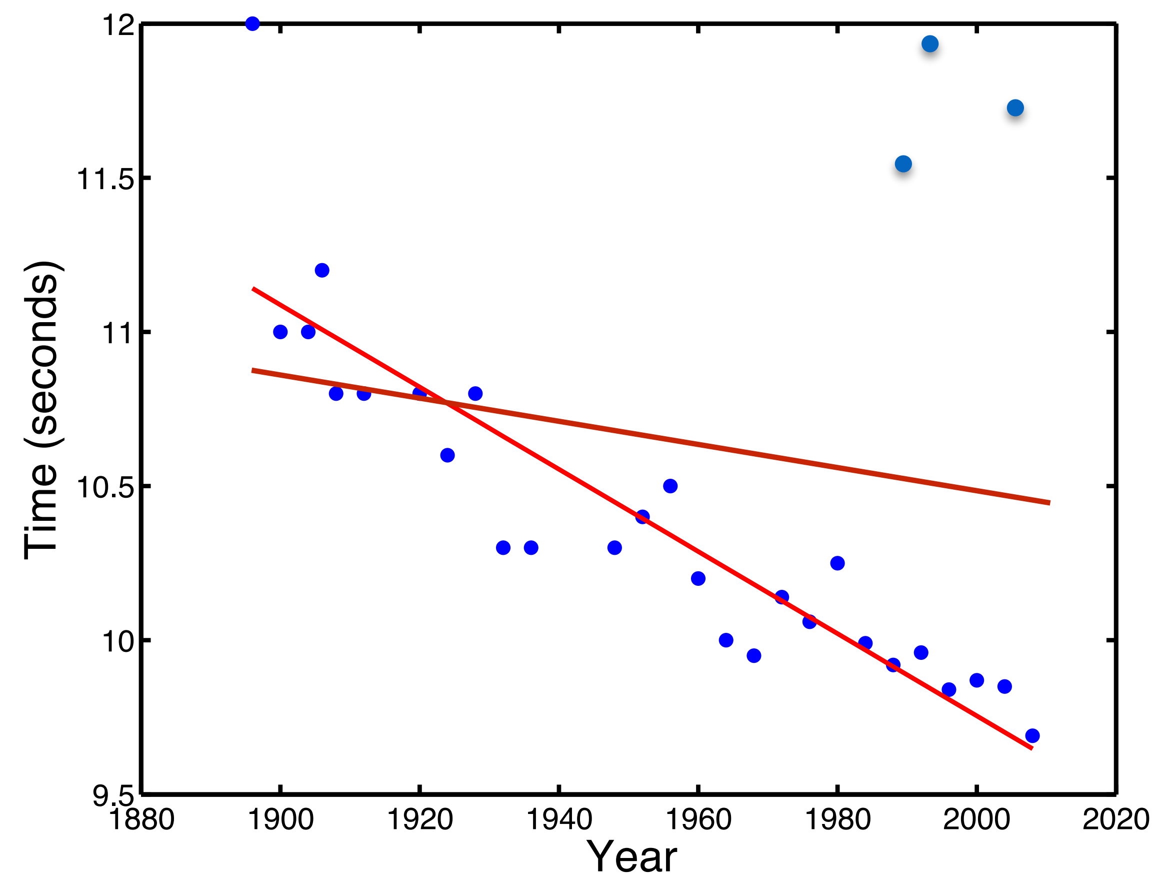
"Outliers": *observations located far away from the rest of the data*.

They can dominate the OLS solution



Outliers strongly affect the OLS solution





And if we were fitting here a polynomial expansion with OLS we would get a complex model

So what can we do to “penalise very complex solutions”?

Department of COMPUTER SCIENCE

Regularisation: Coupling our parameters and Penalising

We need to encode our preference for simpler solutions!

We impose a “competition” between them and constraint their magnitudes.

Rogers & Girolami Ch. 1, p. 33-34

One way: keep this value low $$\sum_d w_d^2 = \mathbf{w}^\mathrm{T}\mathbf{w}$$

Update our Loss function to include it $$\mathcal{L}' = \mathcal{L} + \lambda \mathbf{w}^\mathrm{T}\mathbf{w}$$

Why this value? Why adding it to Loss function?

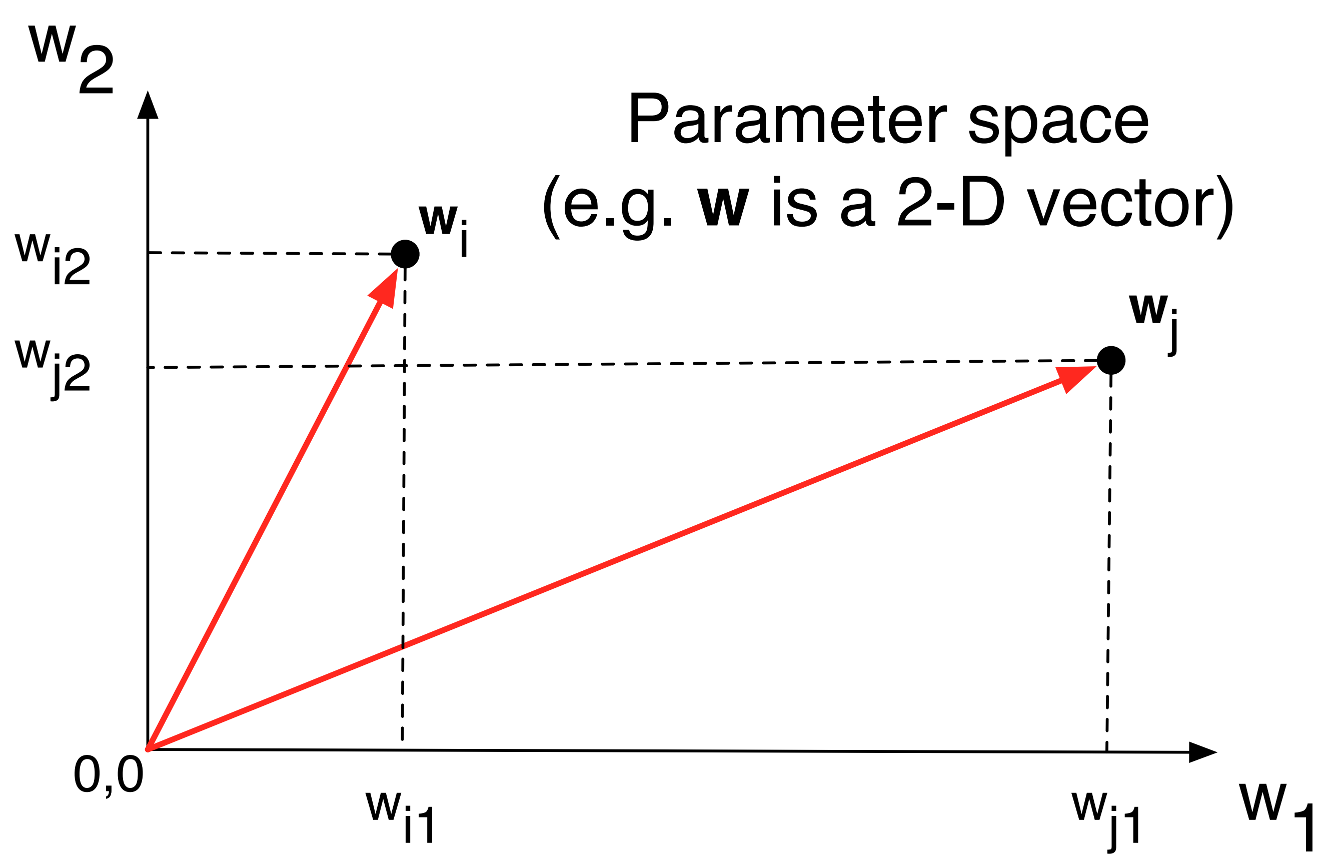
What is this regulariser we have added to the Loss?

$$\sum_d w_d^2 = \mathbf{w}^{\mathrm{T}}\mathbf{w}$$

The square of the Euclidean distance of **w** vector from centre of axis

Wait… isn't that the same as the square of the L_2 norm?

Euclidean (L2) Distance of **w** from (0,0)
= L2 norm of **w**



Regularisation as adding Constraints

Original OLS Loss :
(vector-matrix format)

$$\mathcal{L} = \frac{1}{N}(\mathbf{t} - \mathbf{X}\mathbf{w})^{\mathrm{T}}(\mathbf{t} - \mathbf{X}\mathbf{w})$$

I still want to minimise this loss **but subject to constraints**

constraint: "Keep parameters small by coupling their magnitudes"
or… The (square of the) L2 norm of **w** stays within a limit

$$\text{Minimise } \mathcal{L} \qquad \text{s.t. } \sum_d w_d^2 = \mathbf{w}^{\mathrm{T}}\mathbf{w} \leq \tau$$

Equivalent to minimising $\quad \mathcal{L}' = \mathcal{L} + \lambda \mathbf{w}^{\mathrm{T}}\mathbf{w}$

Penalised Least Squares: Ridge Regression

Known as: Penalised LS - **Ridge Regression**

Repeat derivation for new solution (PLS)

$$\widehat{\mathbf{w}} = (\mathbf{X}^{\mathrm{T}}\mathbf{X} + N\lambda\mathbf{I})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{t}$$

*(**Note**: X doesn't have a column of 1s now and its standardised)*

But what does the added constraint say in geometrical terms?

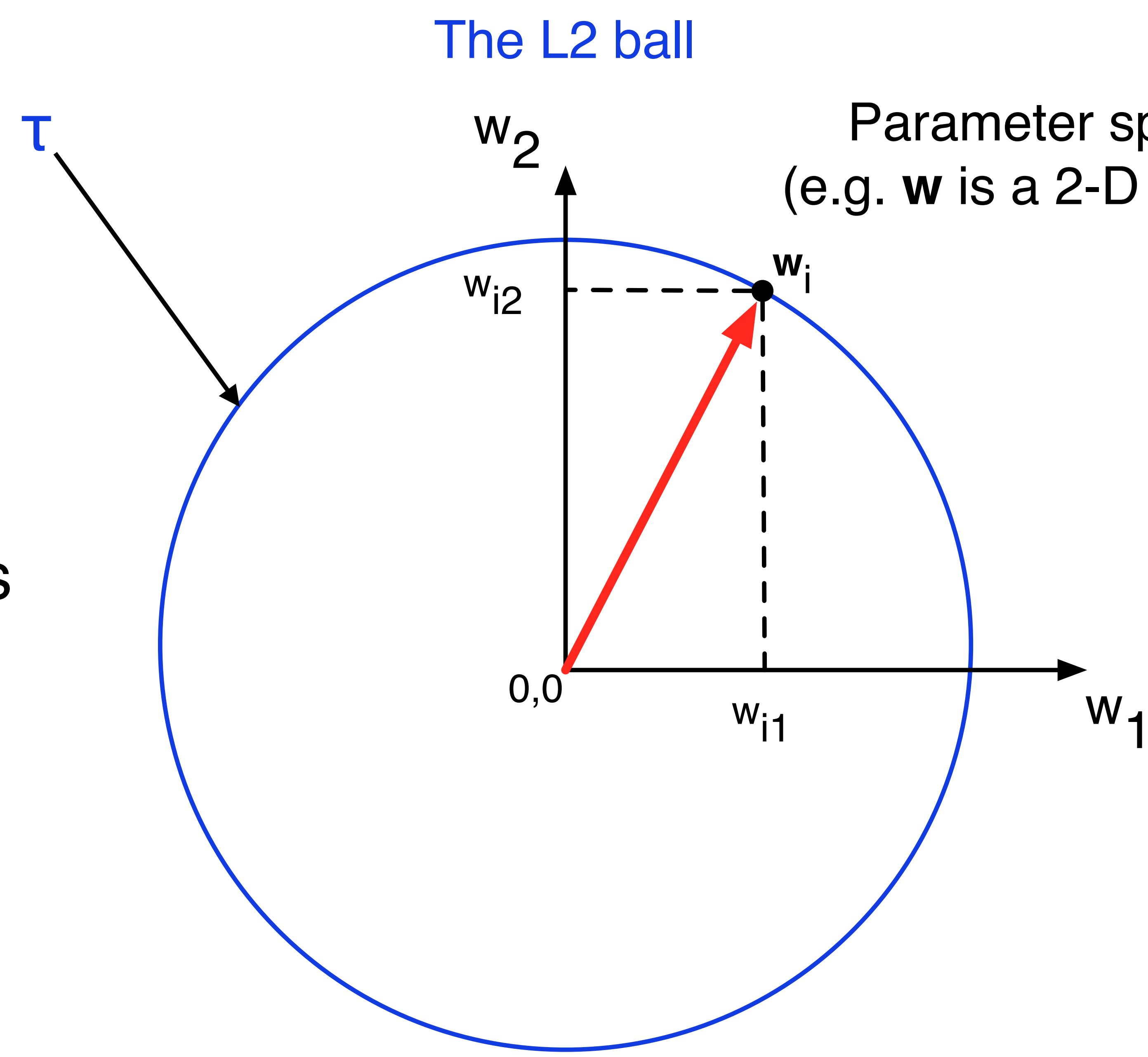
$$L_2^2(\mathbf{w}) = \sum_d w_d^2 = \mathbf{w}^{\mathrm{T}}\mathbf{w} \leq \tau$$

Square of Euclidean distance (L_2) of **w** from (0,0) within a limit τ

Lets forget about the square for now..

Geometry: the L2 ball and hyper-spheres

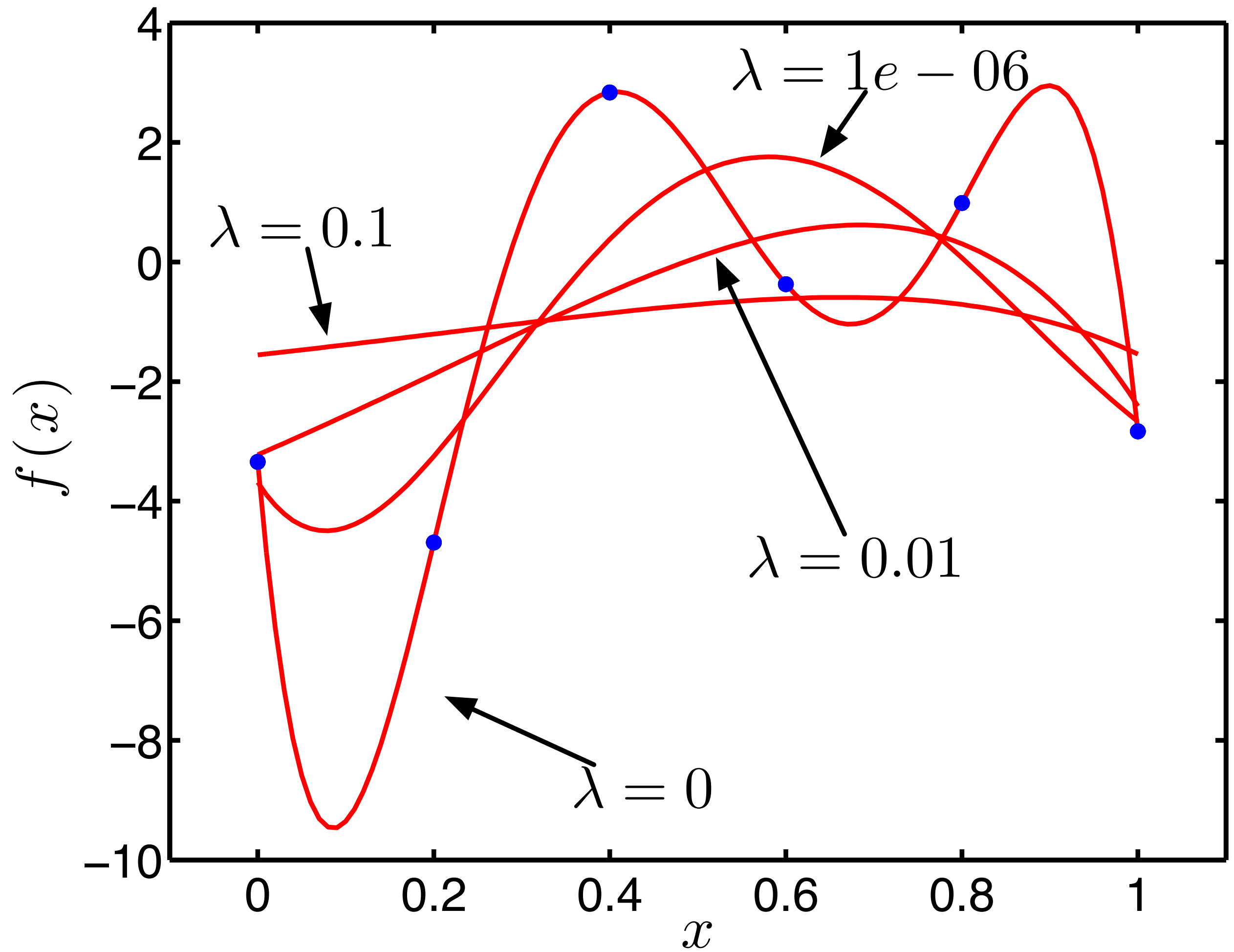
In higher-D this becomes a hyper-sphere (L2 ball)



Penalised Least Squares - Ridge Regression

6 datapoints, 5-order polynomial, varying the strength of regularisation

You can use CV to choose lambda, the strength of penalisation (complexity)



Summary so far

OLS

- OLS can easily overfit complex models and be mislead by outliers
- The magnitudes of the parameters **w** in OLS are not constrained
- Numerical instabilities (high variance) when correlated attributes

PLS / Ridge regression

- We “couple” the parameter magnitudes to constrain them
- We do that by adding a regulariser to the minimisation problem
- In PLS/Ridge regression the regulariser is $L_2^2(\mathbf{w})$
- The solution becomes: $\widehat{\mathbf{w}} = (\mathbf{X}^{\mathrm{T}}\mathbf{X} + N\lambda\mathbf{I})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{t}$
- Lambda controls the strength of regularisation (the volume of the ball)

If you change regulariser you get other algorithm(s)

The Lasso

http://statweb.stanford.edu/~tibs/lasso.html

Use another “stronger” regulariser instead of (a function of) the L_2

Completely shrink some **w**’s to 0

The Lasso

L_2 norm does not really guarantee "sparsity"
It will constraint the parameter magnitudes but will not necessarily set some to 0

L_1 norm is a stronger penaliser and will force parameters (of irrelevant/uncorrelated attributes) to 0

Geometry: L2 ball (hyper-sphere). What is the L1 norm?

L_p definition

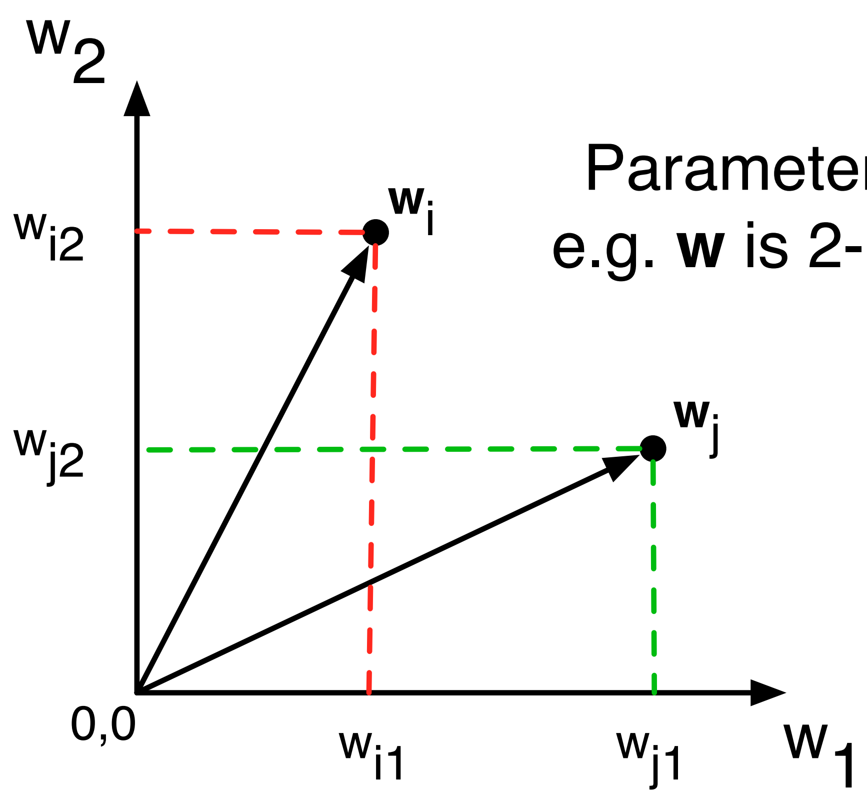
$$\mathrm{L}_p(\mathbf{w}) = \left(\sum_{d=1}^{D} |w_{nd}|^p \right)^{\frac{1}{p}}$$

Department of COMPUTER SCIENCE

L_1 norm

$$L_1(\mathbf{w}) = \sum_{d=1}^{D} |w_d|$$

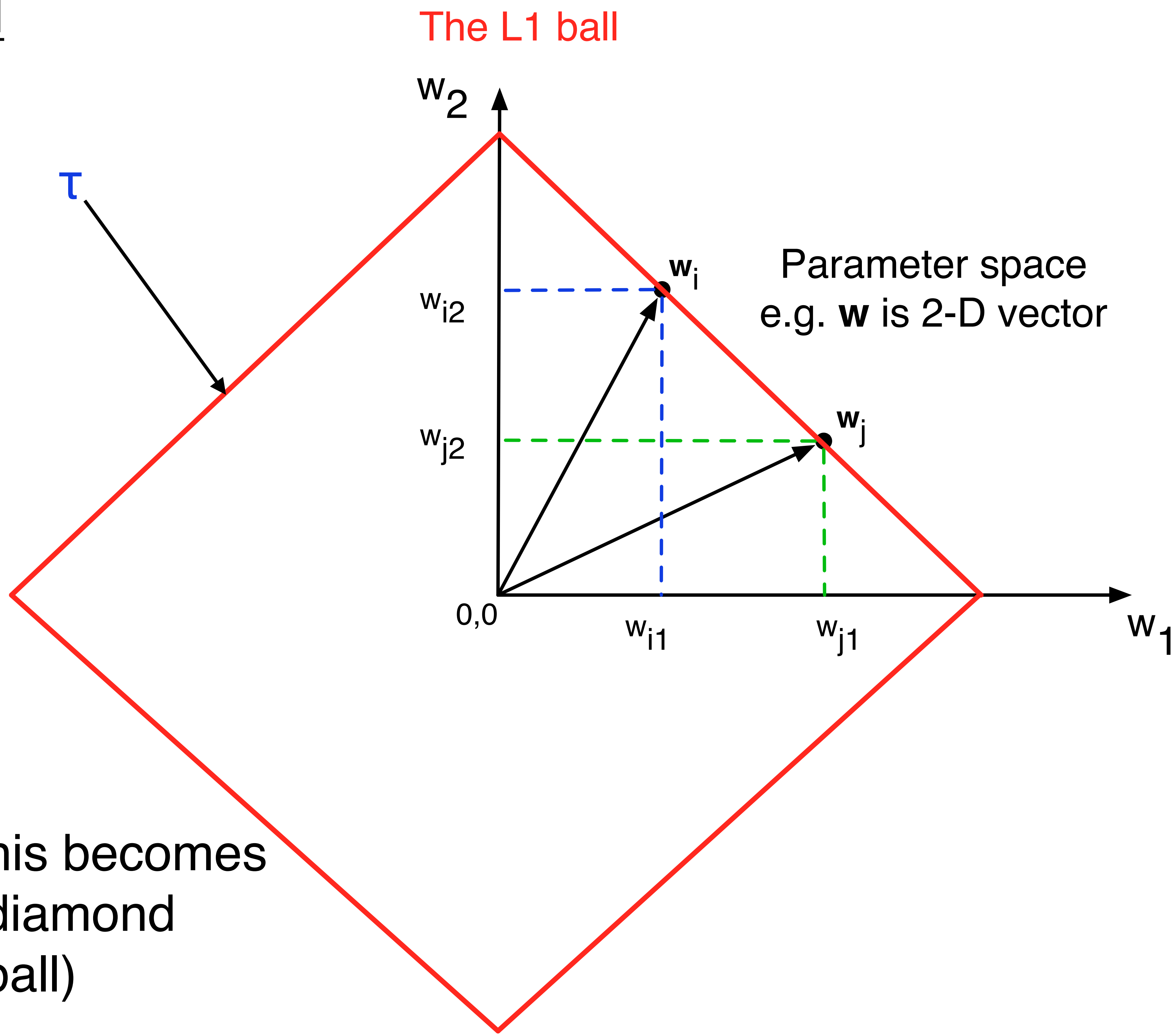
Manhattan (L1) Distance of **w** from (0,0)
= L1 norm of **w**



Parameter space
e.g. **w** is 2-D vector

So if I keep that distance constant and draw all the **w**'s
What shape will I get?

The L_1 ball



In higher-D this becomes
a hyper-diamond
(L1 ball)

Some of the L_p balls

Volume of L_p balls in 3-D for L_p <= 1

Regularisation with L_2 is also known as Tikhonov regularisation

The Lasso vs Ridge Regression

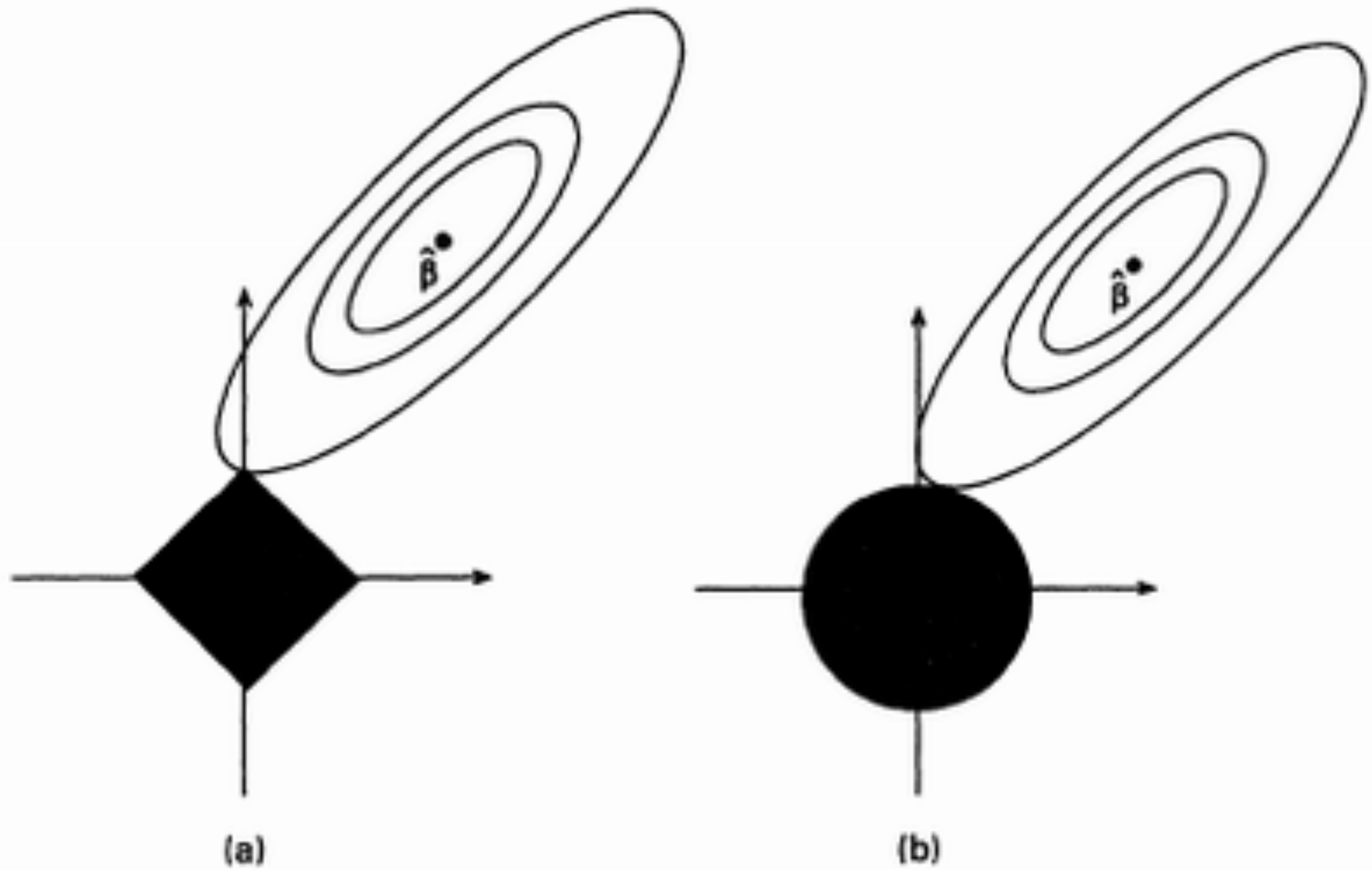


Fig. 2. Estimation picture for (a) the lasso and (b) ridge regression

The Lasso will recover *sparse* solutions where some parameters will be 0. Hence the corresponding attribute will be ignored

The Lasso

Original OLS Loss :
(vector-matrix format)

$$\mathcal{L} = \frac{1}{N}(\mathbf{t} - \mathbf{X}\mathbf{w})^{\mathrm{T}}(\mathbf{t} - \mathbf{X}\mathbf{w})$$

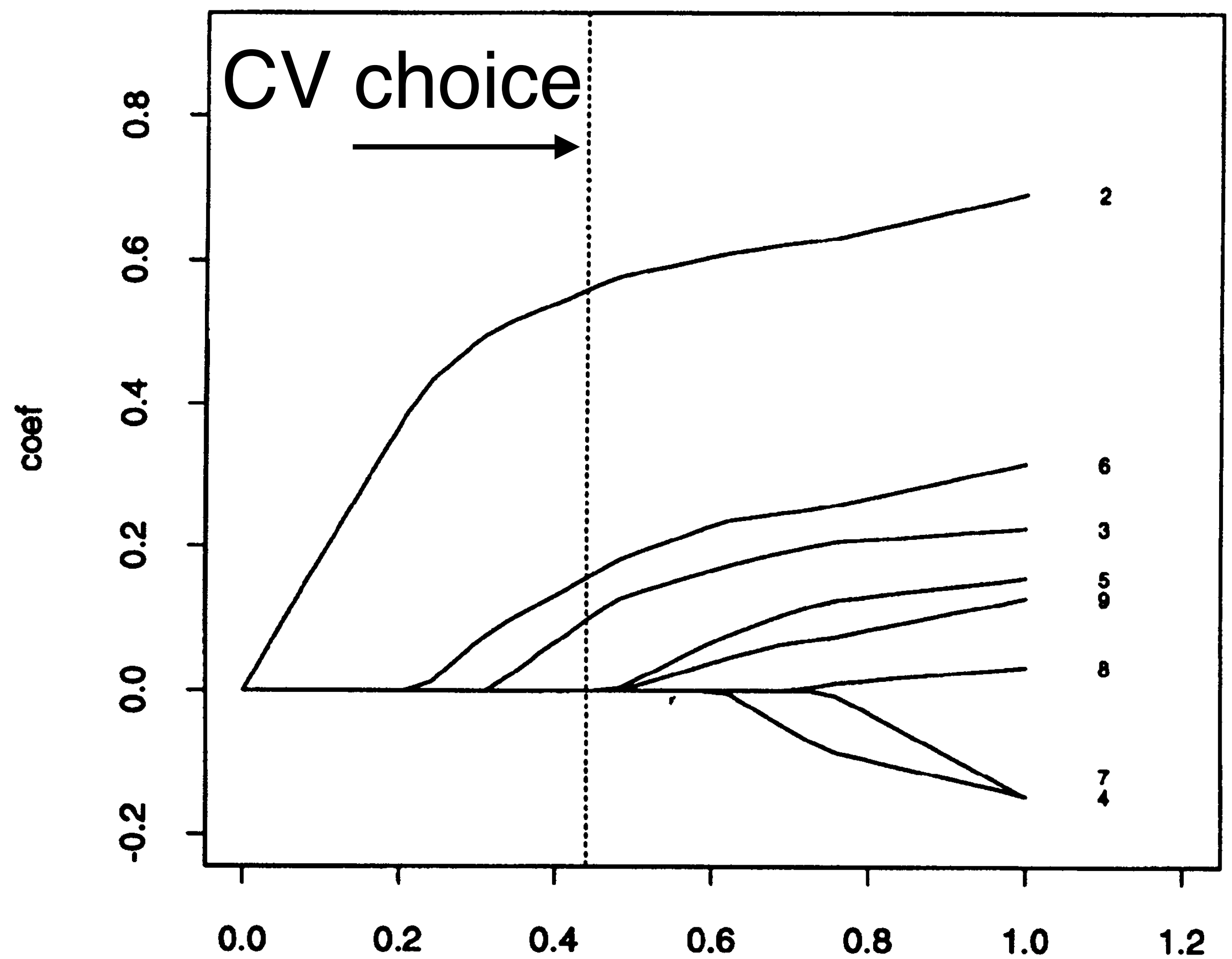
$$\text{Minimise } \mathcal{L} \qquad \text{s.t. } \sum_d |w_d| \leq \tau$$

Equivalent to minimising $\mathcal{L}' = \mathcal{L} + \lambda \sum_d |w_d|$

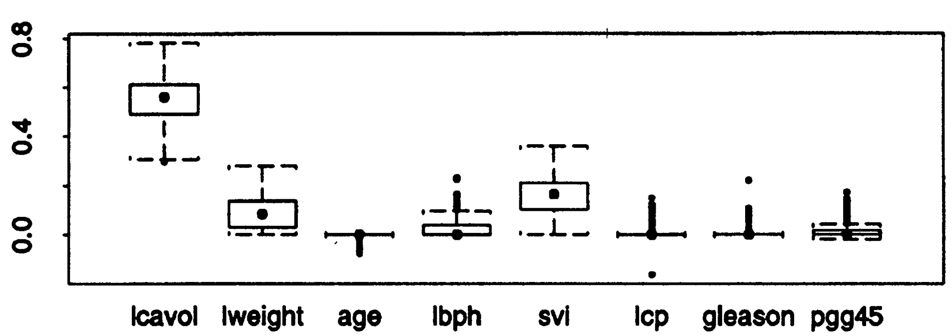
Quadratic programme (optimisation) - Use QP solver

Use cross-validation to pick strength of regularisation lambda

The Lasso: choosing λ (and hence the size of the L1 ball)



200 Bootstrap replications
with fixed λ



Summary of The Lasso

PLS / Ridge regression `sklearn.linear_model`**.Ridge**

- We "couple" the parameter magnitudes to constrain them
- We constraint parameters by regularising with squared **L_2 norm**
- Lambda controls the strength of regularisation (the volume of the ball)

The Lasso `sklearn.linear_model`**.Lasso**

- We "couple" the parameter magnitudes to constrain them
- We constraint parameters by regularising with the **L_1 norm**
- Sparse solutions with some parameters at 0
- Great for Interpretation
- May under-fit if our problem is not really sparse (use Ridge instead)
- Will outperform PLS when many attributes are irrelevant

Other variants (Elastic Net) with different regularisers (mixed norms)!