

INTRODUCTION TO MACHINE LEARNING

TOPIC 1: BASIC CONCEPTS

LECTURE 1

Mattias Villani

**Division of Statistics and Machine Learning
Department of Computer and Information Science
Linköping University**



COURSE OUTLINE

► Nine topics:

- Basic concepts in machine learning. Software for ML.
- Regression, regularization and model selection
- Classification methods
- Dimensionality reduction and uncertainty estimation
- Support vector machines and kernel methods
- Gaussian processes and mixture models
- Splines and additive models
- Neural networks and deep learning
- Ensemble methods and high-dimensional problems

COURSE OUTLINE

► Course structure:

► Lectures.

- Approx 2 per topic.
- Concepts and theory.

► Labs.

- One per topic.
- Practical implemenations of methods.
- Individual report + aggregate group report
- Reports submission via LISAM

► Seminars.

- Two labs discussed at one seminar.
- Presentation of group reports. Randomly selected persons.

► Teachers: Mattias Villani, Oleg Sysoev and Isak Hietala.

OVERVIEW OF LECTURE 1

- ▶ What is machine learning?
- ▶ Motivating examples
- ▶ Unsupervised vs Supervised learning
- ▶ Regression vs Classification
- ▶ Generative vs Discriminative models
- ▶ Parametric vs nonparametric models
- ▶ Overfitting and Regularization
- ▶ Prediction and Model evaluation

OVERVIEW OF LECTURE 2

- ▶ Introduction to Bayesian learning
- ▶ Bernoulli model with beta prior
- ▶ Normal model with normal prior
- ▶ Multinomial model with Dirichlet prior

WHAT IS MACHINE LEARNING?

*Machine learning is a subfield of **computer science** that evolved from the study of **pattern recognition** and computational learning theory in **artificial intelligence**.*

*Machine learning explores the study and construction of **algorithms** that can **learn** from and make **predictions** on **data**. Such algorithms operate by building a model from example inputs in order to make data-driven predictions or **decisions**, rather than following strictly static program instructions.*

*Machine learning is closely related to and often overlaps with **computational statistics**; a discipline that also specializes in prediction-making.*

[Wikipedia \(Oct 11, 2015\).](#)

WHAT IS MACHINE LEARNING?

- ▶ Machine learning is an area in the **intersection** of **computer science**, **statistics** and **artificial intelligence**. It is closely related to **data mining**, **knowledge discovery** and **data science**.
- ▶ Machine learning uses mainly **statistical** (probabilistic) **models** for **analyzing data**. Data mining and knowledge discovery tend to use less rigorous, but often effective, algorithms.
- ▶ Machine learning differs from traditional statistics by a **heavier focus on prediction**, and lesser focus on interpretation.
- ▶ Models in machine learning are often **more flexible** (more parameters) than those in traditional statistics, and **regularization** to **avoid over-fitting** is therefore a much bigger concern.
- ▶ Machine learning applications often involve large data data sets (**big data**), and **computational complexity** of estimation algorithms is therefore important.

STATISTICS OR COMPUTER SCIENCE? BOTH!

I keep saying the sexy job in the next ten years will be statisticians.

Hal Varian, Chief Economist, Google.

*But the challenges for massive data go beyond the storage, indexing, and querying . . . and, instead, hinge on the ambitious goal of inference. **Inference** is the problem of **turning data into knowledge** . . . Statistical rigor is necessary to justify the inferential leap from data to knowledge . . .”*

from the report “Frontiers in Massive Data Analysis”, US National Research Council.

Computer scientists involved in building big-data systems must develop a deeper awareness of inferential issues, while statisticians must concern themselves with scalability, algorithmic issues, and real-time decision-making.

from the report “Frontiers in Massive Data Analysis”, US National Research Council.

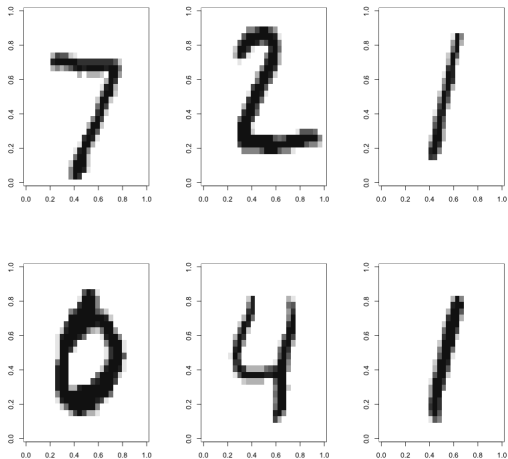
BUT WHY PROBABILITY MODELS?

- ▶ Probability models and statistical inference provide a **framework**.
- ▶ A principled **way to think** about any problem in machine learning.
- ▶ Probabilistic models can be **evaluated** in detail. Locate and understand the deficiencies in the model. Improve.
- ▶ Probabilistic models **quantify uncertainties**. Needed for data-driven decision making.

*As robotics is now moving into the open world, the issue of **uncertainty** has become a major stumbling block for the design of capable robot systems. Managing uncertainty is possibly the most important step towards robust real-world robot systems.*

from the book Probabilistic Robotics by Thrun et al.

CLASSIFYING HANDWRITTEN DIGITS



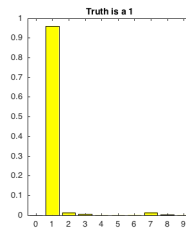
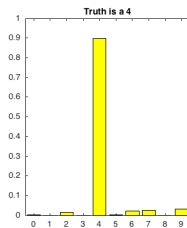
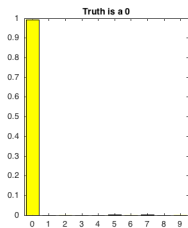
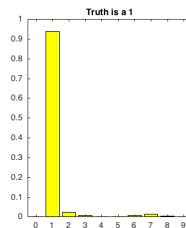
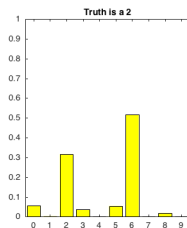
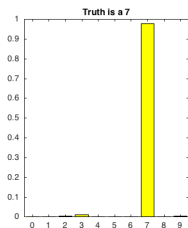
CLASSIFYING HANDWRITTEN DIGITS

- ▶ **Training** data: 60000 images.
- ▶ **Test** data: 10000 images.
- ▶ **Features**: use the intensities (0-255, but scaled to 0-1) in the $28 \times 28 = 784$ pixels as explanatory variables.
- ▶ **Multinomial regression**

$$Pr(\text{Digit} = i | \text{features}) = \frac{\exp(w_{0,i} + w_{1,i}x_1 + \dots + w_{784,i}x_{784})}{\sum_{j=0}^9 \exp(w_{0,j} + w_{1,j}x_1 + \dots + w_{784,j}x_{784})}$$

- ▶ MANY parameters to estimate. **Overfitting** is a major concern.
- ▶ **Lasso** (elastic net) **penalty** on model complexity. The weights $w_{k,i}$ are shrunk toward zero, sometimes exactly to zero.
- ▶ **Support vector machine** (**SVM**).

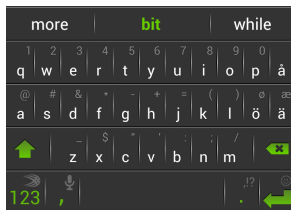
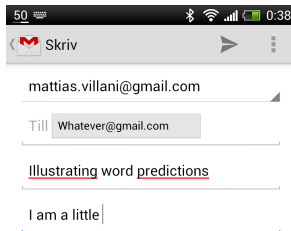
CLASSIFYING HANDWRITTEN DIGITS



HANDWRITTEN DIGITS - CONFUSION MATRIX

	0	1	2	3	4	5	6	7	8	9
0	966	0	8	1	1	7	9	2	4	6
1	0	1121	1	1	0	2	3	13	7	7
2	2	2	957	13	5	4	4	21	7	0
3	0	2	9	947	0	29	1	3	12	10
4	0	0	12	1	940	5	5	9	8	32
5	6	1	3	19	1	816	9	1	24	9
6	4	4	13	1	7	12	926	0	10	1
7	1	0	9	10	2	2	0	954	5	13
8	1	4	17	11	2	10	1	3	892	4
9	0	1	3	6	24	5	0	22	5	927

SMARTPHONE TYPING PREDICTIONS



SMARTPHONE TYPING PREDICTIONS

- ▶ Assume the following Markov model for a sentence

$$p(w_1 w_2 \cdots w_k) = p(w_1 | < s >) \cdot p(w_2 | w_1) p(w_3 | w_2) \cdots p(w_n | w_{n-1})$$

- ▶ Need a model for

$$p(w_k | w_{k-1})$$

- ▶ Example:

$$p(\text{person} | \text{stupid}) = 0.2$$

$$p(\text{Mattias} | \text{stupid}) = 0.0001$$

- ▶ Maximum Likelihood (ML) estimate:

$$\hat{p}(w_k | w_{k-1}) = \frac{\text{Number of times word } w_k \text{ follows directly after } w_{k-1}}{\text{Number of times } w_{k-1} \text{ appears in the text}}$$

- ▶ **Decision** problem: which word to suggest?
- ▶ SwiftKey is now using Neural Networks:
<https://youtu.be/-vZL3e02SnE>.

DETECTING BANKNOTE FRAUD

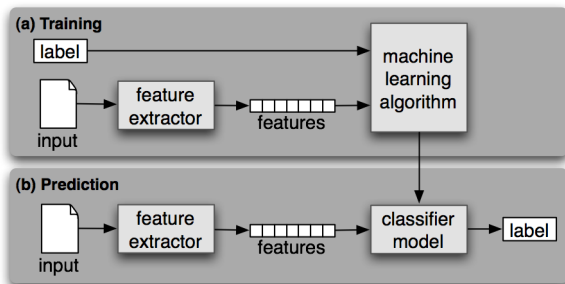
- ▶ Dataset with 1372 photographed banknotes. 610 of them are fraud.
- ▶ Raw data: $400 \times 400 = 160000$ gray-scale pixels.
- ▶ Often better performance if using a smaller set of summarizing variables, so called **features**, that capture the important aspects of the images.
- ▶ The 160000 pixel variables are condensed to four **features** using wavelets:
 - ▶ variance of Wavelet Transformed image
 - ▶ skewness of Wavelet Transformed image
 - ▶ curtosis of Wavelet Transformed image
 - ▶ entropy of image

DETECTING BANKNOTE FRAUD

- ▶ 1000 images for training.
- ▶ Predictions on 372 test images.
- ▶ **Decision:** signal fraud if $P(\text{Fraud}|\mathbf{x}) > 0.5$.
- ▶ **Confusion matrix**

		Truth	
		No fraud	Fraud
Decision	No Fraud	208	1
	Fraud	3	160

THE MACHINE LEARNING WORK FLOW



UNSUPERVISED VS SUPERVISED LEARNING

- ▶ **Unsupervised**: **target values** (responses/labels) are unknown.
Goals:
 - ▶ to discover groups or patterns in the data.
 - ▶ which observations are similar?
 - ▶ to learn a compact representation of the inputs. PCA.
- ▶ Example models for unsupervised learning:
 - ▶ **Mixture models**
 - ▶ **Association learning**
- ▶ **Supervised**: targets are known. Goals:
 - ▶ Build a prediction machine: inputs \rightarrow targets.
- ▶ Example models for supervised learning:
 - ▶ **Regression**: Linear regression. Gaussian processes.
 - ▶ **Classification**: Logistic regression, Support vector machines etc.
- ▶ **Semi-supervised**: targets are known only for some observations.
- ▶ **Active learning**. Strategies for deciding which observations to label.

HISTOGRAM AND KERNEL DENSITY ESTIMATION

- ▶ Histograms: Partition x -space into g bins of length Δ : B_1, \dots, B_g . Let n_i denote the number of observations falling in bin i .
- ▶ **Histogram estimate of the density**: density is constant over each bin and the density over bin i is

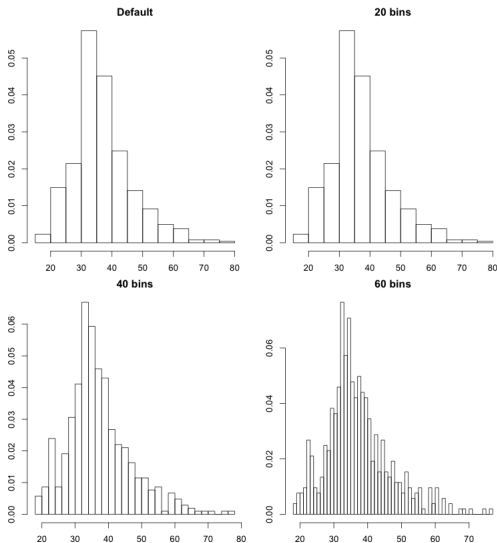
$$p(x) = \frac{n_i}{N\Delta} \text{ for } x \in B_i$$

- ▶ **Kernel density estimator**

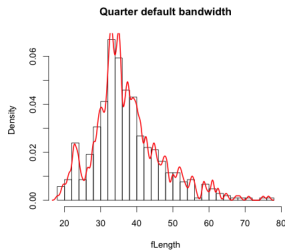
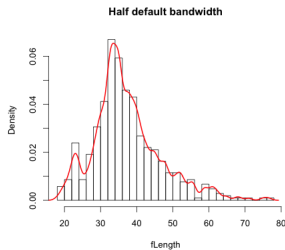
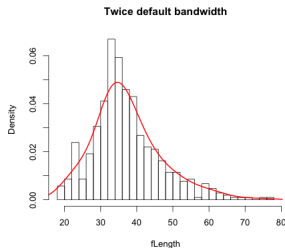
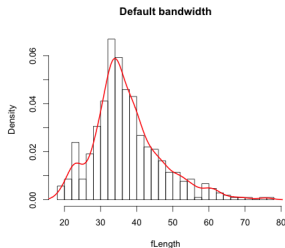
$$p(x) = \frac{1}{N} \sum_{i=1}^n \frac{1}{h} k\left(\frac{x - x_i}{h}\right)$$

where $k(\cdot)$ is a kernel function, e.g. the normal density. h is a bandwidth parameter, e.g. the standard deviation σ in a normal density.

FISH LENGTH - HISTOGRAM DENSITY ESTIMATES



FISH LENGTH - KERNEL DENSITY ESTIMATES



K-NEAREST NEIGHBOURS DENSITY ESTIMATION

- ▶ *k*-nearest neighbours (kNN):

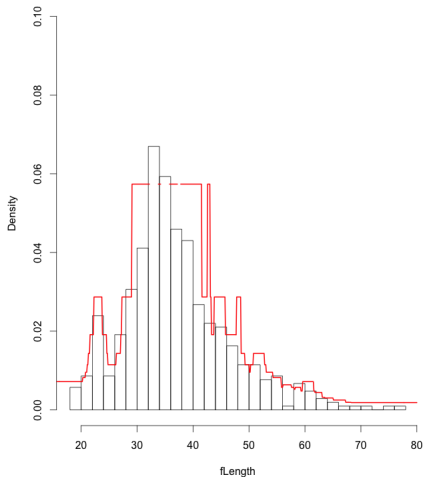
$$p(x) = \frac{K}{NV}$$

where

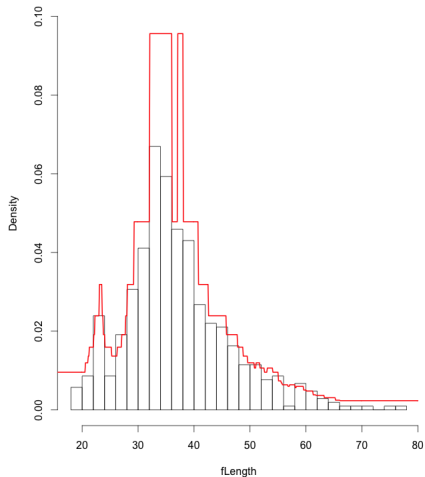
- ▶ N is the number of data observations
 - ▶ K is the number of neighbours
 - ▶ V is the width of the smallest bin that contains exactly the K neighbours of \mathbf{x} .
-
- ▶ Histograms: fix Δ and count the number of observations in a bin
 - ▶ *k*-nearest neighbours: fix the number of observations K in a bin, and make sure that the bin width V is large enough.

FISH LENGTH - KNN DENSITY ESTIMATES

k=15



k=25



MIXTURE OF NORMALS DENSITY ESTIMATION

- ▶ Two-component **mixture of normals**

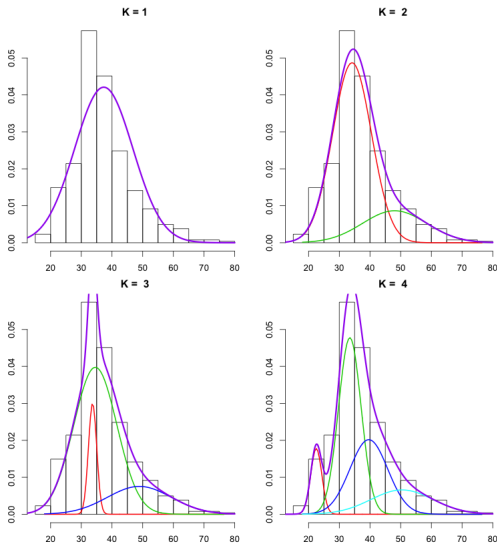
$$p(y) = \pi_1 \cdot \phi(y|\mu_1, \sigma_1^2) + \pi_2 \cdot \phi(y|\mu_2, \sigma_2^2)$$

- ▶ Mixture of normals with K components

$$p(y) = \sum_{k=1}^K \pi_k \cdot \phi(y|\mu_k, \sigma_k^2)$$

- ▶ Maximum likelihood or Bayesian methods to estimate the model parameters: π_1, \dots, π_K , μ_1, \dots, μ_K and $\sigma_1, \dots, \sigma_K$.
- ▶ It is possible to use other distributions than the normal in the mixture. Example: mixture of Poissons for count data.

FISH LENGTH - MIXTURE OF NORMALS



THREE TYPES OF SUPERVISED MODELS

- ▶ **Generative models:** models inputs and targets jointly:
 - ▶ $p(\mathbf{Y}, \mathbf{X})$ when the target Y is continuous
 - ▶ $p(\mathcal{C}_k, \mathbf{X})$ when the target is a class \mathcal{C}_k .
- ▶ **Discriminative models:** Models $p(\mathcal{C}_k | \mathbf{x})$ directly.
- ▶ **Discriminant function models:** uses a function $f(\mathbf{x})$ that maps any \mathbf{x} to a class label \mathcal{C}_k . No probabilities are involved.
- ▶ Generative models can be used to generate synthetic data.
- ▶ Discriminative models typically have fewer parameters than generative models.
- ▶ Discriminative models tend to give better predictions, especially when the generative model for \mathbf{x} is bad.

GENERATIVE MODELS - BAYESIAN CLASSIFICATION

- **Bayesian classification** among K classes

$$\operatorname{argmax}_{k \in \{1, \dots, K\}} p(\mathcal{C}_k | \mathbf{x})$$

where $\mathbf{x} = (x_1, \dots, x_n)$ is a feature vector.

- By **Bayes' theorem**

$$p(\mathcal{C}_k | \mathbf{x}) = \frac{p(\mathbf{x} | \mathcal{C}_k) p(\mathcal{C}_k)}{p(\mathbf{x})} \propto p(\mathbf{x} | \mathcal{C}_k) p(\mathcal{C}_k)$$

- Bayesian classification

$$\operatorname{argmax}_{k \in \{1, \dots, K\}} p(\mathbf{x} | \mathcal{C}_k) p(\mathcal{C}_k)$$

- $p(\mathcal{C}_k)$ can easily be estimated by the proportion of observations from class k .
- But how do we compute $p(\mathbf{x} | \mathcal{C}_k)$?

k -NEAREST NEIGHBOURS CLASSIFICATION

- ▶ **kNN** models the class-conditional distributions by

$$p(\mathbf{x}|\mathcal{C}_k) = \frac{K_k}{N_k \cdot V},$$

where

- ▶ V is the volume of the smallest sphere centered on \mathbf{x} that contains exactly K neighbours.
- ▶ K_k is number observations from class k in that sphere.
- ▶ N_k is the total number of observations in class k .
- ▶ kNN model of $p(\mathbf{x})$ and $p(\mathcal{C}_k)$

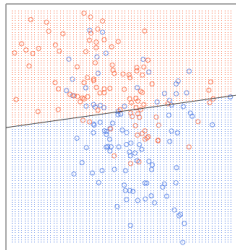
$$p(\mathbf{x}) = \frac{K}{N \cdot V} \quad p(\mathcal{C}_k) = \frac{N_k}{N}$$

- ▶ kNN classification probabilities:

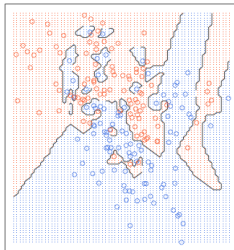
$$p(\mathcal{C}_k|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{p(\mathbf{x})} = \frac{K_k}{K}.$$

CLASSIFICATION

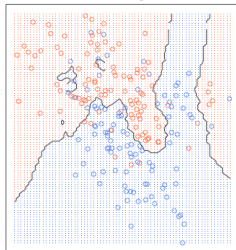
logistic regression



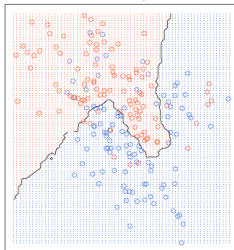
1-nearest neighbour



5-nearest neighbour



15-nearest neighbour



GENERATIVE MODELS - NAIVE BAYES

- ▶ **Naive Bayes**: features are assumed **independent**

$$p(\mathbf{x}|\mathcal{C}_k) = \prod_{j=1}^n p(x_j|\mathcal{C}_k)$$

- ▶ Naive Bayes solution

$$\operatorname{argmax}_{k \in \{1, \dots, K\}} \left[\prod_{j=1}^n p(x_j|\mathcal{C}_k) \right] p(\mathcal{C}_k)$$

- ▶ Example: $K = 2$. \mathcal{C}_1 = positive movie review and \mathcal{C}_2 = negative movie review
- ▶ Binary word features: $x_j = \text{has('fantastic')}$

$$\hat{p}(\text{has('fantastic')}|\text{positive}) = \frac{\# \text{ positive reviews with the word 'fantastic'}}{\# \text{ of positive reviews}}$$

- ▶ For **continuous features** we can use a normal distribution for each class:

$$x_j|\mathcal{C}_k \sim N(\mu_{jk}, \sigma_{jk}^2)$$

DISCRIMINATIVE MODELS

- ▶ Direct modeling of $p(\mathcal{C}_k|\mathbf{x})$. No need to model $p(\mathbf{x}|\mathcal{C}_k)$.
- ▶ **Logistic regression**

$$Pr(\text{positive review}|\mathbf{x}) = \frac{\exp(w_0 + w_1 \cdot x_1 + \dots + w_p x_p)}{1 + \exp(w_0 + w_1 \cdot x_1 + \dots + w_p x_p)}$$

- ▶ When the response is continuous: direct models of $p(Y|\mathbf{x})$.
- ▶ **Linear regression**:

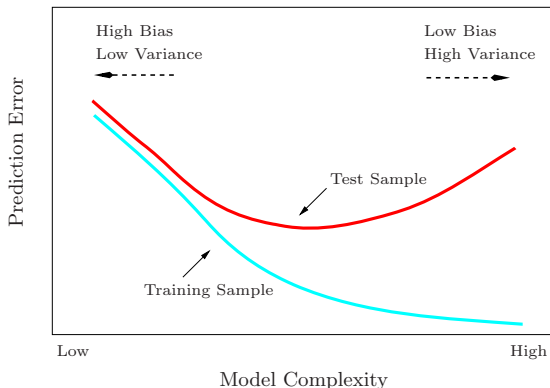
$$Y = w_0 + w_1 \cdot x_1 + \dots + w_p x_p + \varepsilon, \quad \varepsilon \stackrel{\text{indep}}{\sim} N(0, \sigma^2)$$

- ▶ **Poisson regression** for counts ($Y \in \{0, 1, 2, \dots\}$):

$$Y|\mathbf{x} \stackrel{iid}{\sim} \text{Poisson}[\lambda = \exp(w_0 + w_1 \cdot x_1 + \dots + w_p x_p)]$$

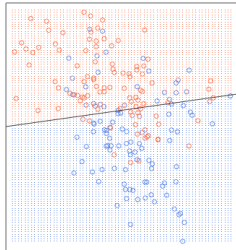
THE BIAS-VARIANCE TRADE-OFF

- ▶ Logistic regression has low variance, but may have large bias if truth is non-linear.
- ▶ kNN can have large variance, but low bias since it can adapt to non-linearities.
- ▶ Bias variance trade-off

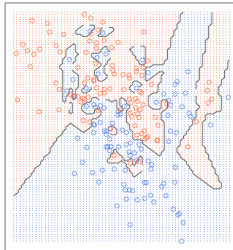


CLASSIFICATION

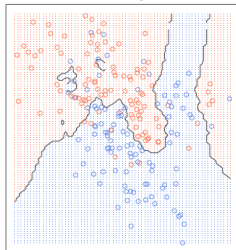
logistic regression



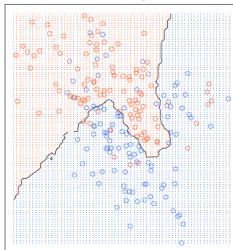
1-nearest neighbour



5-nearest neighbour

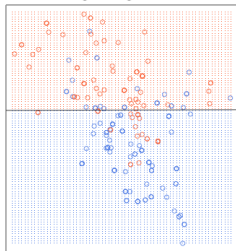


15-nearest neighbour

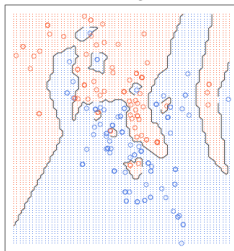


CLASSIFICATION - BOOTSTRAP SAMPLE 1

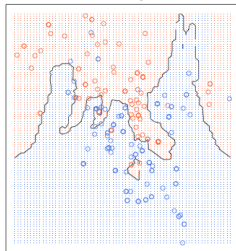
logistic regression



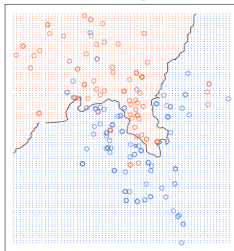
1-nearest neighbour



5-nearest neighbour

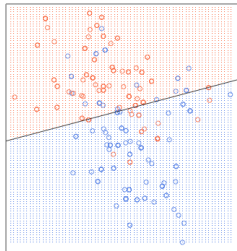


15-nearest neighbour

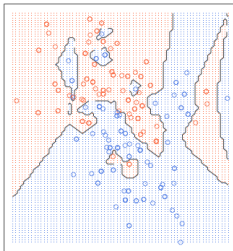


CLASSIFICATION - BOOTSTRAP SAMPLE 2

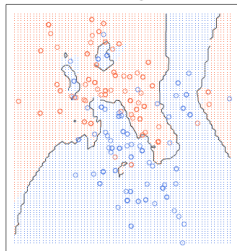
logistic regression



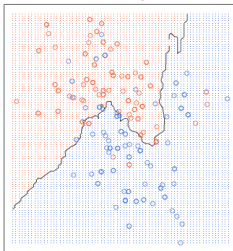
1-nearest neighbour



5-nearest neighbour

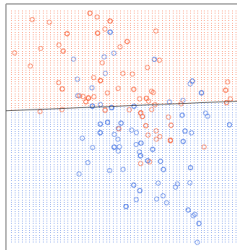


15-nearest neighbour

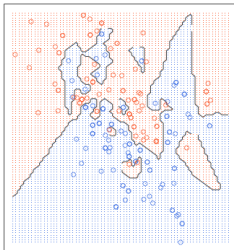


CLASSIFICATION - BOOTSTRAP SAMPLE 3

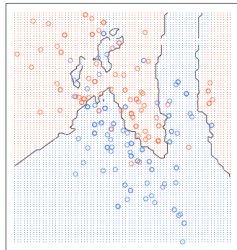
logistic regression



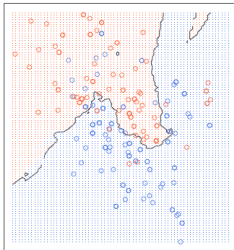
1-nearest neighbour



5-nearest neighbour

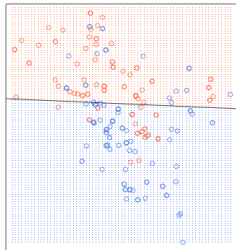


15-nearest neighbour

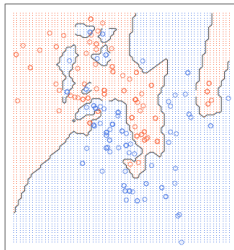


CLASSIFICATION - BOOTSTRAP SAMPLE 4

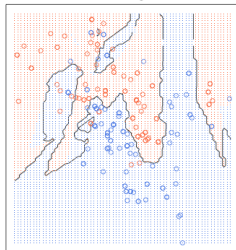
logistic regression



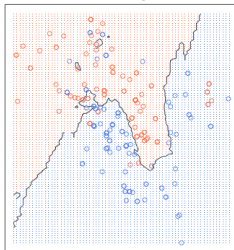
1-nearest neighbour



5-nearest neighbour

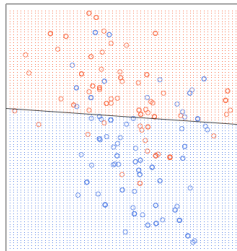


15-nearest neighbour

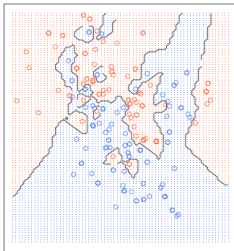


CLASSIFICATION - BOOTSTRAP SAMPLE 5

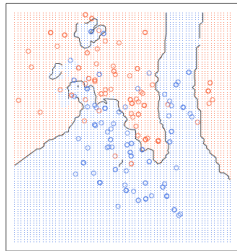
logistic regression



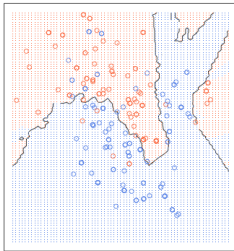
1-nearest neighbour



5-nearest neighbour



15-nearest neighbour



OVERFITTING

- ▶ Many problems in machine learning need **flexible models**:
 - ▶ Flexible mean (non-linear)
 - ▶ Flexible decision boundary in classification
 - ▶ Flexible distributions (heavy tails for outliers)
- ▶ But allowing for too much flexibility leads to **overfitting**.
- ▶ Overfitting leads to poor **generalization performance** on **new** data.

REGULARIZATION

- ▶ The solution to the flexible-without-overfitting dilemma is **regularization**.
- ▶ Example: polynomial regression

$$y = w_0 + w_1x + w_2x^2 + \dots + w_px^p + \varepsilon \quad \varepsilon \stackrel{iid}{\sim} N(0, \sigma^2)$$

- ▶ Two sides of the same (regularization) coin:
 - ▶ **Complexity penalty**

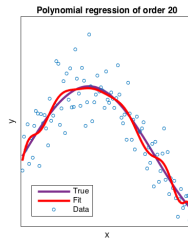
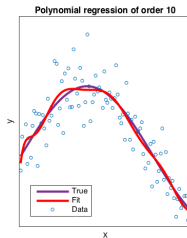
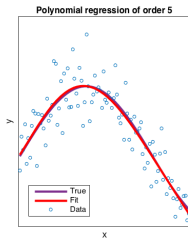
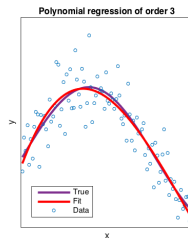
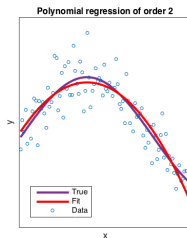
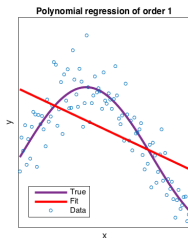
$$\log p(y_1, \dots, y_n | \mathbf{w}) - \lambda \sum_{j=1}^p w_j^2$$

- ▶ **Bayesian prior**

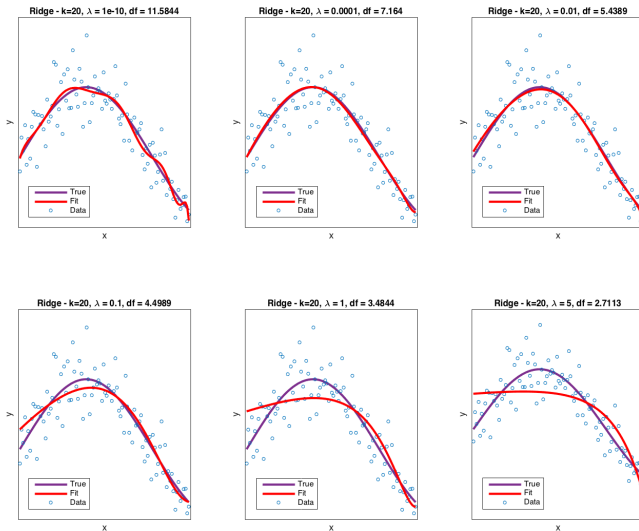
$$w_j \stackrel{iid}{\sim} N(0, \lambda^{-1})$$

- ▶ Regularization or **Shrinkage parameter** λ .

OVERFITTING - POLYNOMIALS



OVERFITTING - SMOOTHNESS PRIOR



PREDICTION AND MODEL EVALUATION

- ▶ The ultimate test of a model: does it **predict new data** well?
- ▶ Split the data into **Training** and **Test** sets.
- ▶ Sometimes split into Training, Validation and Testing.
- ▶ The test set is sometime called **hold-out sample**.
- ▶ **K -fold cross-validation**:
 - ▶ Partition the data into K sets (folds).
 - ▶ Use $K - 1$ folds for training and then predict the remaining fold.
 - ▶ Repeat K times, each time with a new fold as test.

EVALUATING A REGRESSION MODEL

- ▶ Point predictions based on features \mathbf{x}_i : $\hat{y}_i = f(\mathbf{x}_i)$.
- ▶ **Mean Squared Prediction Errors** (MSPE)

$$MSPE = \frac{1}{n_{test}} \sum_{i \in Test} (y_i - f(\mathbf{x}_i))^2$$

- ▶ $RMSPE = \sqrt{MSPE}$
- ▶ **Log Predictive Score** (LPS)

$$LPS = \frac{1}{n_{test}} \sum_{i \in Test} \log p(y_i | \mathbf{x}_i, \hat{\mathbf{w}})$$

- ▶ Om $p(y_i | \mathbf{x}_i, \hat{\mathbf{w}})$ is Gaussian, then $LPS \propto -MSPE$.

EVALUATING A CLASSIFIER: CONFUSION MATRIX

- **Confusion** matrix:

		Truth	
		Positive	Negative
Decision	Positive	tp	fp
	Negative	fn	tn

- tp = true positive, fp = false positive, fn = false negative, tn = true negative.
- Example: **spam/ham**

		Truth	
		Spam	Ham
Decision	Spam	tp	fp
	Ham	fn	tn

EVALUATING A CLASSIFIER: ACCURACY

- **Accuracy** is the proportion of correctly classified items

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fn + fp}$$

		Truth	
		Positive	Negative
Decision	Positive	tp	fp
	Negative	fn	tn

EVALUATING A CLASSIFIER: PRECISION

- **Precision** is the proportion of truly positive items among those signaled as positive:

$$\text{Precision} = \frac{tp}{tp + fp}$$

		Truth	
		Positive	Negative
Decision	Positive	tp	fp
	Negative	fn	tn

- High precision: when you say positive we can trust you to be right. People pointed out as fraudulent are almost always frauds.

EVALUATING A CLASSIFIER: RECALL

- ▶ **Recall** is the proportion of signaled positive items among those that are truly positive:

$$\text{Recall} = \frac{tp}{tp + fn}$$

		Truth	
		Positive	Negative
Decision	Positive	tp	fp
	Negative	fn	tn

- ▶ High recall: you will find the positive items. Don't be fraudulent, you will be caught.
- ▶ Recall is also called the **True Positive Rate (TPR)**
- ▶ There is a trade-off between Precision and Recall.

EVALUATING A CLASSIFIER: FALSE POSITIVE RATE

- False Positive Rate (FPR) is the proportion of signaled positive items among those that are truly negative:

$$\text{FPR} = \frac{fp}{fp + tn}$$

		Truth	
		Positive	Negative
Decision	Positive	tp	fp
	Negative	fn	tn

- Low FPR: you will very rarely signal a positive for negative items. People will not be falsely accused of fraud.

EVALUATING A CLASSIFIER: ROC CURVE

- ▶ Precision and recall depends on the **decision threshold**.
- ▶ $Pr(\text{Spam}|\text{text in an email}) = 0.9$. Do we send it to the spam-box?
- ▶ Is $Pr(\text{Spam}|\text{text in an email}) > 0.5$ a good threshold?
- ▶ **Optimal decisions** depend on the consequences.
- ▶ **ROC-curve**: Receiver Operating Characteristic.
- ▶ ROC: Plots the true positive rate (TPR) against the false positive rate (FPR) at various thresholds.
- ▶ **AUC** = Area Under Curve. Area under the ROC curve.

EVALUATING A CLASSIFIER: ROC CURVE

