Jaime S. Cardoso
jaime.cardoso@inesctec.pt
http://www.inescporto.pt/~jsc/VISUM/

INESC TEC and Faculdade de Engenharia, Universidade do Porto, Portugal

Introduction to Machine Learning

VISUM
July 05th, 2018, Porto, Portugal

Roadmap

- What's Machine Learning
- Distinct Learning Problems
- For the same problem, different solutions
- Different solutions but with common traits
- Avoiding overfitting and data memorization
- A fair judgement of your algorithm
- Some classical ML algorithms
- Beyond the classics

2

WHAT'S MACHINE LEARNING

3

An example*

- **Problem**: sorting incoming fish on a conveyor belt according to species
- Assume that we have only two kinds of fish:
  - Salmon
  - Sea bass

Picture taken with a camera

**Adapted from Duda, Hart and Stork, Pattern Classification, 2nd Ed.*

4

An example: the problem

What ***humans*** see

0	3	2	5	4	7	6	9	8
3	0	1	2	3	4	5	6	7
2	1	0	3	2	5	4	7	6
5	2	3	0	1	2	3	4	5
4	3	2	1	0	3	2	5	4
7	4	5	2	3	0	1	2	3
6	5	4	3	2	1	0	3	2
9	6	7	4	5	2	3	0	1
8	7	6	5	4	3	2	1	0

What ***computers*** see

5

An example: decision process

- What kind of information can distinguish one species from the other?
  - Length, width, weight, number and shape of fins, tail shape, etc.
- What can cause problems during sensing?
  - Lighting conditions, position of fish on the conveyor belt, camera noise, etc.
- What are the steps in the process?
  - Capture image -> isolate fish -> take measurements -> make decision

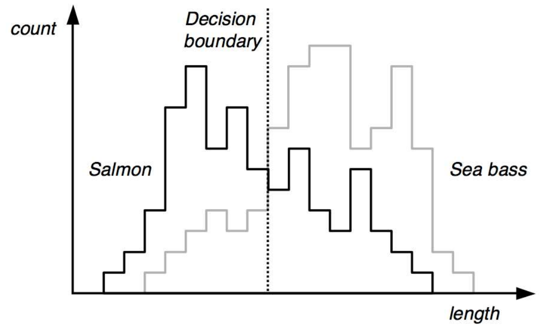
6

An example: our system

- **Sensor**
  - The camera captures an image as a new fish enters the sorting area
- **Preprocessing**
  - Adjustments for average intensity levels
  - Segmentation to separate fish from background
- **Feature Extraction**
  - Assume a fisherman told us that a sea bass is generally longer than a salmon. We can use **length** as a feature and decide between sea bass and salmon according to a threshold on length.
- **Classification**
  - Collect a set of examples from both species
    - Plot a distribution of lengths for both classes
  - Determine a decision boundary (threshold) that minimizes the classification error

7

An example: features



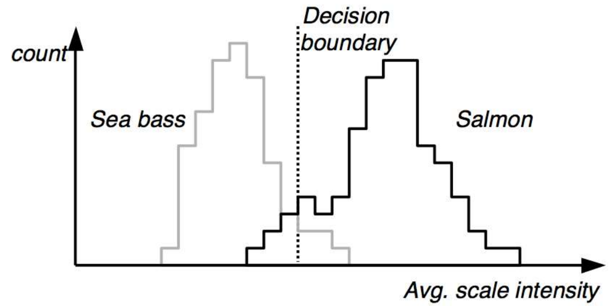
We estimate the system's probability of error and obtain a discouraging result of 40%. Can we improve this result?

8

An example: features

- Even though sea bass is longer than salmon on the average, there are many examples of fish where this observation does not hold
- Committed to achieve a higher recognition rate, we try a number of features
  - Width, Area, Position of the eyes w.r.t. mouth...
  - only to find out that these features contain no discriminatory information
- Finally we find a “good” feature: **average intensity of the fish scales**

An example: features



Histogram of the lightness feature for two types of fish in **training samples**. It looks easier to choose the threshold but we still can not make a perfect decision.

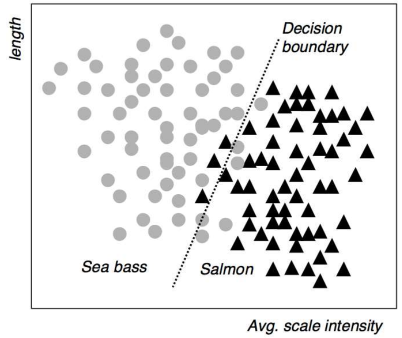
An example: multiple features

- We can use two features in our decision:
  - lightness: $\boldsymbol{x}_1$
  - length: $\boldsymbol{x}_2$
- Each fish image is now represented as a point (feature vector)

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

in a two-dimensional **feature space**.

An example: multiple features

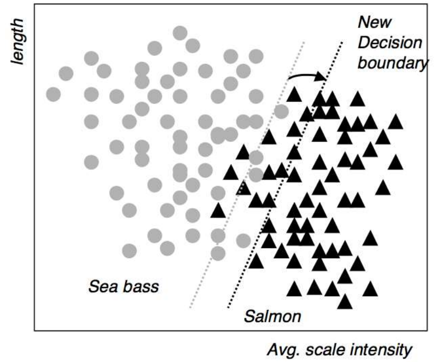


Scatter plot of lightness and length features for training samples. We can compute a **decision boundary** to divide the feature space into two regions with a classification rate of 95.7%.

An example: cost of error

- We should also consider **costs of different errors** we make in our decisions.
- For example, if the fish packing company knows that:
  - Customers who buy salmon will object vigorously if they see sea bass in their cans.
  - Customers who buy sea bass will not be unhappy if they occasionally see some expensive salmon in their cans.
- How does this knowledge affect our decision?

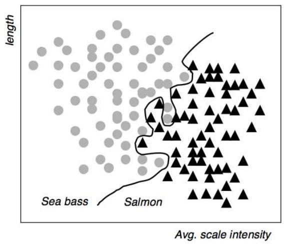
An example: cost of error



We could intuitively shift the decision boundary to minimize an alternative cost function

An example: generalization

- **The issue of generalization**
  - The recognition rate of our linear classifier (95.7%) met the design specifications, but we still think we can improve the performance of the system
  - We then design a classifier that obtains an impressive classification rate of 99.9975% with the following decision boundary



An example: generalization

- **The issue of generalization**
  - Satisfied with our classifier, we integrate the system and deploy it to the fish processing plant
  - A few days later the plant manager calls to complain that the system is misclassifying an average of 25% of the fish
- **What went wrong?**

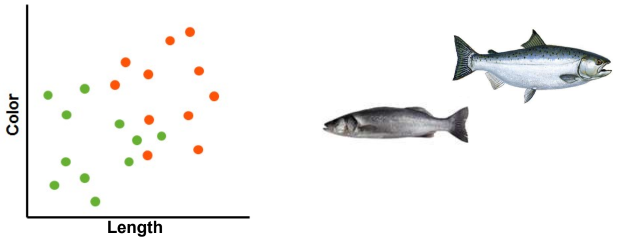
Data Driven Design

- When to use?
  - Difficult to reason about a generic rule that solves the problem
  - Easy to collect examples (with the solution)

17

Data Driven Design

- When to use?
  - Difficult to reason about a generic rule that solves the problem
  - Easy to collect examples (with the solution)



18

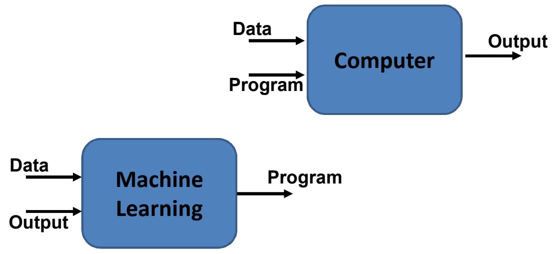
Data Driven Design

- There is little or no domain theory
- Thus the system will learn (i.e., generalize) from training data the general input-output function
  - Programming computers to use example data or past experience
- The system produces a program that implements a function that assigns the decision to any observation (and not just the input-output patterns of the training data)

19

What is Machine Learning?

- Automating the Automation



20

Data Driven Design

- A good learning program learns something about the data beyond the specific cases that have been presented to it
  - Indeed, it is trivial to just store and retrieve the cases that have been seen in the past
    - This does not address the problem of how to handle new cases, however
- Over-fitting a model to the data means that instead of general properties of the population we learn idiosyncracies (i.e., non-representative properties) of the sample.

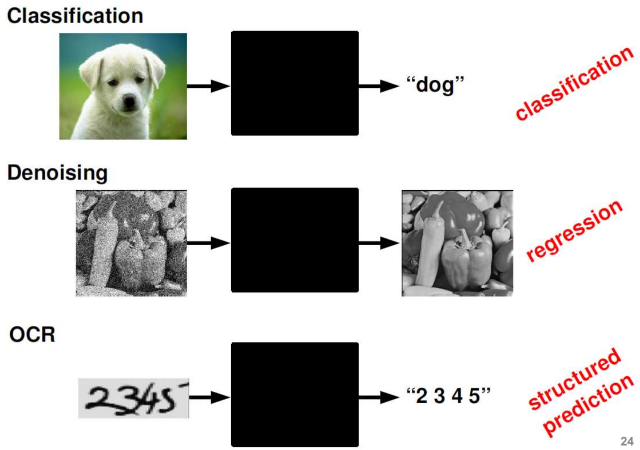
DISTINCT LEARNING PROBLEMS

Taxonomy of the Learning Settings

Goals and available data dictate the type of learning problem

- Supervised Learning
  - Classification
    - Binary
    - Multiclass
      - Nominal
      - Ordinal
  - Regression
  - Ranking
  - Counting
- Semi-supervised Learning
- Unsupervised Learning
- Reinforcement Learning
- etc.

Supervised Learning: Examples



Classification/Regression

$$y = f(\mathbf{x})$$

output **prediction function** **feature vector**

- **Training:** given a *training set* of labeled examples $\{(\mathbf{x}_1,y_1), \ldots, (\mathbf{x}_N,y_N)\}$, estimate the prediction function f by minimizing the prediction error on the training set
- **Testing:** apply f to a never before seen *test example* **x** and output the predicted value $y = f(\mathbf{x})$

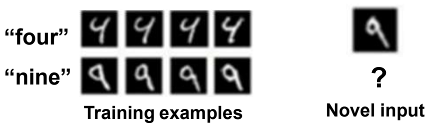
Regression

- Predicting house price
  - Output: price (a scalar)
  - Inputs: size, orientation, localization, distance to key services, etc.

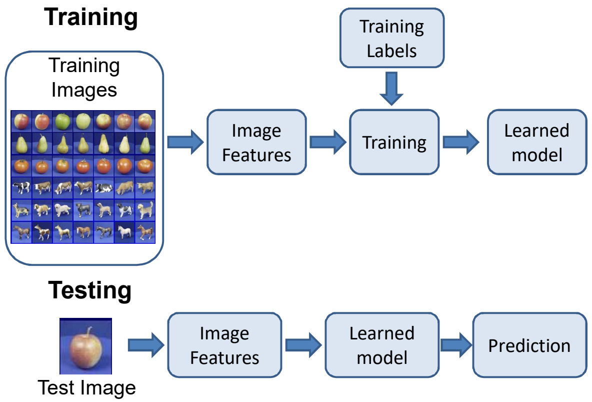
- Given a collection of labelled examples (= houses with known price), come up with a function that will predict the price of new examples (houses).

Classification

- Given a collection of *labelled* examples, come up with a function that will predict the labels of new examples.



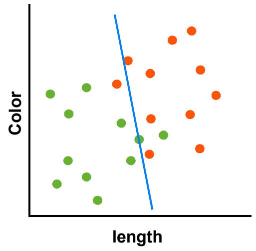
Classification in computer vision



FOR THE SAME PROBLEM,
DIFFERENT SOLUTIONS

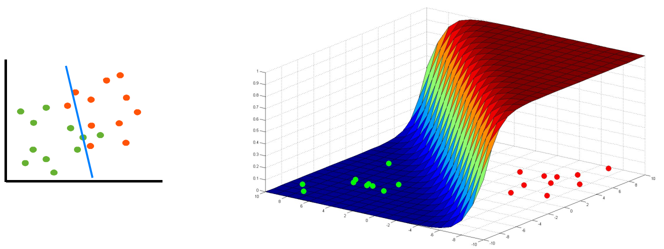
29

Design of a Classifier



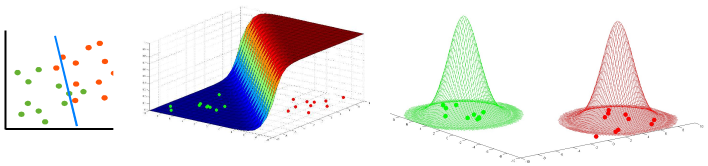
30

Design of a Classifier



31

Design of a Classifier



32

Taxonomy of the Learning Tools

Classifier

- no computation of posterior probabilities (probability of certain class given the data) → **Discriminant function**
- computation of posterior probabilities → **Probabilistic Discriminative Models**, **Probabilistic Generative Models**

Discriminant function

Properties
- directly map each x onto a class label

Tools
- Least Square Classification
- Fisher's Linear Discriminant
- SVM
- Etc.

Probabilistic Discriminative Models

Properties
- Model posterior probabilities ($p(C_k|x)$) directly

Tools
- Logistic Regression

Probabilistic Generative Models

Properties
- model class priors ($p(C_k)$) & class-conditional densities ($p(x|C_k)$)
- use to compute posterior probabilities ($C_k|x$)

Tools
- Bayes

Pros and Cons of the three approaches

- Discriminant Functions are the most simple and intuitive approach to classify data, but do not allow to
  - compensate for class priors (e.g. class 1 is a very rare disease)
  - minimize risk (e.g. classifying sick person as healthy more costly than classifying healthy person as sick)
  - implement reject option (e.g. person cannot be classified as sick or healthy with a sufficiently high probability)

Pros and Cons of the three approaches

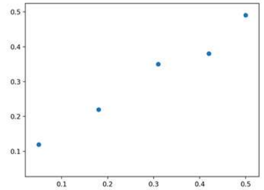
- Generative models provide a probabilistic model of *all* variables that allows to synthesize new data and to do novelty detection but
  - generating all this information is computationally expensive and complex and is not needed for a simple classification decision
- Discriminative models provide a probabilistic model for the target variable (classes) conditional on the observed variables
  - this is usually sufficient for making a well-informed classification decision without the disadvantages of the simple Discriminant Functions

DIFFERENT SOLUTIONS BUT WITH COMMON TRAITS

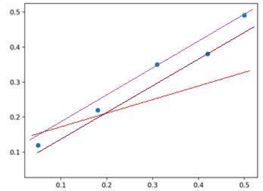
Common steps

- The learning of a model from the data entails:
  - **Representation**
  - **Evaluation**
  - **Optimization**

Linear Regression

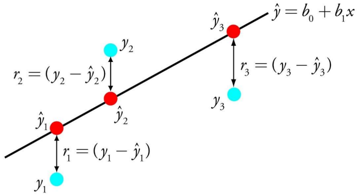


- Representation



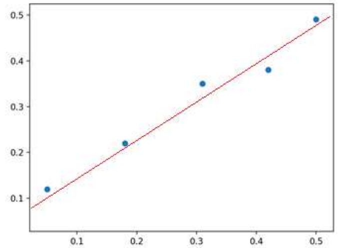
Linear Regression

- Quality



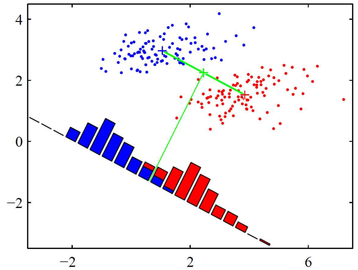
Linear Regression

- Optimization: finding the model that maximizes our measure of quality



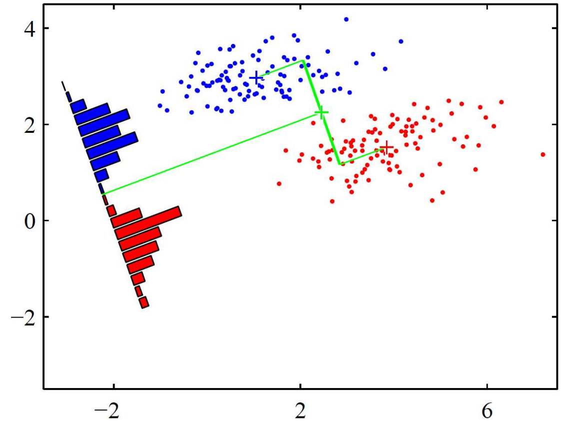
Let's design a classifier

- Use the (hyper-)plane orthogonal to the line joining the means
  - project the data in the direction given by the line joining the class means



41

Let's design a classifier



42

Fisher's linear discriminant

- Every algorithm has three components:
  - **Representation**
  - **Evaluation**
  - **Optimization**
- Representation: class of linear models
- Evaluation: find the direction **w** that maximizes J(**w**)= $\frac{(m_2-m_1)^2}{s_1^2+s_2^2}$ $\qquad J(\mathbf{w}) = \frac{\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{B}}\mathbf{w}}{\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{W}}\mathbf{w}}$
- Optimization

$$\mathbf{w} \propto \mathbf{S}_{\mathrm{W}}^{-1}(\mathbf{m}_2 - \mathbf{m}_1)$$

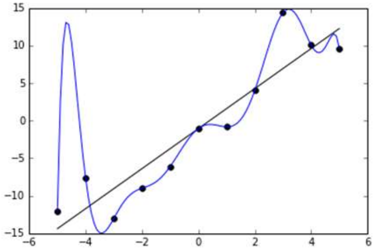
43

Hyper parameters / user defined parameters

AVOIDING OVERFITTING AND DATA MEMORIZATION

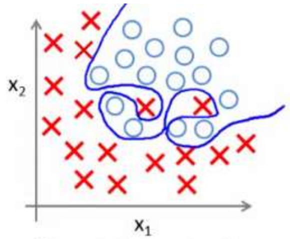
44

Regularized Regression



Regularized classifier

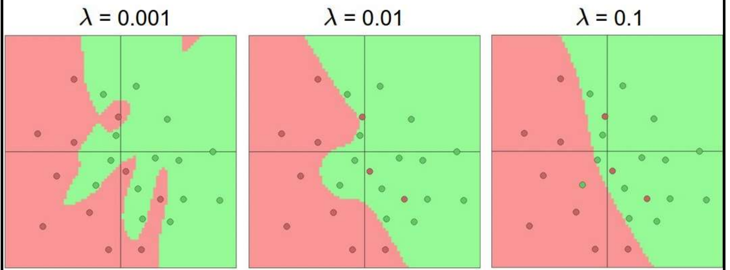
- Hyper parameters / user defined parameters



$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \theta_6 x_1^3 x_2 + \dots$$

Regularization

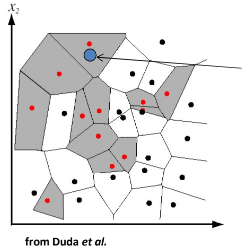
- Evaluation
  - Minimize (error in data) + λ (model complexity)



1-Nearest neighbour classifier

Assign label of nearest training data point to each test data point

Black = negative
Red = positive



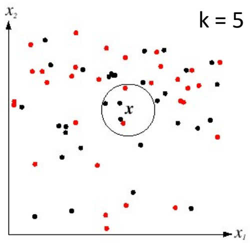
Novel test example

Closest to a positive example from the training set, so classify it as positive.

Voronoi partitioning of feature space for 2-category 2D data

k-Nearest neighbour classifier

- For a new point, find the k closest points from training data
- Labels of the k points "vote" to classify



If the query lands here, the 5 NN consist of 3 negatives and 2 positives, so we classify it as negative.

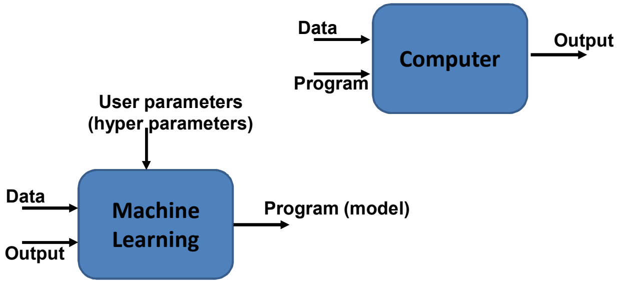
Black = negative
Red = positive

kNN as a classifier

- **Advantages**:
  - Simple to implement
  - Flexible to feature / distance choices
  - Naturally handles multi-class cases
  - Can do well in practice with enough representative data
- **Disadvantages:**
  - Large search problem to find nearest neighbors → Highly susceptible to the **curse of dimensionality**
  - Storage of data
  - Must have a meaningful distance function

What is Machine Learning?

- Automating the Automation



THERE ARE SO MANY OPTION TO DESIGN A CLASSIFIER…

A FAIR JUDGEMENT OF YOUR ALGORITHM

Model assessment, selection

- How to Compare Models?
- How can we select the right complexity model ?

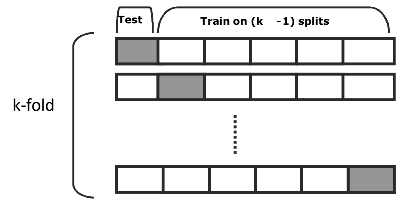
Training - general strategy

- We try to simulate the real world scenario.
- Test data is our future data. It should not be used in any design option of the classifier.
- Validation set can be our test set - we use it to select our model.
- The whole aim is to estimate the models' true error on the sample data we have.

Hold out / test set method

- It is simple, however
  - We waste some portion of the data
  - If we do not have much data, we may be lucky or unlucky with our test data
- With **cross-validation** we reuse the data

K-fold cross validation



In 3 fold cross validation, there are 3 runs.
In 5 fold cross validation, there are 5 runs.
In 10 fold cross validation, there are 10 runs.

the error is averaged over all runs

Evaluation

- Accuracy
- Precision and recall
- Squared error
- Likelihood
- Posterior probability
- Cost / Utility
- Margin
- Entropy
- K-L divergence
- Etc.

57

SOME CLASSICAL ML ALGORITHMS

58

Classical ML algorithms

- Top 10 algorithms in data mining (in 2007)
  - C4.5, k-Means, SVM, Apriori, EM, PageRank, AdaBoost, kNN, Naive Bayes, and CART

59

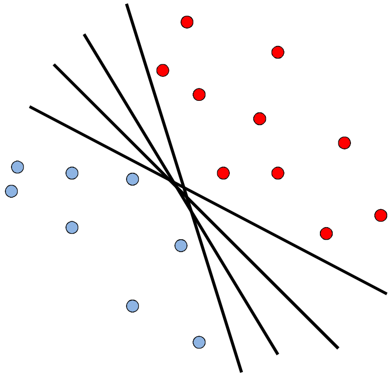
The Classics

SUPPORT VECTOR MACHINES

60

Linear classifiers

Find linear function to separate positive and negative examples



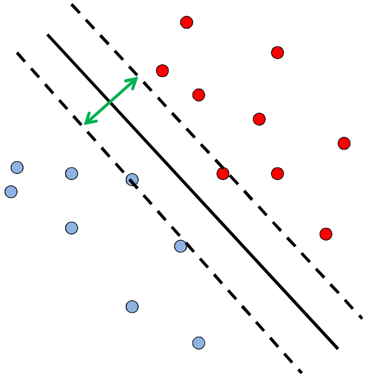
$\mathbf{x}_i$ positive: $\quad \mathbf{x}_i \cdot \mathbf{w} + b \geq 0$

$\mathbf{x}_i$ negative: $\quad \mathbf{x}_i \cdot \mathbf{w} + b < 0$

Which line is best?

61

Support Vector Machines

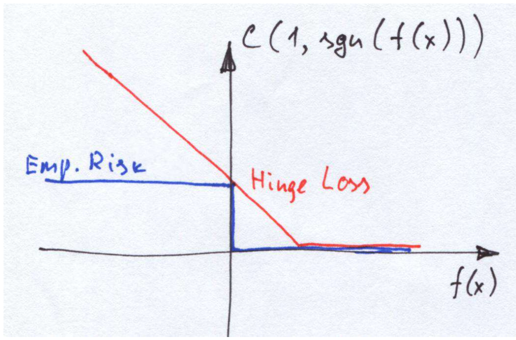


Classifier based on *optimal separating line (for 2D case)*

Maximize the ***margin*** between the positive and negative training examples

62

- Hinge Loss



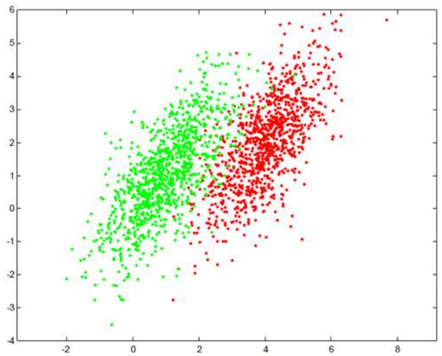
63

The Classics

GAUSSIAN MODELS

64

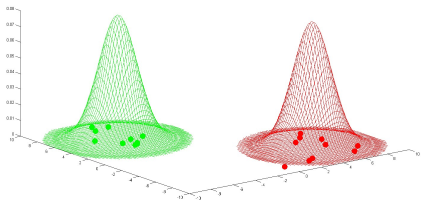
Let’s design a classifier



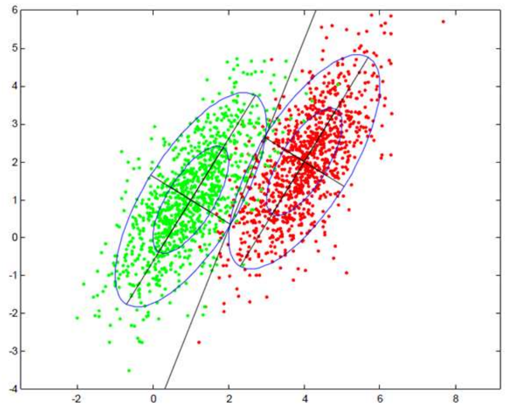
Pattern Recognition

- The learning of the model entails
  - **Representation:** Gaussian distribution for each class (maybe with shared co-variance)
  - **Evaluation:** maximum likelihood estimation (MLE) - find the parameters of the distribution that maximize the probability of the data
  - **Solve the optimization problem**

Gaussian Model



Gaussian Models



Bayes linear classifier

- Let us assume that the **class-conditional densities** are **Gaussian** and then explore the resulting form for the posterior probabilities.
- Assume that all classes share the same covariance matrix, thus the density for class C_k is given by

$$p(\mathbf{x} \mid C_k) = \frac{1}{(2\pi)^{D/2} |\Sigma|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T \Sigma^{-1} (\mathbf{x}-\boldsymbol{\mu}_k)^T}$$

- We then model the class-conditional densities p($\mathbf{x}$|C_k) and class priors p(C_k) and use these to compute **posterior probabilities** p(C_k|$\mathbf{x}$) through Bayes' theorem
- The **maximum likelihood** estimates of a Gaussian are

$$\hat{\boldsymbol{\mu}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i \text{ and } \hat{\Sigma} = \frac{1}{n}\sum_{i=1}^{n} (\mathbf{x}_i - \hat{\boldsymbol{\mu}})(\mathbf{x}_i - \hat{\boldsymbol{\mu}})^T$$

- Assuming only **2 classes** the **decision boundary is linear**

69

Making a decision

- How can we make a decision after observing the value of x?

$$\text{Decide} \begin{cases} C_1 & \text{if } P(C_1 \mid x) > P(C_2 \mid x) \\ C_2 & \textit{otherwise} \end{cases}$$

- Rewriting the rule gives

$$\text{Decide} \begin{cases} C_1 & \text{if } \dfrac{P(x \mid C_1)}{P(x \mid C_2)} > \dfrac{P(C_2)}{P(C_1)} \\ C_2 & \textit{otherwise} \end{cases}$$

- Bayes decision rule **minimizes** the error of this decision

70

Bayesian decision theory

- Bayesian decision theory gives the **optimal decision** rule under the assumption that the “true” values of the probabilities are **known**.
- But, how can we estimate (learn) the unknown p($\boldsymbol{x}$|C_j), j = 1, …, K ?

- **Parametric models**: assume that the form of the density functions is known
- **Non-parametric models**: no assumption about the form

71

Bayesian decision theory

- Parametric models
  – Density models (e.g., Gaussian)
  – Mixture models (e.g., mixture of Gaussians)
  – Hidden Markov Models
  – Bayesian Belief Networks

- Non-parametric models
  – Nearest neighbour estimation
  – Histogram-based estimation
  – Parzen window estimation

72

BEYOND THE CLASSICS

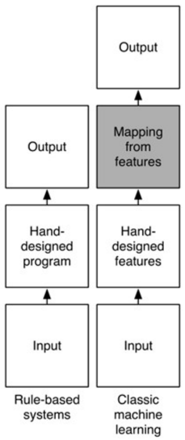
73

Limitations of the Classics

- Learning is disconnected from representation
  - It would be nice to bring learning to the beginning of the chain
- Almost only local constraints: global constraints are (almost) absent
  - Holistic structured representation
- Learning is not over time

74

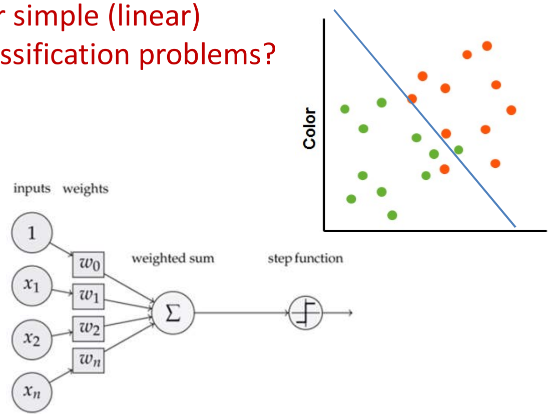
Classical Machine Learning



75

Do we need deep learning?

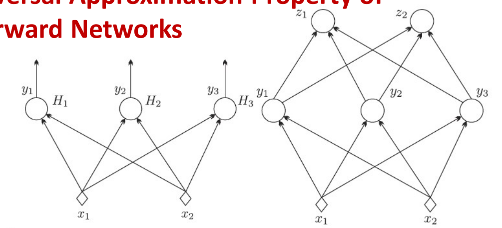
- For simple (linear) classification problems?

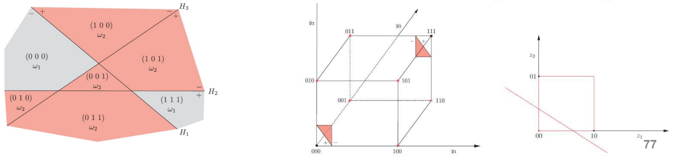


76

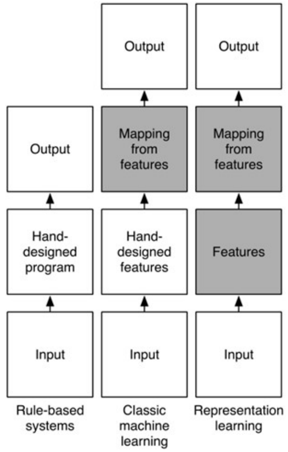
Do we need deep learning?

- For complex binary classification problems?
  - No: **Universal Approximation Property of Feed-forward Networks**



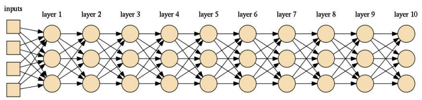


Representation Learning

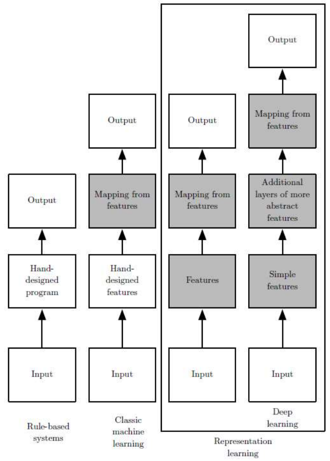


Do we need deep learning?

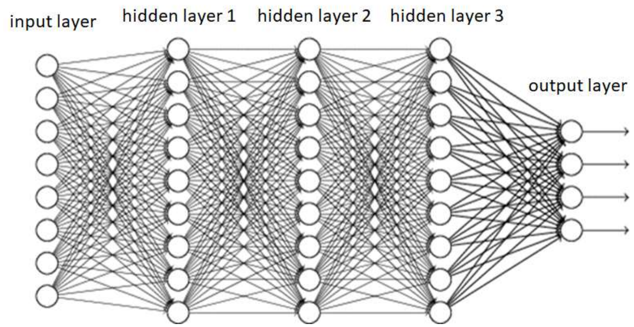
- For complex binary classification problems?
  Yes
  - In any learning task, we have to be concerned with what is feasibly "learnable" in a given representation.
  - Using networks with more layers, one can obtain more **compact representations** of the input-output relation.
    - We say that a network is **compact** if it consists of relatively few free parameters (few computational elements) to be learned/tuned during the training phase.
    - For a given number of training points, we expect compact representations to result in better generalization performance.
  - For complex tasks, where more complex concepts have to be learned, for example, recognition of a scene in a video recording, language and speech recognition, the underlying functional dependence is of a very complex nature so that we are unable to express it analytically in a simple way.



Representation Learning



Deep multilayer perceptron



81

References

- Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.
- Richard O. Duda, Peter E. Hart, David G. Stork, Pattern Classification, John Wiley & Sons, 2001
- Thomas Mitchell, Machine Learning, McGraw-Hill, 1997.
- P. Domingos, “A few useful things to know about machine learning,” CACM, 2012
- Andrew Moore, Support Vector Machines Tutorial, http://www.autonlab.org/tutorials/svm.html

82

References

- Selim Aksoy, Introduction to Pattern Recognition, Part I, http://retina.cs.bilkent.edu.tr/papers/patrec_tutorial1.pdf
- Ricardo Gutierrez-Osuna, Introduction to Pattern Recognition, http://research.cs.tamu.edu/prism/lectures/pr/pr_l1.pdf
- Pedro Domingos, Machine Learning, http://courses.cs.washington.edu/courses/cse446/14wi/
- Kristen Grauman, Discriminative classifiers for image recognition, http://www.cs.utexas.edu/~grauman/courses/spring2011/slides/lecture22_classifiers.pdf
- Victor Lavrenko and Nigel Goddard, Introductory Applied Machine Learning, http://www.inf.ed.ac.uk/teaching/courses/iaml/

83

References

- Recognizing and Learning Object Categories http://people.csail.mit.edu/torralba/shortCourseRLOC/index.html
- Using the Forest to See the Trees: A Graphical Model Relating Features, Objects, and Scenes, (K. Murphy, A. Torralba, W. Freeman), NIPS 2003
- Max-Margin Markov Networks , (B. taskar, C. Guestrin, D. Koller), NIPS 2004
- Large Margin Methods for Structured and Interdependent Output Variables, (I. Tsochantaridis, T. Joachims, T. Hofmann, Y. Altun), JMLR, vol 6, 2005
- Learning Spatial Context: Using Stuff to Find Things, (G. heitz, D. Koller), ECCV 2008, http://ai.stanford.edu/~gaheitz/Research/TAS/
- An Empirical Study of Context in Object Detection, (S. K. Divvala, D. Hoiem, J. H. Hays, A. A. Efros, M. Hebert), CVPR 2009 http://www.cs.cmu.edu/~santosh/projects/context.html
- Generative Models for Visual Objects and Object Recognition via Bayesian Inference, L. Fei-Fei, 2006
- Modeling Mutual Context of Object and Human Pose in Human-Object Interaction Activities, (B. Yao, L. Fei-Fei), CVPR 2010 http://videolectures.net/cvpr2010_fei_fei_mmco/
- No Hype, All Hallelujah: Structured Models in Computer Vision, (S. Nowozin), NIPS 2010

84

References

- Graphical Models for Time Series, (D. Barker, A. T. Cemgil), IEEE Signal Processing Magazine, vol 27, 2010
- Dynamic Graphical Models, (J. Bilmes), IEEE Signal Processing Magazine, vol 27, 2010
- A Martingale Framework for Detecting Changes in Data Streams by Testing Exchangeability, (S. Ho, H. Wechsler), TPAMI 2010
- Introduction to Statistical Relational Learning, (L. Getoor, B. Taskar), The MIT Press 2007
- Combining Video and Sequential Statistical Relational Techniques to Monitor Card Games, (L. Antanas, B. Gutmann, I. Thon, K. Kersting, L. De Raedt), ICML 2010
- Relational Learning for Collective Classification of Entities in Images, (A. Chechetka, D. Dash, M. Philipose), AAAI 2010
- Grouplet: A Structured Image Representation for Recognizing Human and Object Interactions, (B. Yao, L. Fei-Fei), CVPR 2010

Thank You for Your Attention!