UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
School of Professional & Executive Development

Algorithms for linear regression

Lluís A. Belanche
Computer Science Department
belanche@cs.upc.edu

Big Data Management and Analytics

February 15, 2017

Introduction

Machine learning (ML) is the field of CS that studies automatic methods for developing **models** based on past experience (**data**) of a system:

- Many classical techniques in Multivariate Statistics (MS) are linear: PCA, Logistic Regression, (Ridge) Linear regression, Fisher's discriminant analysis, ...
- Many classical techniques in ML are non-linear: neural networks, kernel methods, random forests, ...
- The goals and problems are similar, and the techniques can often be rooted in the same (statistical) theory

The primary **goal** in ML is to produce good models

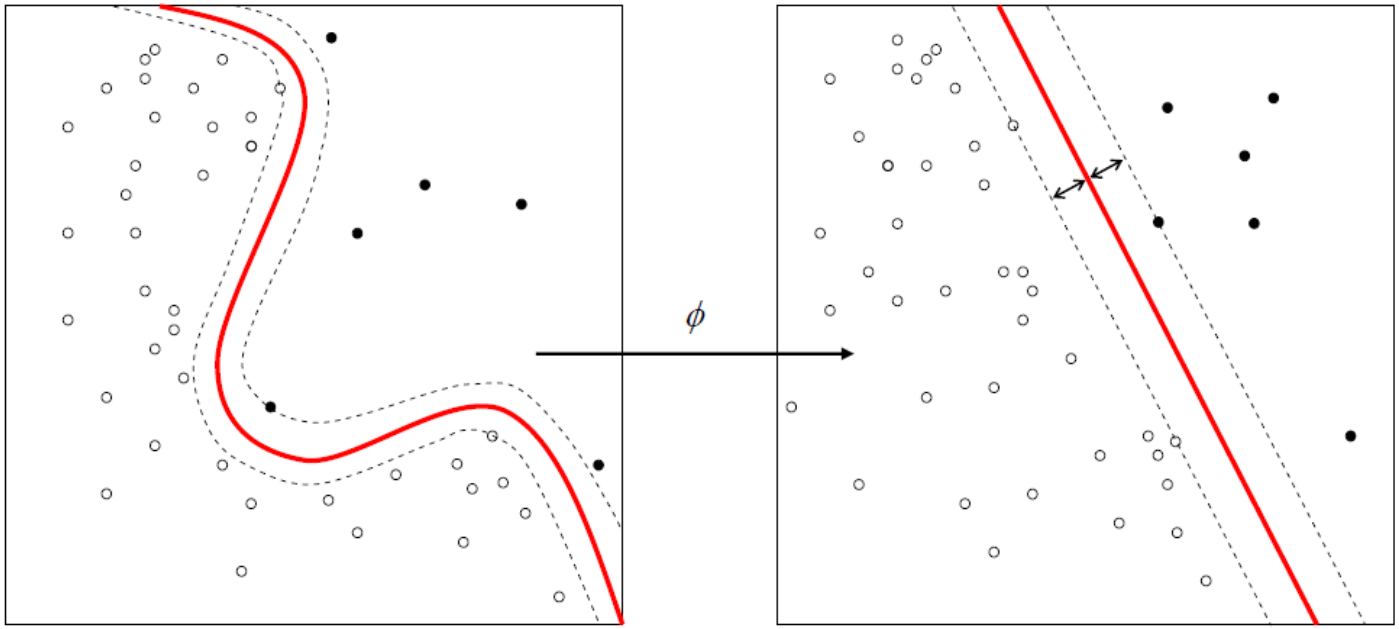
What is a model?

A **model** is a (usually) compact description of a data sample, that permits making **predictions**: statements about *unseen* examples

Desirable **properties** of models:

1. good generalization (MS, ML)
2. interpretability (MS)
3. amenable to inference (MS)
4. sparsity (MS?, ML)
5. efficiency (time, space) (MS?, ML)

Many statistical and machine learning techniques use some form of **regularization** to find the optimal **bias-variance** tradeoff in order to limit **overfitting**



Machine learning: an SVM in action

(from the Wikipedia)

Part 1

BIAS AND VARIANCE

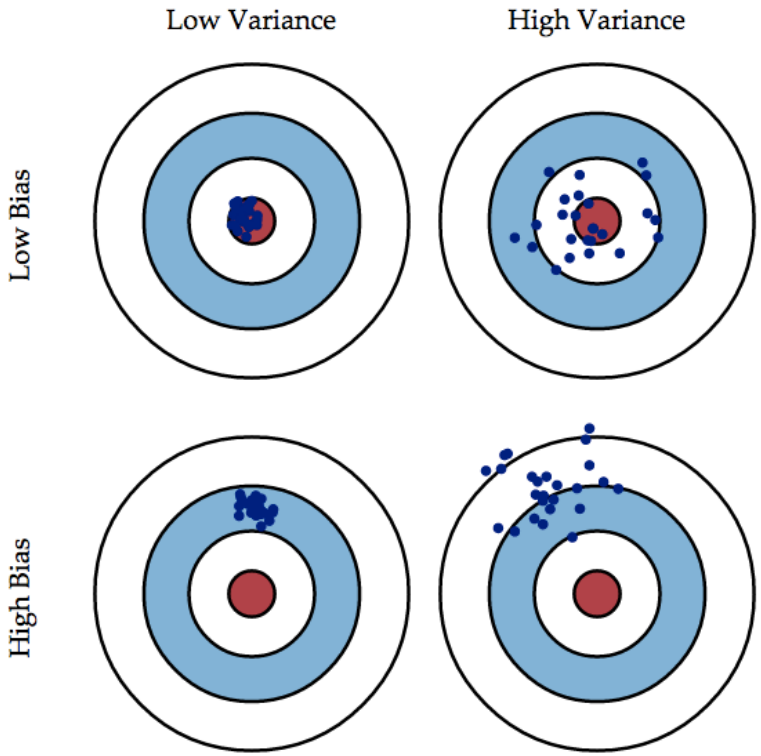
A gentle idea of bias and variance

Any learning algorithm has a prediction **error** that comes from three sources:

- **Bias** tendency to consistently learn wrong models by ignoring important information in the data
- **Variance** tendency to consistently learn wrong models by incorporating unimportant (randomly fluctuating) information in the data
- **Noise** intrinsic stochastic dependence between target and predictors

Ideally, we want to choose a model that accurately captures the regularities in its **training** data, and therefore **generalizes** well to unseen data (as well as possible)

A gentle idea of bias and variance

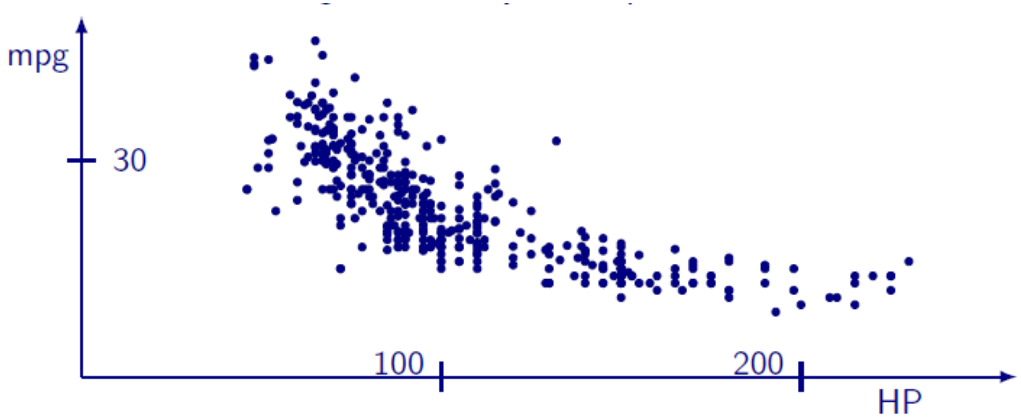


A dart-throwing example

Part 2

LINEAR REGRESSION THEORY

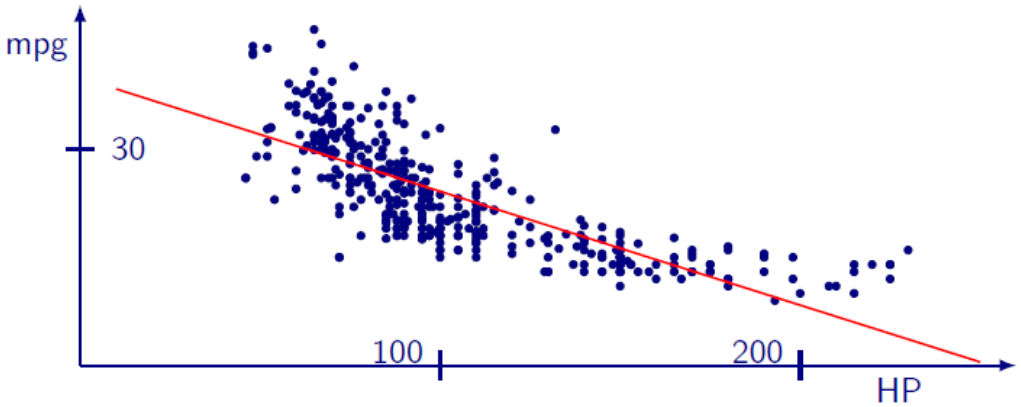
Linear regression



- A **regression** problem: predict some quantitative outcome subject to probabilistic uncertainty
- Example: predict gas mileage (mpg) of a car as a function of horsepower (HP)

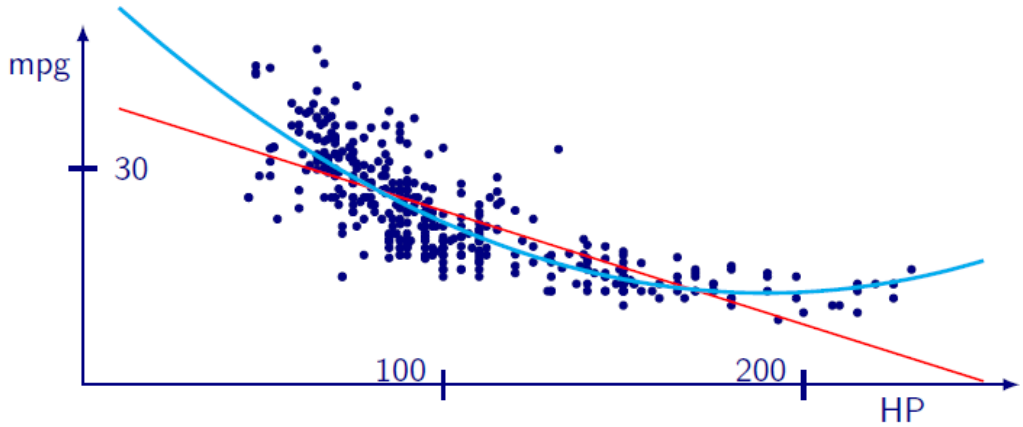
(auto-mpg data set from UCI Machine Learning Repository)

Linear regression



We can start by fitting a straight line to explain the relationship ...

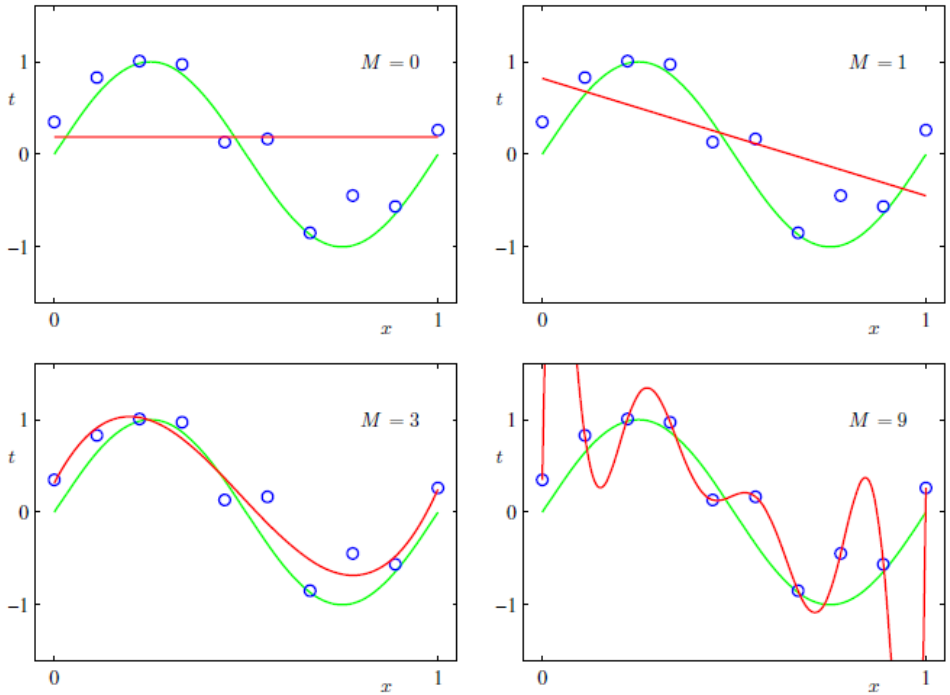
Linear regression



We can then fit a quadratic function ...

Is it a better model? Will it lead to better predictions?

Linear regression: problem setting



1. Which model is better? (it will lead to better predictions)
2. How can we choose it? (on the basis of the available data)

(from C. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2007)

Linear regression

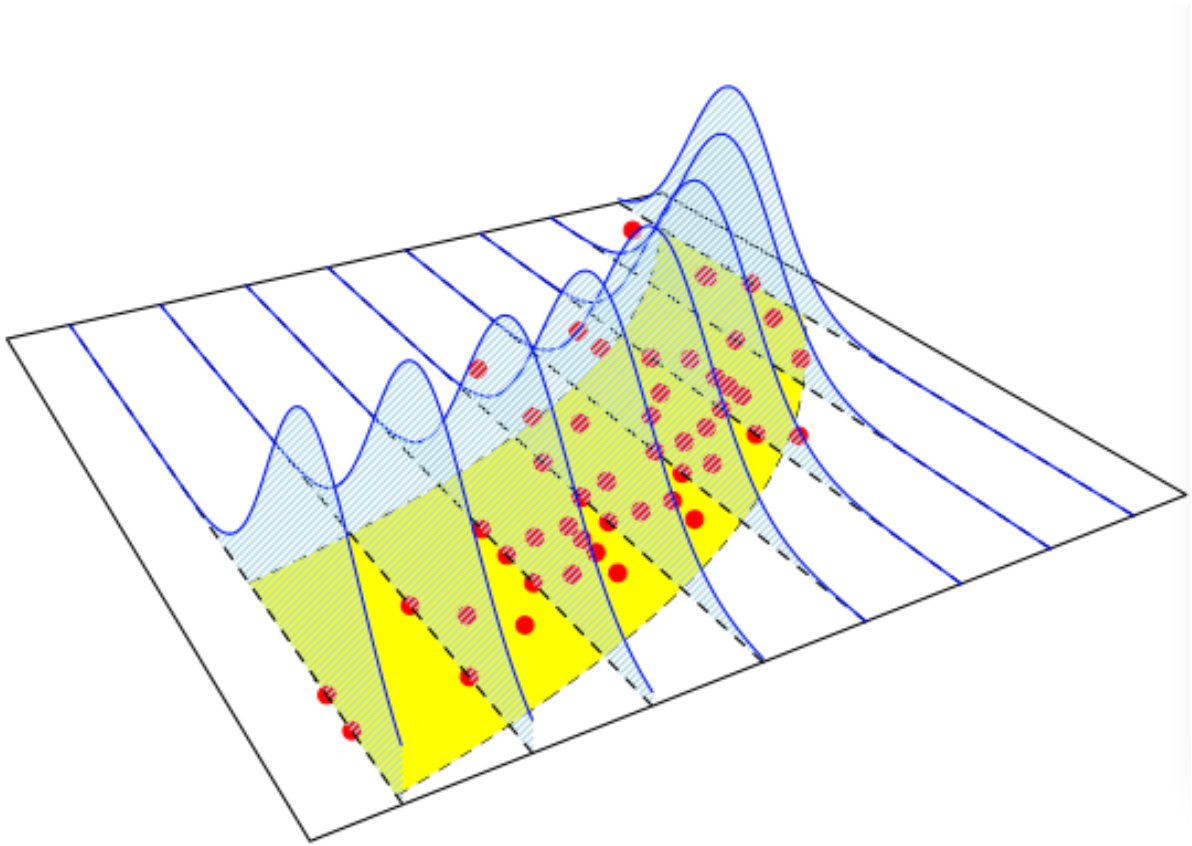
Statistical modeling of a continuous random variable (r.v.) T based on a finite number of examples. The departing **model** is

$$t_n = y(\mathbf{x}_n; \boldsymbol{\beta}) + \varepsilon_n, \qquad \mathbf{x}_n \in \mathbb{R}^d, \ t_n \in \mathbb{R}$$

- ε_n is a continuous r.v. such that $\mathbb{E}[\varepsilon_n] = 0$ and $Var[\varepsilon_n] = \sigma^2 < \infty$
- We choose $y(\mathbf{x}_n; \boldsymbol{\beta}) := \boldsymbol{\beta}^\top \mathbf{x}_n + \beta_0$
- A standard choice is $\varepsilon_n \sim \mathcal{N}(0, \sigma^2)$

$$y^*(\mathbf{x}) = \int_{\mathbb{R}} t \, p(t|\mathbf{x}) \, dt$$

known as the **regression function**



Interpretation of the modelling assumptions

Linear models

A model in $\mathbb{R}^d$ is **linear** if it can be written as a strictly monotonic function h of a linear combination of the **model parameters**:

$$y(\mathbf{x};\boldsymbol{\beta}) = h\Big(\beta_0 + \sum_{i=1}^{M} \beta_i \phi_i(\mathbf{x})\Big) = h\Big(\boldsymbol{\beta}^\top \phi(\mathbf{x})\Big)$$

with $\phi(\mathbf{x}) = (1, \phi_1(\mathbf{x}), \ldots, \phi_M(\mathbf{x}))^\top$ and $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_M)^\top$

Example

$$y(x;\boldsymbol{\beta}) = \beta_0 + \sum_{i=1}^{M} \beta_i x^i = \beta_0 + \beta_1 x + \ldots + \beta_M x^M, \qquad x \in \mathbb{R}$$

is a polynomial in x but a linear model in $\mathbb{R}$, where $\phi_i(x) = x^i$

Generalized Linear Models

GLMs allow for general conditional target distributions:

$$g(\mathbb{E}[T_n|\mathbf{X}_n]) = \boldsymbol{\beta}^\top \mathbf{X}_n + \beta_0$$

Generalized Linear Model

- A linear predictor of a convenient function of the expected value of the target variable, conditioned on the predictors
- This convenient function g is typically a smooth invertible function and called the **link function**
- The T_n are taken as i.i.d. and drawn from a distribution of the exponential family (Poisson, Gaussian, Binomial, Multinomial, Gamma, ...)

Generalized Linear Models

The idea is that the modeller chooses a suitable distribution (alt. a link function); in particular,

1. If g is the identity (Gaussian distribution): Linear regression
2. If g is the logit (Binomial distribution): Logistic regression
3. If g is the ln (Poisson distribution): Poisson regression

- This generality comes at a cost: in general we need an iterative procedure for the β parameters (model fitting)
- A very popular method is to set it up as a **Maximum Likelihood** problem and use a preferred numerical optimization method (e.g. **Newton-Raphson**)

Maximum likelihood

- We have a continuous r.v. X (e.g., the height of a randomly chosen Dutch)
- The population has some distribution, which is often assumed to have a special form. A common choice for a continuous distribution is the Gaussian (or Normal) density*:

$$p(x;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}\right),$$

and written $X \sim \mathcal{N}(x;\mu,\sigma^2)$, where μ, the mean, and σ^2, the variance, are the **parameters** θ of the distribution

---

(*) $\Pr\{X \in (x-\Delta x, x+\Delta x)\} \longrightarrow 2p(x)\Delta x$ as $\Delta x \longrightarrow 0$.

Maximum likelihood

Suppose we take an i.i.d. sample $D = \{x_1, \ldots, x_N\}$ of the r.v. X

- From the sample, we wish to **estimate** μ (it could be σ^2)
- It's not clear *a priori* what is the best way to do this:
  1. the average (mean) of D?
  2. the median of D?
  3. the average of the minimum and the maximum in D?

Maximum likelihood

The **likelihood** of seeing all the sample D is $\prod_{n=1}^{N} p(x_n; \mu, \sigma^2)$

Viewing this as *a function of the parameters*, we define

$$\mathcal{L}\left(\mu, \sigma^2; D\right) := P(D|\mu, \sigma^2) = \prod_{n=1}^{N} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2}\frac{(x_n - \mu)^2}{\sigma^2}\right)$$

"how likely it is that the population has parameters μ and σ^2 *given* the observed data sample D"

- The **maximum likelihood** estimators for the parameters are the values $\hat{\mu}$ and $\hat{\sigma}^2$ that maximize $\mathcal{L}\left(\mu, \sigma^2; D\right)$
- The likelihood is considered a function of θ for fixed data (whereas the density is considered a function of x for fixed θ)

Maximum likelihood

It is sometimes convenient (and equivalent) to maximize the **log-likelihood**:

$$l := \ln \mathcal{L}\left(\mu, \sigma^2; D\right) = \sum_{n=1}^{N} \ln p(x_n; \mu, \sigma^2)$$

In the Gaussian case, we have

$$l = \sum_{n=1}^{N} \left[\ln \frac{1}{\sqrt{2\pi}} - \ln \sigma - \frac{1}{2} \left(\frac{x_n - \mu}{\sigma} \right)^2 \right]$$

Maximum likelihood

Now

$$\frac{\partial l}{\partial \mu} = \frac{1}{\sigma^2} \sum_{n=1}^{N} (x_n - \mu),$$

so if $\frac{\partial l}{\partial \mu} = 0$, then $\hat{\mu} = \frac{1}{N} \sum_{n=1}^{N} x_n$, the average of the sample

Also $\frac{\partial^2 l}{\partial \mu \partial \mu} = -\frac{N}{\sigma^2} < 0$, and therefore we have found a maximum

The estimator for the variance is $\hat{\sigma}^2 = \frac{1}{N} \sum_{n=1}^{N} (x_n - \hat{\mu})^2$

Maximum likelihood

Example

Suppose we flip a coin that turns up heads with probability p. Find the ML estimator for p

- We take a sample $D = \{x_1, \ldots, x_N\}$ of N flips and get n_1 heads and $N - n_1$ tails. The number of heads follows a Binomial distribution $B(N, p)$
- The likelihood is $\mathcal{L}(p; D) = \binom{N}{n_1} p^{n_1} (1-p)^{N-n_1}$
- The log-likelihood is $l = \ln \binom{N}{n_1} + n_1 \ln p + (N - n_1) \ln(1-p)$

$$\frac{\partial l}{\partial p} = \frac{n_1}{p} - \frac{N - n_1}{1-p} = 0, \qquad \text{therefore} \qquad \hat{p} = \frac{n_1}{N}$$

Linear regression

We have a finite i.i.d. **learning data** sample of N *labelled* observations $D = \{(\mathbf{x}_n, t_n)\}_{n=1,\dots,N}$, where $\mathbf{x}_n \in \mathbb{R}^d$, $t_n \in \mathbb{R}$

Therefore our **statistical model** is $T_n \sim N(y(\mathbf{X}_n; \boldsymbol{\beta}), \sigma^2)$ or:

$$p(t_n|\mathbf{x}_n; \theta) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}\left(t_n - \boldsymbol{\beta}^\top \mathbf{x}_n\right)^2\right),$$

with parameters $\theta = \{\beta_0, \beta_1, \dots, \beta_d, \sigma^2\}$ and g the identity.

Note $\mathbb{E}[T_n|\mathbf{X}_n] = g^{-1}(\boldsymbol{\beta}^\top \mathbf{X}_n) = \boldsymbol{\beta}^\top \mathbf{X}_n$ (GLM setting)

We define $\mathbf{X} := (1, X_1, \dots, X_d)^\top$ and $\boldsymbol{\beta} := (\beta_0, \beta_1, \dots, \beta_d)^\top$

Linear regression

Define $\mathbf{t} := (t_1, \ldots, t_N)^\top$ and $X_{N\times(d+1)}$ the matrix of the $\mathbf{x}_n$.
Let us maximize the log-likelihood:

$$
\begin{aligned}
l(\theta) &= \ln \mathcal{L}(\theta) = \ln \prod_{n=1}^{N} p(t_n|\mathbf{x}_n;\theta) \\
&= \sum_{n=1}^{N} \ln p(t_n|\mathbf{x}_n;\theta) \\
&= -\frac{N}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{n=1}^{N}\left(t_n - \boldsymbol{\beta}^\top \mathbf{x}_n\right)^2 \\
&= -\frac{N}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}(\mathbf{t}-X\boldsymbol{\beta})^\top(\mathbf{t}-X\boldsymbol{\beta}) \\
&= -\frac{N}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\|\mathbf{t}-X\boldsymbol{\beta}\|^2
\end{aligned}
$$

$$\frac{\partial l}{\partial \boldsymbol{\beta}} = -\frac{1}{2\sigma^2}(-2X^\top \mathbf{t} + 2X^\top X\boldsymbol{\beta}) = \mathbf{0}$$
$$\frac{\partial l}{\partial \sigma^2} = -\frac{N}{2\sigma^2} + \frac{1}{2\sigma^4}(\mathbf{t} - X\boldsymbol{\beta})^\top(\mathbf{t} - X\boldsymbol{\beta}) = \mathbf{0}$$

Therefore,

$$\hat{\boldsymbol{\beta}} = (X^\top X)^{-1}X^\top \mathbf{t} = X^\dagger \mathbf{t}$$
$$\hat{\sigma}^2 = \frac{1}{N}(\mathbf{t} - X\hat{\boldsymbol{\beta}})^\top(\mathbf{t} - X\hat{\boldsymbol{\beta}}) = \frac{1}{N}\|\mathbf{t} - X\hat{\boldsymbol{\beta}}\|^2$$

Linear regression

- The matrix $X^\dagger := (X^\top X)^{-1} X^\top$ is known as the Moore-Penrose **pseudo-inverse** of X
- It is the generalization of the notion of inverse matrix to non-square matrices
- It has the property that $X^\dagger X = I$ (although in general $XX^\dagger \neq I$) (note both $X^\dagger X$ and $XX^\dagger$ are symmetric)

Typically $X^\dagger$ is computed with the **Singular Value Decomposition** (SVD) of X

Linear regression

- In the context of GLMs, $-2l = -2\ln\mathcal{L}$ is called the **deviance** (in ML, this is called the **error**)
- This would be $N\ln(2\pi\sigma^2) + \frac{1}{\sigma^2}\|\mathbf{t} - X\hat{\beta}\|^2 \equiv \|\mathbf{t} - X\hat{\beta}\|^2$
- A much better quantity to report is the NRMSE:

$$\mathrm{NRMSE}(\hat{\beta}) := \sqrt{\frac{\|\mathbf{t} - X\hat{\beta}\|^2}{(N-1)Var[\mathbf{t}]}}$$

In statistics, $R^2 = 1 - \mathrm{NRMSE}^2$ is the proportion of the (target) variability *explained* by the model

Trouble ahead!

The regression framework can yield unstable parameter estimates, especially when:

1. The explanatory variables are highly correlated
2. There is an insufficient number of observations relative to the number of predictors

A good technique is that of maximizing a **penalized** log-likelihood:

$$l_\lambda(\theta) := \sum_{n=1}^{N} \ln p(t_n|\mathbf{x}_n;\theta) - \frac{N\lambda}{2} R(\theta), \qquad \lambda > 0$$

In the context of regression with Gaussian noise (square error), the choice for the **regularizer** is $R(\boldsymbol{\beta};\sigma^2) = \|\boldsymbol{\beta}\|^2$ and we get:

$$l_\lambda(\theta) = -\frac{N}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\|\mathbf{t} - X\boldsymbol{\beta}\|^2 - \frac{N\lambda}{2}\|\boldsymbol{\beta}\|^2$$

1. Drop the first term, multiply by $-\sigma^{-2}$
2. Let λ "absorb" $N/\sigma^2 > 0 \Rightarrow$ minimize $\frac{1}{2}\|\mathbf{t} - X\boldsymbol{\beta}\|^2 + \frac{\lambda}{2}\|\boldsymbol{\beta}\|^2$
3. Set again the derivative w.r.t. $\boldsymbol{\beta}$ to $\mathbf{0}$:

$$(-2X^\top \mathbf{t} + 2X^\top X\boldsymbol{\beta}) + 2\lambda\boldsymbol{\beta} = \mathbf{0}$$

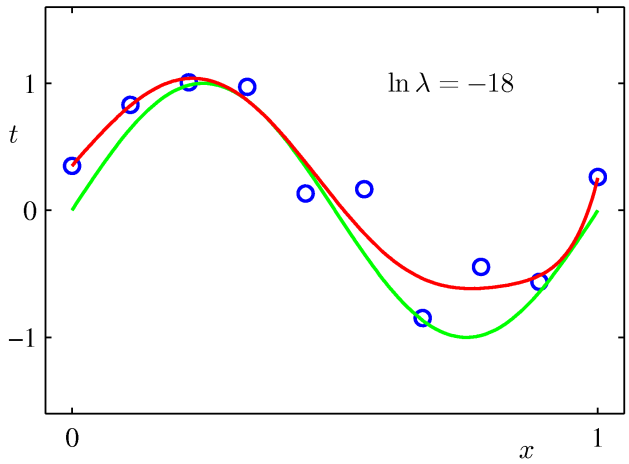
$$\hat{\boldsymbol{\beta}} = (X^\top X + \lambda I)^{-1} X^\top \mathbf{t}$$

- This technique is known as:
  - Tikhonov **regularization** in mathematics;
  - **ridge regression** in statistics; and
  - L_2-**regularization** in ML
- Advantages:
  1. Pushing the length of the parameter vector $\|\beta\|$ to 0 allows the fit to be under explicit control with the regularization parameter λ
  2. The matrix $X^\top X$ is p.s.d.; therefore $X^\top X + \lambda I$ is guaranteed to be p.d. (hence non-singular), for all $\lambda > 0$

Linear regression

How to do the **explicit control** on the fit?

- regularization permits the specification of models that are more complex than needed because it limits the effective complexity
- instead of trial-and-error on complexity, we can set a large complexity and adjust the λ



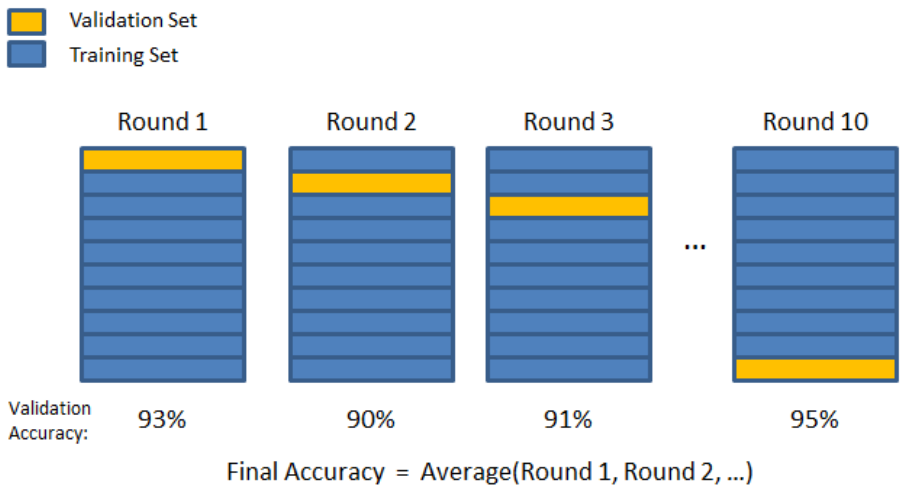
Fitting models with regularization ($M = 9$) (from Bishop's book)

Part 3
RESAMPLING

Resampling methods

We need to perform three different tasks:

1. **Fit** models to data (estimate coefficients)
2. **Choose** one of these models (based on prediction error)
3. **Estimate** the true performance of the chosen model



10-fold CV ($K = 10$)

(from `https://chrisjmccormick.files.wordpress.com/2013/07/10_fold_cv.png`)

Resampling methods

Method:

- **Training data** is used to fit models
- **Validation data** is used to average prediction errors and choose the model with the lowest prediction error
- The chosen model is **refit** using the full training data
- **Test data** is used to estimate true performance of the chosen model

How many folds are needed (the value of K)?

- With a large number of folds ...
  - $\Uparrow$ The bias of the true error rate estimator will be small
  - $\Downarrow$ The variance of the true error rate estimator will be large
  - $\Downarrow$ The computational time will be very large as well
- With a small number of folds ...
  - $\Uparrow$ The number of experiments and, therefore, computation time are reduced
  - $\Uparrow$ The variance of the estimator will be small
  - $\Downarrow$ The bias of the estimator will be large (conservative)
- In practice, the choice for K depends on the size of the dataset:
  - For large datasets, even 3-CV will be quite accurate
  - For small datasets, we may have to use leave-one-out CV (LOOCV or $K = N$) to train on as many examples as possible
  - Common choices are $K = 5, 10$

Regularization in practice

How to set the value of λ in ridge regression?

Using LOOCV, because

- λ is a very forgiving parameter (we usually perform a log search)
- there is a closed efficient formula for the LOOCV (for **linear models**): generalized cross-validation or GCV

LASSO regression

The **LASSO** (Least Absolute Shrinkage and Selection Sperator) is L_1-Regularized Linear regression

The choice for the regularizer is $R(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\|_1$ and we get to maximize:

$$\|\mathbf{t} - X\boldsymbol{\beta}\|^2 + \tau\|\boldsymbol{\beta}\|_1, \qquad \tau > 0$$

which turns out to be equivalent to the maximization of:

$$\|\mathbf{t} - X\boldsymbol{\beta}\|^2, \text{ subject to } \|\boldsymbol{\beta}\|_1 \leq \tau$$

LASSO regression

- In ridge regression, as λ is increased, all coefficients are reduced **while still remaining non-zero**
- In the LASSO, increasing τ causes more of the coefficients to be **driven to zero**
- In addition, as d increases, it is even more likely that some coefficients will be set to zero

---

Hence, the LASSO performs **feature selection**

Part 4

WRAPPING UP ...

Bias-Variance analysis

In regression, the prediction **risk** at any given data point $\mathbf{x}_0$ is the sum of three components:

The (squared) bias: average (square) deviation of our prediction at $\mathbf{x}_0$ and the best possible prediction

The variance: variability of our prediction as a function of the used sample (regardless of the underlying function!)

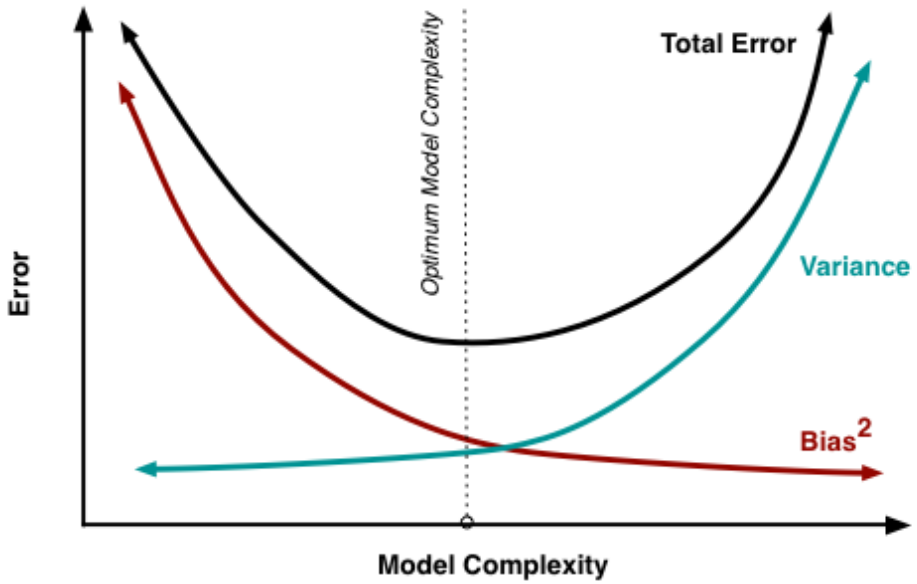
The variance noise: variability of the target value around its conditional mean

$$\text{Risk}(y_D(\mathbf{x}_0)) = \mathit{Bias}^2(y_D(\mathbf{x}_0)) + \mathit{Var}(y_D(\mathbf{x}_0)) + \sigma^2$$

Underfitting, overfitting and complexity

The "ability to fit" has a name: **complexity**

- "more complex than needed" models will have a large prediction error, dominated by the **variance** term (overfitted)
- "less complex than needed" models will have a large prediction error, dominated by the **(square) bias** term (underfitted)



Exercise: Yacht Hydrodynamics Data Set

The task is to predict the hydrodynamic performance of sailing yachts from basic hull dimensions and the boat velocity

- Prediction of residuary resistance of sailing yachts at the initial design stage is of a great value for evaluating the ship's performance and for estimating the required propulsive power
- The data set comprises 308 full-scale experiments, which were performed at the Delft Ship Hydromechanics Laboratory
- The **input** (predictive) variables are:
  1. Longitudinal position of the center of buoyancy, adimensional.
  2. Prismatic coefficient, adimensional.
  3. Length-displacement ratio, adimensional.
  4. Beam-draught ratio, adimensional.
  5. Length-beam ratio, adimensional.
  6. Froude number, adimensional.
- The **target** variable is the residuary resistance per unit weight of displacement

Exercise: Yacht Hydrodynamics Data Set

Play with the dataset, trying to find the best possible predictive model:

1. Take the log of the target?
2. Scale and center the data?
3. Consider some ordered variables?
4. Compare against ridge regression and the LASSO

Deliver a **two-page** pdf with what you finally did and what you got