Machine Learning
Classification

1 finding some options that optimaly seperats most of good from bads

Classification: a marketing example for telco

attributes

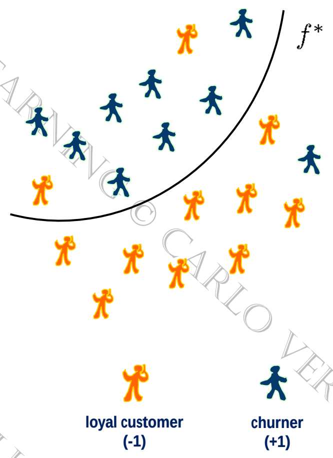
customers

Area	Pothers	Pmob	...	NumSMS	Class
2	0.14	0.59	...	18	1
3	0.26	0.35	...	9	-1
1	0.37	0.23	...	1	1
⋮	⋮	⋮	⋮	⋮	⋮
4	0.41	0.27	...	64	-1

past data

Area	Pothers	Pmob	...	NumSMS	Class
1	0.27	0.67	...	36	?
4	0.44	0.22	...	50	?
4	0.31	0.47	...	14	?
⋮	⋮	⋮	⋮	⋮	⋮
2	0.31	0.14	...	49	?

future data



2

Classification: prediction of HIV protease-cleavable peptides

attributes

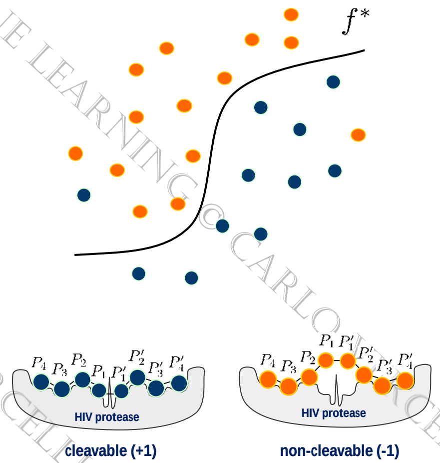
peptides

P_4	P_3	P_2	P_1	P_1'	P_2'	P_3'	P_4'	Class
T	Q	I	M	F	E	T	F	1
G	Q	V	N	Y	E	E	F	1
Y	T	F	L	F	M	N	S	-1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
D	T	V	L	E	E	M	S	1

past data

P_4	P_3	P_2	P_1	P_1'	P_2'	P_3'	P_4'	Class
S	Q	N	Y	P	I	V	Q	?
T	Q	I	M	F	E	T	F	?
G	L	A	A	P	Q	F	S	?
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
R	Q	A	G	F	L	G	L	?

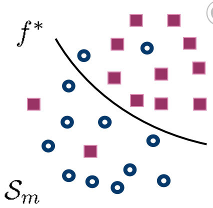
future data



3

Classification as a learning task

- $\mathcal{S}_m = \{(\mathbf{x}_i, y_i),\ i \in \mathcal{M}\}$ ing set, where $\mathbf{x}_i \in \Re^n$ $y_i \in \mathcal{D}$
- $\mathcal{H}$ denotes a set of functions $f(\mathbf{x}) : \Re^n \mapsto \mathcal{D}$
- Classification problem: define a hypotheses space and a $\mathcal{H}$ ıction $f^* \in \mathcal{H}$ optimally describes the relationship between $\mathbf{x}_i$ and y_i



$\Rightarrow$ $f^* \in \mathcal{H}$ $\Rightarrow$ $\overline{\mathbf{x}}$ $\Rightarrow$ $f(\overline{\mathbf{x}}) = -1$

learning — prediction

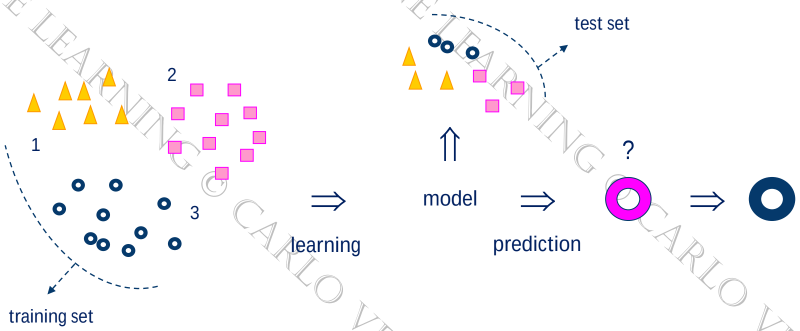
new patient $\Downarrow$

healthy !

- healthy patient (-1)
- ill patient (+1)

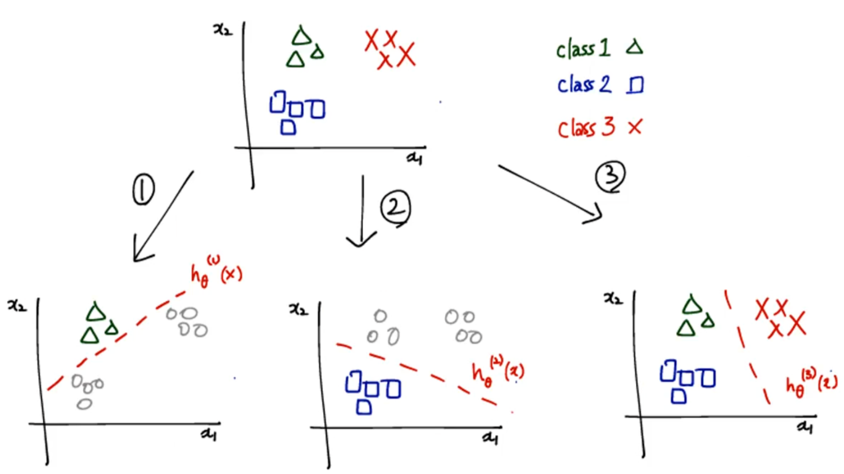
4

Classification models



Multi-class Approaches: 1-against-all

multi category ==> binary problems

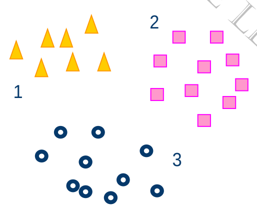


generage K category, three binary problems, we solve it and come back by majority voting to the main problem which was multi ...

Multi-class Approaches: 1-against-all vs. 1-against-1

arround robin (k number of categroy ,
take pairs of calssification
then based on majority

time consimuing!

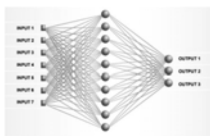


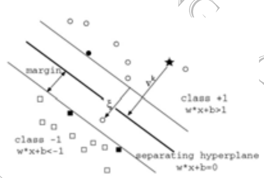
- Train K binary classifiers
  - Class 1 against {2,3,...K}
  - Class 2 against {1,3,...K}
  - ....
  - Class K against {1,2,...K-1}
- Predict for a new record by applying the K trained classifiers and assigning the label by majority voting

- Train K(K-1)/2 binary classifiers
  - Class 1 against {2}
  - Class 1 against {3}
  - ....
  - Class K-1 against {K}
- Predict for a new record by applying the K(K-1)/2 trained classifiers and assigning the label by majority voting

Taxonomy of classification methods

- Heuristic methods
  - classification trees
  - nearest neighbor
- Separation methods
  - perceptron
  - neural networks
  - support vector machines
- Regression methods
  - logistic regression
- Probabilistic methods
  - bayesian methods





$$y_{MAP} = \arg\max_{y \in \mathcal{H}} P(y|\mathbf{x}) = \arg\max_{y \in \mathcal{H}} \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})}$$

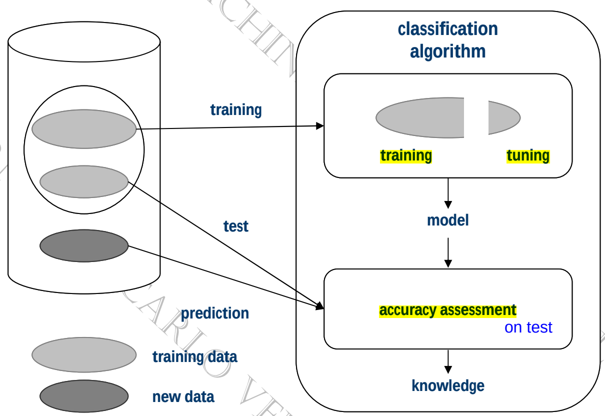
criteria for evaluting / how to compare

Assessing classification methods

- Prediction accuracy several metrics in next slide
- Speed
  - time required to train a model
- Robustness
  - handling noise and missing data
- Scalability
  - effectiveness in handling large datasets
- Interpretability
  - value of the knowledge extracted from the model
- Rules effectiveness
  - number of rules
  - conciseness and significance of the rules

9

Development of a classification model



Source: C. Vercellis, Business Intelligence. Data Mining and Optimization for Decision Making, Wiley, 2009.

10

Prediction accuracy

- **Loss function** **counting the number of misclassifications**

$$L(y, f(\mathbf{x})) = \begin{cases} 0 & \text{if } y = f(\mathbf{x}) \\ 1 & \text{if } y \neq f(\mathbf{x}) \end{cases}$$

- **Empirical error**

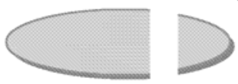
$$R_{\text{emp}} = \frac{1}{m}\sum_{i=1}^{m} L(f(\mathbf{x}_i))$$

average number of miscalssifications

11

Prediction accuracy

- **Prediction accuracy can be estimated through four main approaches:**
  - **holdout method:**



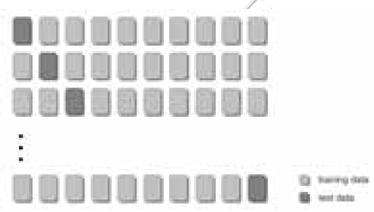
training test

  - **repeated random sampling method: repeated applications of holdout method**

12

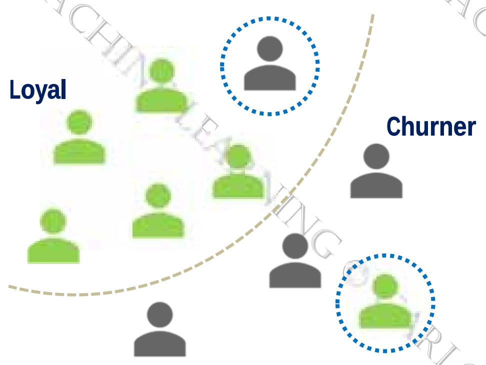
Prediction accuracy

- **k-fold cross-validation:** systematic approach, 99% of times is used



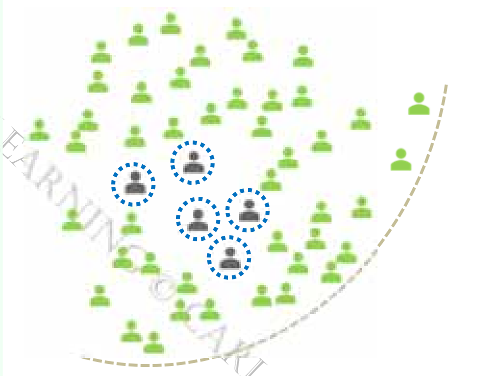
- **leave-one-out method: k-fold with k=m**
  best approximation of accuracy

Model evaluation



Accuracy = % accurate predictions= 80%
Error = 100 – Accuracy = 20%

when the data is imbalanced, the probabality is not 50 50 and it might:



Accuracy = 90%
Error = 10%

Confusion matrix contingency tables

- p: true negative
- q: false positive
- u: false negative
- v: true positive

		predictions		
		−1 (negative)	+1 (positive)	total
examples	−1 (negative)	p	q	$p+q$
	+1 (positive)	u	v	$u+v$
	total	$p+u$	$q+v$	m

Accuracy = % records correctly classified= (p + v)/m

Recall (true positive rate tp) = % true positives correctly classified = v /(u + v)

Precision (prc) = % true positives among those predicted positives = v /(q + v)

False positive rate = q/(p+q)

True negative rate = p/(p+q)

False negative rate = u/(u+v)

F-measure $$F = \frac{(\beta^2 + 1)\,\mathrm{tp} \times \mathrm{prc}}{\beta^2\,\mathrm{prc} + \mathrm{tp}},$$

Geometric mean $$\mathrm{gm} = \sqrt{\mathrm{tp} \times \mathrm{prc}},$$

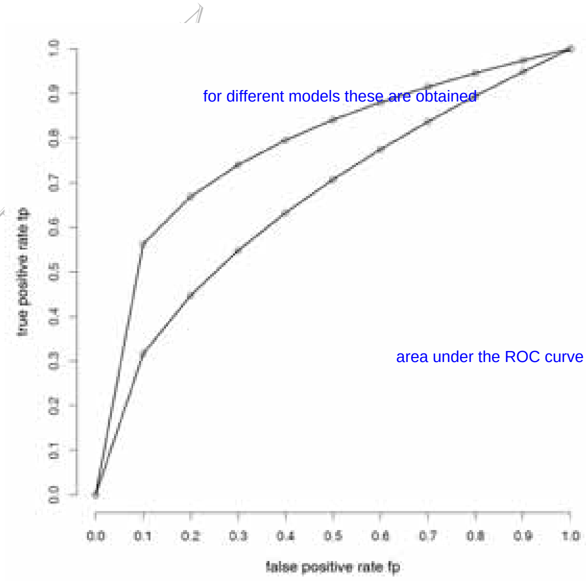
بتا رو ما فرض میگیریم. فکر کن وقتی هیچ سنسی نسبت به کاست نداریم میگیریم ۱
بتا رو به صورت رفلکت ریشوی بین کاستشون پیدا میکنیم. خیلی دقیق نگفت چطوری

15

ROC curves recieving operating carachtersitics

the top one is higher and better

for different models these are obtained



area under the ROC curve => the greater the better

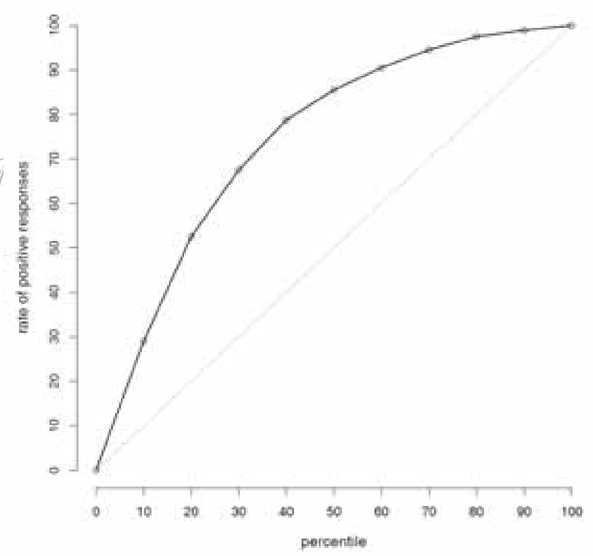
16

هیچ مدلی نیست که توی همه متریک ها بهتر باشه ... برای همین ما باید انتخاب کنیم.

Cumulative gain

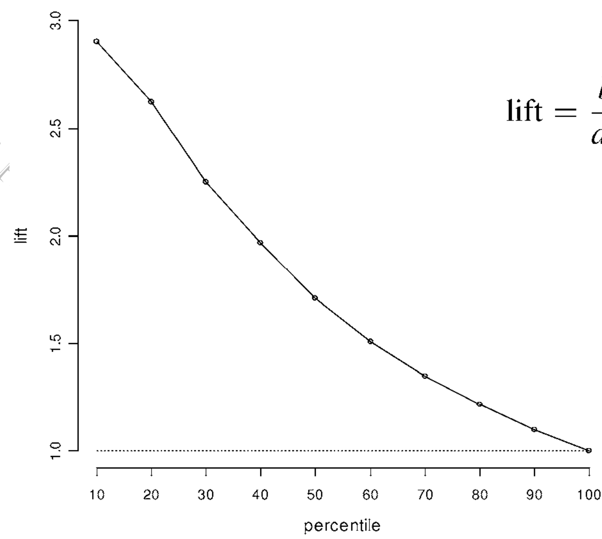
... خیلی نفهمیدم دقیقا چه شکلی تحلیل میشه

رندوم یعنی ۵۰ ۵۰ پس خط چین ینی این ... شبیه به انداختن کوین



17

Lift



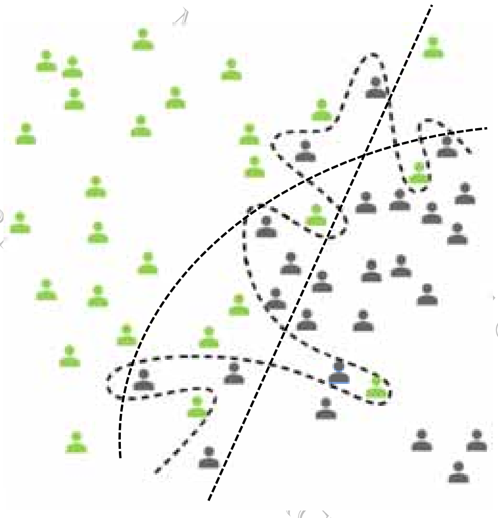
$$\text{lift} = \frac{b/s}{a/m}$$

بین نقطه و خیچین ریشو گرفت فک کنم

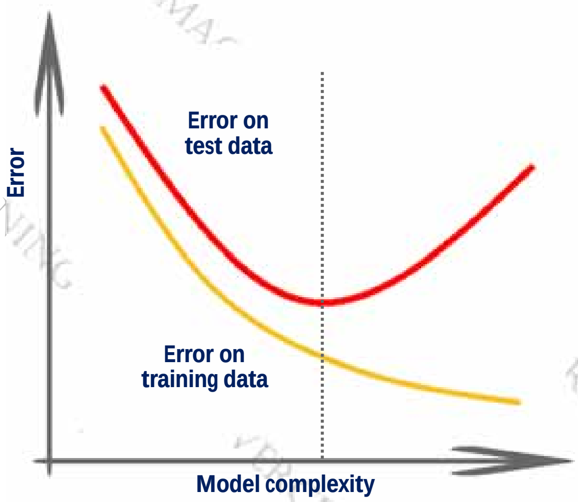
18

if accuracy of training set is larger than test set ==>>>> overfitting Alert

Overfitting



Bias-variance trade-off



Numerical representation of categorical variables

we need numerical expect for classification trees and bayesian methods

- **For example, to represent the month associated to an observation**

Other Columns	Month	D_Jan	D_Feb	D_Mar	D_Apr	D_May	D_Jun	D_Jul	D_Aug	D_Sep	D_Oct	D_Nov	D_Dec
....	Jan	**1**	0	0	0	0	0	0	0	0	0	0	0
....	Mar	0	0	**1**	0	0	0	0	0	0	0	0	0
....	Jun	0	0	0	0	0	**1**	0	0	0	0	0	0
....	Dec	0	0	0	0	0	0	0	0	0	0	0	**1**
....	Nov	0	0	0	0	0	0	0	0	0	0	**1**	0
....	Jan	**1**	0	0	0	0	0	0	0	0	0	0	0
....	...	...	...	...	...	...	...	...	...	...	...	...	...
....	...	...	...	...	...	...	...	...	...	...	...	...	...
....	...	...	...	...	...	...	...	...	...	...	...	...	...
....	...	...	...	...	...	...	...	...	...	...	...	...	...
....	...	...	...	...	...	...	...	...	...	...	...	...	...
....	...	...	...	...	...	...	...	...	...	...	...	...	...
....	Feb	0	**1**	0	0	0	0	0	0	0	0	0	0

Numerical representation of categorical variables

- **variable X_j assumes values in the set** $V = \{v_1, v_2, \ldots, v_H\}$

- ***H*-1 dummy variables**

$$D_{j1}, D_{j2}, \ldots, D_{j,H-1}$$

Predicting survival in patients after myocardial infarction: introduction

All the patients in the dataset suffered heart attacks at some point in the past. Some are still alive and some are not. The problem addressed by past researchers was to predict from the other variables whether or not the patient will survive at least one year.

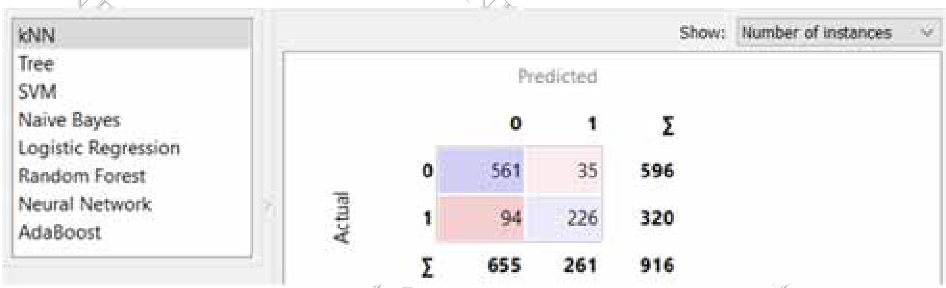
Myocardial infarction (MI) or **acute myocardial infarction (AMI)** occurs when blood flow stops to a part of the heart causing damage to the heart muscle.
Most MIs occur due to coronary artery disease. Risk factors include high blood pressure, smoking, diabetes, lack of exercise, obesity, high blood cholesterol, poor diet, and excessive alcohol intake, among others. Blockages on the **left side** of your heart are usually **more dangerous.**
Binary target Survival is explained by Social Security Number, Nationality, Area, Shortening fraction, Atrial dimension of the LV, Wall motion index, Hematocrit, Pericardial effusion.

Cross-validation for different methods on Heart dataset

Evaluation Results are under the curve

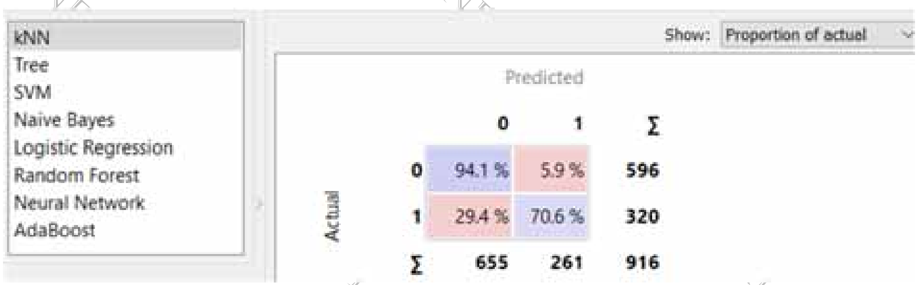
Model	AUC	CA	F1	Precision	Recall
kNN	0.928	0.859	0.855	0.860	0.859
Tree	0.941	0.936	0.935	0.936	0.936
SVM	0.892	0.866	0.863	0.866	0.866
Random Forest	0.992	0.961	0.961	0.961	0.961
Neural Network	0.922	0.841	0.839	0.839	0.841
Naive Bayes	0.890	0.796	0.797	0.798	0.796
Logistic Regression	0.922	0.846	0.845	0.845	0.846
AdaBoost	0.965	0.971	0.970	0.970	0.971

Confusion matrix for different methods on Heart dataset



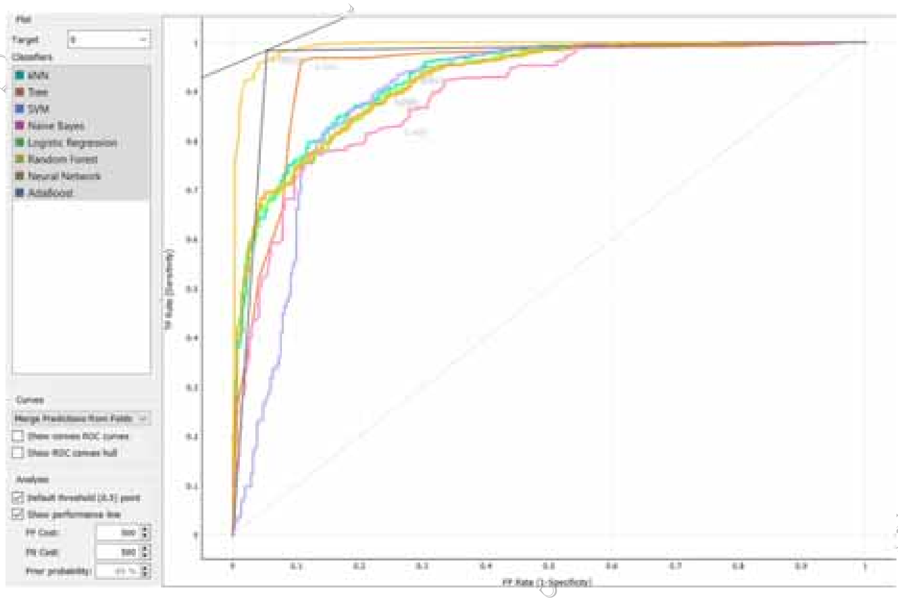
29

Confusion matrix for different methods on Heart dataset



30

ROC curve for different methods on Heart dataset



31

1st algo. no training phase, prediciting based on distances

K-nearest neighbour (K-NN)

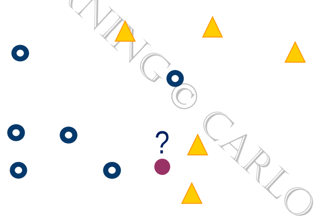
To a new record is attributed the class to which belong the majority of records out of the K closest ones (*majority voting*)

فاصله با هر کدوم رو بررسی میکنه و نزدیک ترین رو به عنوان لیبل انتخاب میکنه.

The choice of K affects model robustness and accuracy

By increasing K ...

به صورت تجربی انتخاب میکنیم.



less sensitive to outliers

less distinction among classes

هایپر پارامتر رو با اکسپریمنت انتخاب میکنیم.

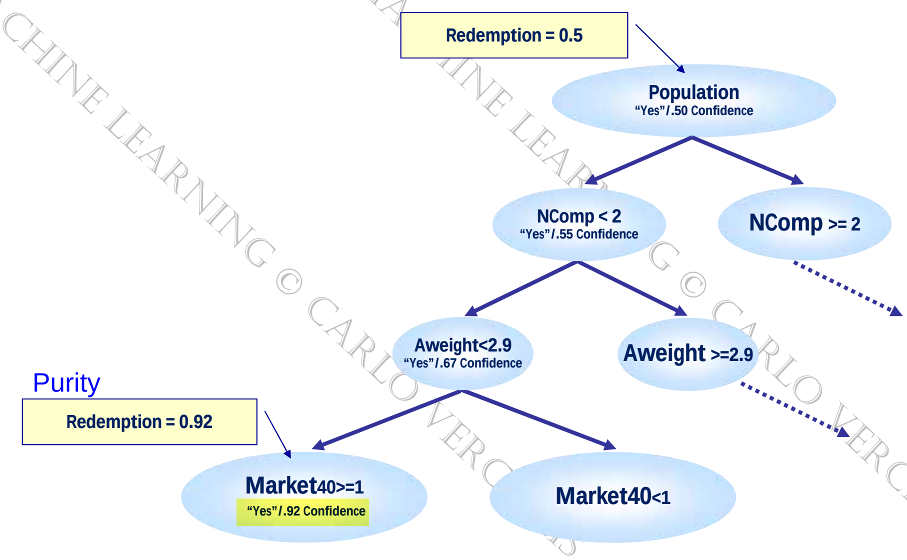
32

imbalanced => ۵۰ ۵۰
نیست کلاسا

Session Oct 16th

Business case automotive – customer acquisition

Purity



بیشترین فاصله رو بده.
it's linear

hyper params:
- purity treshhold (90%)
- minumim cardinality - minimum number in a leaf to split
- information gain - difference of hetrogenieity reduction.

Classification trees

- Base recursive greedy algorithm
  - All records are placed in the root node
  - Records in each node are split according to a heuristic test based on the value of one or more attributes
- To prevent overfitting
  - pre pruning by stopping criteria
    - purity threshold
    - minimum cardinality
    - minimum information gain
  - post pruning: identifies and removes ramifications affected by noise
- Majority voting is used to classify a leaf

LaMasa_BL27_BL...: An iterative process in which some branches are eliminated from the tree. Again, the idea is if it is better without on the internal test set, the tuning set, then it means the that branching the eliminated branching were.
LaMasa_BL27_BL...: Were producing overfitting. Okay, it's pretty reasonable, pretty rational.

Classification trees

- **Classification trees are distinguished in:**
  - binary trees
  - general trees

and in

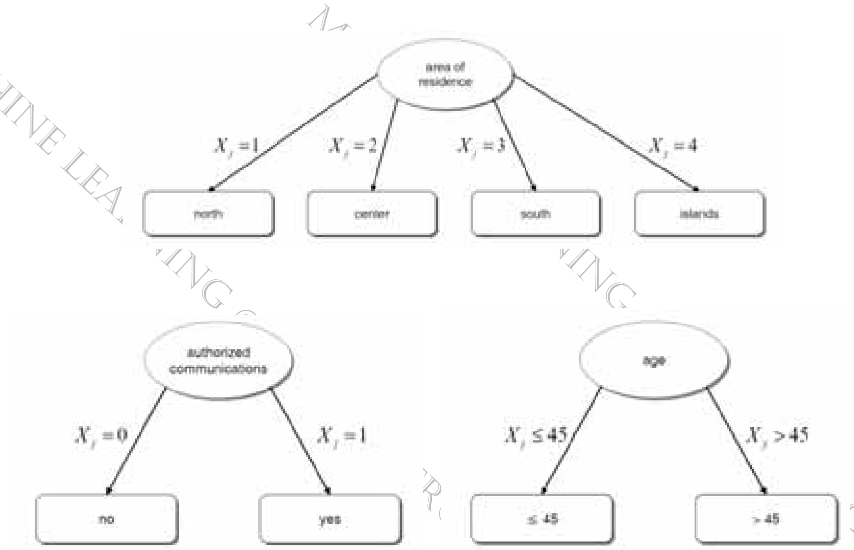
- **univariate trees (or axis parallel)**

$$\Rightarrow \quad X_j \leq b$$

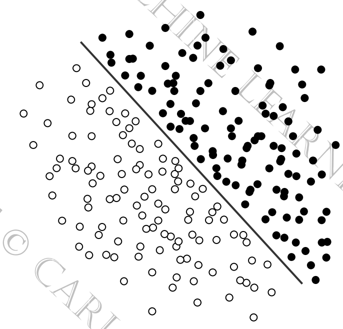
- **multivariate trees (or oblique)**

$$\Rightarrow \quad \sum_{j=1}^{n} w_j x_j \leq b$$

Classification trees



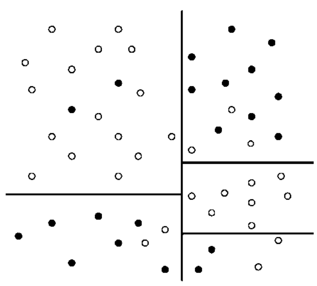
Classification trees



single univariate rule

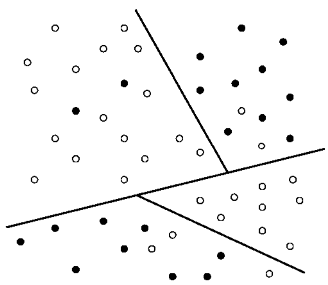
37

Classification trees



classification by an
axis parallel tree

interperatability اینجا داریم.



classification by an
oblique tree

38

به نظر خوب فیت نمیشن ولی به
خاطر اینترنپتیلیتی باید بریم
سراغشون یا شاید حتی مسائل
حقوقی مارو مجبور کنن.

گوش ندادم چرا ببین دوباره

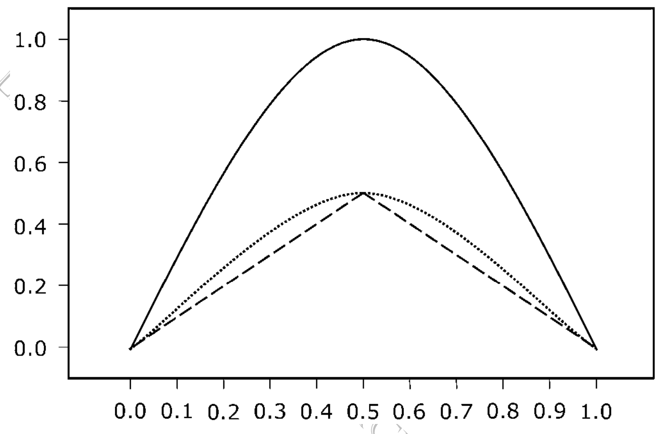
Classification trees

- In axis parallel trees the best separation rule identifies the most effective attribute and split
- Main criteria for splitting
  - misclassification index $$\mathrm{Miscl}(q) = 1 - \max_h p_h$$
  - entropy index $$\mathrm{Entropy}(q) = -\sum_{h=1}^{H} p_h \log_2 p_h$$
  - Gini index $$\mathrm{Gini}(q) = 1 - \sum_{h=1}^{H} p_h^2$$

P_1 negative
P_2 positive
node q

sum is total number of obs in this note.

Classification trees

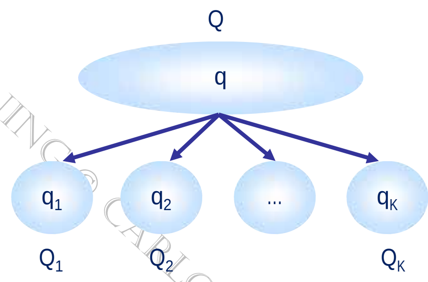


Classification trees

- **Impurity of a splitting rule**

$$I(q_1, q_2, \ldots, q_K) = \sum_{k=1}^{K} \frac{Q_k}{Q} I(q_k).$$

- **At each node select the rule minimizing the impurity or, equivalently, maximizing the information gain**



$$\Delta(q, q_1, q_2, \ldots, q_K) = I(q) - I(q_1, q_2, \ldots, q_K)$$
$$= I(q) - \sum_{k=1}^{K} \frac{Q_k}{Q} I(q_k).$$

41

Classification trees

area	numin	timein	numout	Pothers	Pmob	Pland	numsms	numserv	numcall	diropt	churner
3	32	8093	45	0.14	0.75	0.12	18	1	0	0	0
3	277	157842	450	0.26	0.35	0.38	9	3	0	1	0
1	17	15023	20	0.37	0.23	0.40	1	1	0	0	0
1	46	22459	69	0.10	0.39	0.51	33	1	0	0	0
1	19	8640	9	0.00	0.00	1.00	0	0	0	0	0
2	17	7652	66	0.16	0.42	0.43	1	3	0	1	0
3	47	17768	11	0.45	0.00	0.55	0	0	0	0	0
3	19	9492	42	0.18	0.34	0.48	3	1	0	0	1
1	1	84	9	0.09	0.54	0.37	0	0	0	0	1
2	119	87605	126	0.84	0.02	0.14	12	1	0	0	0
4	24	6902	47	0.25	0.26	0.48	4	1	0	0	0
1	32	28072	43	0.28	0.66	0.06	0	1	0	0	0
3	103	112120	24	0.61	0.28	0.11	24	2	0	0	0
3	45	21921	94	0.34	0.47	0.19	45	2	0	1	0
1	8	25117	89	0.02	0.39	0.09	189				
3	4	945	16	0.00	0.00	1.00	0				
2	83	44263	83	0.00	0.00	0.67	0				
2	22	15979	59	0.05	0.53	0.41	5				
2	0	0	57	0.00	1.00	0.00	15				
4	162	114108	273	0.18	0.15	0.41	2				
4	21	4141	70	0.14	0.58	0.28	0				
4	33	10066	45	0.12	0.21	0.67	0				
4	5	965	40	0.41	0.27	0.32	64				

Table 5.3 Meaning of the attributes in Table 5.2

attribute	meaning
area	residence area
numin	number of calls received in period $t-2$
timein	duration in seconds of calls received in period $t-2$
numout	number of calls placed in the period $t-2$
Pothers	percentage of calls placed to other mobile telephone companies in period $t-2$
Pmob	percentage of calls placed to the same mobile telephone company in period $t-2$
Pland	percentage of calls placed to land numbers in period $t-2$
numsms	number of messages sent in period $t-2$
numserv	number of calls placed to special services in period $t-2$
numcall	number of calls placed to the call center in period $t-2$
diropt	binary variable indicating whether the customer corresponding to the record has subscribed to a special rate plan for calls placed to selected numbers
churner	binary variable indicating whether the customer corresponding to the record has left the service in period t

42

Classification trees

attribute	class 1	class 2	class 3	class 4
numin	[0, 20)	[20, 40)	[40, ∞)	
timein	[0, 10 000)	[10 000, 20 000)	[20 000, 30 000)	[30 000, ∞)
numout	[0, 30)	[30, 60)	[60, 90)	[90, ∞)
Pothers	[0, 0.1)	[0.1, 0.2)	[0.2, 0.3)	[0.3, ∞)
Pmob	[0, 0.2)	[0.2, 0.4)	[0.4, 0.6)	[0.6, ∞)
Pland	[0, 0.25)	[0.25, 0.5)	[0.5, ∞)	
numsms	[0, 1)	[1, 10)	[10, 20)	[20, ∞)
numserv	[0, 1)	[1, 2)	[2, 3)	[3, ∞)
numcall	[0, 1)	[1, 3)	[3, ∞)	

43

Classification trees

area	numin	timein	numout	Pothers	Pmob	Pland	numsms	numserv	numcall	diropt	churner
2	1	1	2	1	4	1	3	2	2	0	1
1	1	3	3	2	4	1	4	2	3	0	0
3	2	1	2	2	4	1	3	2	1	0	0
1	2	3	2	3	4	1	1	2	1	0	0
2	3	4	4	4	1	1	3	2	1	0	0
3	3	4	1	4	2	1	4	3	1	0	0
3	3	3	4	4	3	1	4	3	1	1	0
1	1	1	1	1	3	2	1	1	1	0	1
2	2	2	2	1	3	2	2	3	1	1	1
4	2	1	3	2	3	2	1	2	1	1	1
3	1	1	2	2	2	2	2	2	1	0	1
4	3	4	4	2	1	2	2	4	1	1	1
2	1	1	3	2	3	2	2	4	1	1	0
4	2	1	2	3	2	2	2	2	1	0	0
3	3	4	4	3	2	2	2	4	1	1	0
1	1	2	1	4	2	2	2	2	1	0	0
4	1	1	2	4	2	2	4	2	1	0	1
1	1	1	1	1	1	3	1	1	1	0	0
3	1	1	1	1	1	3	1	1	1	0	1
2	3	4	3	1	1	3	1	1	1	0	1
1	3	3	3	1	2	3	4	2	1	0	0
4	2	2	2	2	2	3	1	1	1	0	1
3	3	2	1	4	1	3	1	1	1	0	0

44

Classification trees

$$I_E(q) = \text{Entropy}(q) = -\frac{13}{23}\log_2\frac{13}{23} - \frac{10}{23}\log_2\frac{10}{23} = 0.988$$

$$p_0(q_1) = 5/6, \quad p_0(q_2) = 2/5, \quad p_0(q_3) = 5/7, \quad p_0(q_4) = 1/5,$$
$$p_1(q_1) = 1/6, \quad p_1(q_2) = 3/5, \quad p_1(q_3) = 2/7, \quad p_1(q_4) = 4/5.$$

$$\begin{aligned} I_E(q_1, q_2, q_3, q_4) &= \frac{6}{23}I_E(q_1) + \frac{5}{23}I_E(q_2) + \frac{7}{23}I_E(q_3) + \frac{5}{23}I_E(q_4) \\ &= \frac{6}{23}0.650 + \frac{5}{23}0.971 + \frac{7}{23}0.863 + \frac{5}{23}0.722 = 0.8. \end{aligned}$$

$$\begin{aligned} \Delta_E(area) = \Delta_E(q, q_1, q_2, q_3, q_4) &= I_E(q) - I_E(q_1, q_2, q_3, q_4) \\ &= 0.988 - 0.8 = 0.188. \end{aligned}$$

$$\begin{aligned} \Delta_E(numin) &= 0.057, & \Delta_E(Pland) &= 0.125, \\ \Delta_E(timein) &= 0.181, & \Delta_E(numsms) &= 0.080, \\ \Delta_E(numout) &= 0.065, & \Delta_E(numserv) &= 0.057, \\ \Delta_E(Pothers) &= 0.256, & \Delta_E(numcall) &= 0.089, \\ \Delta_E(Pmob) &= 0.043, & \Delta_E(diropt) &= 0.005. \end{aligned}$$

Classification trees

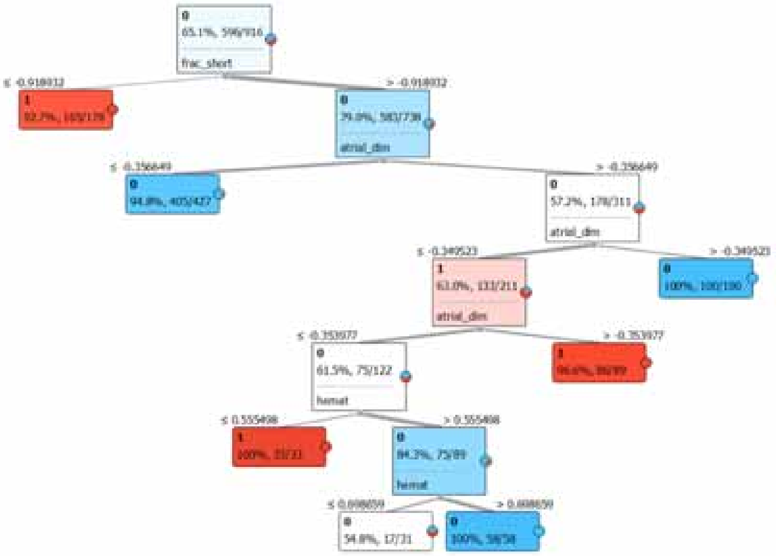
$$I_G(q) = \text{Gini}(q) = 1 - \left(\frac{13}{23}\right)^2 - \left(\frac{10}{23}\right)^2 = 0.491$$

$$\begin{aligned} I_G(q_1, q_2, q_3, q_4) &= \frac{6}{23}I_G(q_1) + \frac{5}{23}I_G(q_2) + \frac{7}{23}I_G(q_3) + \frac{5}{23}I_G(q_4) \\ &= \frac{6}{23}0.278 + \frac{5}{23}0.638 + \frac{7}{23}0.194 + \frac{5}{23}0.528 = 0.370. \end{aligned}$$

$$\begin{aligned} \Delta_G(area) = \Delta_G(q, q_1, q_2, q_3, q_4) &= I_G(q) - I_G(q_1, q_2, q_3, q_4) \\ &= 0.491 - 0.370 = 0.121. \end{aligned}$$

$$\begin{aligned} \Delta_G(numin) &= 0.037, & \Delta_G(Pland) &= 0.078, \\ \Delta_G(timein) &= 0.091, & \Delta_G(numsms) &= 0.052, \\ \Delta_G(numout) &= 0.043, & \Delta_G(numserv) &= 0.038, \\ \Delta_G(Pothers) &= 0.146, & \Delta_G(numcall) &= 0.044, \\ \Delta_G(Pmob) &= 0.028, & \Delta_G(diropt) &= 0.003. \end{aligned}$$

Classification tree for the Heart dataset



Bayesian methods

- compute posterior probability:

$$P(y|\mathbf{x})$$

- estimate of prior probability

$$P(y) = P(y = v_h) = \frac{m_h}{m}$$

- compute posterior using Bayes theorem

$$P(y|\mathbf{x}) = \frac{P(\mathbf{x}|y)P(y)}{\sum_{l=1}^{H} P(\mathbf{x}|y)P(y)} = \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})}$$

Bayesian methods

- To classify new observations, use *maximum a posteriori hypothesis* **(MAP) principle:**

$$y_{MAP} = \arg\max_{y \in \mathcal{H}} P(y|\mathbf{x}) = \arg\max_{y \in \mathcal{H}} \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})}$$

- Maximize numerator

$$P(\mathbf{x}|y = v_h)P(y = v_h) \geq P(\mathbf{x}|y = v_l)P(y = v_l), \quad l = 1, 2, \ldots, H, \; l \neq h$$

...but sample estimation of $P(\mathbf{x}|y)$ cannot be hardly done in practice

49

Naïve Bayesian methods

- assume attributes are independent:

$$P(\mathbf{x}|y) = P(x_1|y) \times P(x_2|y) \times \cdots \times P(x_n|y) = \prod_{j=1}^{n} P(x_j|y)$$

- two possible cases:
  - categorical attributes:

$$P(x_j|y) = P(x_j = r_{jk}|y = v_h) = \frac{s_{jhk}}{m_h}$$

  - numerical attributes: postulate a given distribution, such as

$$P(x_j|y = v_h) = \frac{1}{\sqrt{2\pi\sigma_{jh}}} e^{-\frac{(x_j - \mu_{jh})^2}{2\sigma_{jh}^2}}$$

50

Naive Bayesian methods

area

$$P(area = 1 | churner = 0) = \frac{5}{13}, \quad P(area = 1 | churner = 1) = \frac{1}{10},$$
$$P(area = 2 | churner = 0) = \frac{2}{13}, \quad P(area = 2 | churner = 1) = \frac{3}{10},$$
$$P(area = 3 | churner = 0) = \frac{5}{13}, \quad P(area = 3 | churner = 1) = \frac{2}{10},$$
$$P(area = 4 | churner = 0) = \frac{1}{13}, \quad P(area = 4 | churner = 1) = \frac{4}{10}.$$

Pothers

$$P(Pothers = 1 | churner = 0) = \frac{2}{13}, \quad P(Pothers = 1 | churner = 1) = \frac{5}{10},$$
$$P(Pothers = 2 | churner = 0) = \frac{3}{13}, \quad P(Pothers = 2 | churner = 1) = \frac{4}{10},$$
$$P(Pothers = 3 | churner = 0) = \frac{3}{13}, \quad P(Pothers = 3 | churner = 1) = 0,$$
$$P(Pothers = 4 | churner = 0) = \frac{5}{13}, \quad P(Pothers = 4 | churner = 1) = \frac{1}{10}.$$

Naive Bayesian methods

...and relative frequencies of the two classes

$$P(churner = 0) = \frac{13}{23} = 0.56, \quad P(churner = 1) = \frac{10}{23} = 0.44.$$

Predict the target class for a new observation

$$\mathbf{x} = (1, 1, 1, 2, 1, 4, 2, 1, 2, 1, 0)$$

Naive Bayesian methods

Compute probabilities

$$P(\mathbf{x}|0) = \frac{5}{13}\cdot\frac{4}{13}\cdot\frac{4}{13}\cdot\frac{3}{13}\cdot\frac{2}{13}\cdot\frac{3}{13}\cdot\frac{4}{13}\cdot\frac{3}{13}\cdot\frac{7}{13}\cdot\frac{12}{13}\cdot\frac{10}{13} = 0.81\cdot 10^{-5}$$
$$P(\mathbf{x}|1) = \frac{1}{10}\cdot\frac{5}{10}\cdot\frac{6}{10}\cdot\frac{5}{10}\cdot\frac{5}{10}\cdot\frac{1}{10}\cdot\frac{6}{10}\cdot\frac{5}{10}\cdot\frac{4}{10}\cdot\frac{9}{10}\cdot\frac{7}{10} = 5.67\cdot 10^{-5}$$

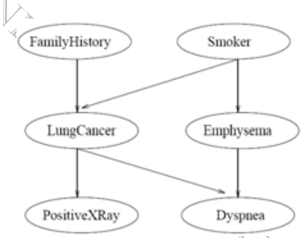
obtaining the posterior probability

$$\begin{aligned} P(churner = 0|\mathbf{x}) &= P(\mathbf{x}|0)P(churner = 0) \\ &= 0.81\cdot 10^{-5}\cdot 0.56 = 0.46\cdot 10^{-5}, \\ P(churner = 1|\mathbf{x}) &= P(\mathbf{x}|1)P(churner = 1) \\ &= 5.67\cdot 10^{-5}\cdot 0.44 = 2.495\cdot 10^{-5}. \end{aligned}$$

53

Bayesian belief networks

BAYESIAN BELIEF NETWORKS allow to define dependency relationships between (small) subsets of attributes



Acyclic oriented graph

Conditional probability table

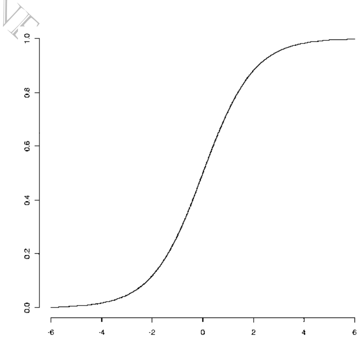
	FH,S	FH,~S	~FH,S	~FH, ~S
LC	0.8	0.5	0.7	0.1
~LC	0.2	0.5	0.3	0.9

54

Logistic regression

- A logistic function (sigmoid)

$$S(t) = \frac{1}{1+e^{-t}}$$



Postulate posterior probability

$$P(y=0|\mathbf{x}) = \frac{1}{1+e^{\mathbf{w}'\mathbf{x}}}$$

$$P(y=1|\mathbf{x}) = \frac{e^{\mathbf{w}'\mathbf{x}}}{1+e^{\mathbf{w}'\mathbf{x}}}$$

55

Logistic regression

- odds ratio

$$\log \frac{P(y=1|\mathbf{x})}{P(y=0|\mathbf{x})} = \mathbf{w}'\mathbf{x}$$

- linear regression model

$$z = \log \frac{P(y=1|\mathbf{x})}{P(y=0|\mathbf{x})} \Rightarrow z = \mathbf{w}'\mathbf{x}$$

56

Logistic regression: an example

January 1986: Explosion of the Space Shuttle Challenger

Launch temperature and *O Rings* failures

launch	temp.	failure	launch	temp.	failure	launch	temp.	failure
1	53	Y	9	68	N	17	75	N
2	56	Y	10	69	N	18	75	Y
3	57	Y	11	70	N	19	76	N
4	63	N	12	70	Y	20	76	N
5	66	N	13	70	Y	21	78	N
6	67	N	14	70	Y	22	79	N
7	67	N	15	72	N	23	80	N
8	67	N	16	73	N	24	81	N

Odds Ratio = $\ln \frac{p(x)}{1-p(x)} = \beta_0 + \beta_1 x$

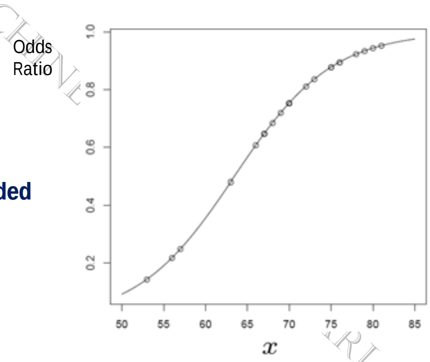
$p(x)$ **is the success probability (failure = N) when temp. =** x

Logistic regression: an example

Parameters estimate:

$\beta_1 = 0.17$

$\beta_0 =$ -10.8 (intercept)



Temperature when Challenger exploded was 31°F...

... the Odds Ratio was ?

$x = 31$ Odds Ratio = 0.0038 !!!

Classification and statistical learning theory

Choose the function $f^* \in \mathcal{H}$ (hypothesis) to minimize the empirical risk

$$R_{emp}(f) = \frac{1}{m}\sum_{i=1}^{m} V(y_i, f(\mathbf{x}_i))$$

Two critical issues:

- overfitting
- ill-posed optimization problem

Choose $f^* \in \mathcal{H}$ to minimize the structural risk (SRM principle)

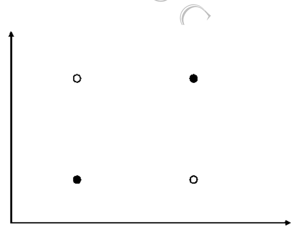
$$\hat{R}(f) = \frac{1}{m}\sum_{i=1}^{m} \underbrace{V(y_i, f(\mathbf{x}_i))}_{\text{loss function}} + \underbrace{\lambda \|f\|_K}_{\text{regularizer}}$$

Classification and statistical learning theory

Probabilistic upper bound on the expected risk (structural risk)

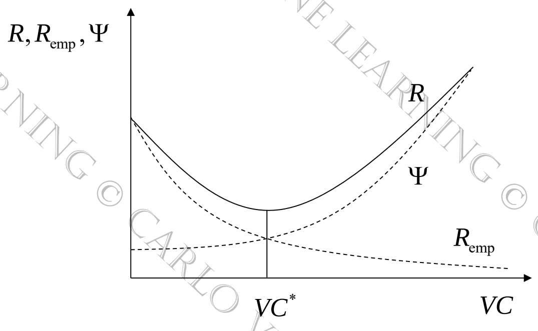
$$R(f) \leq \sqrt{\frac{\gamma \log(2t/\gamma) + 1 - \log(\eta/4)}{t}} + R_{emp}(f)$$
$$\leq \Psi(t, \gamma, \eta) + R_{emp}(f),$$

γ is the VC-dimension (Vapnik-Chervonenkis) expressing the complexity of the hypothesis space $\mathcal{F}$



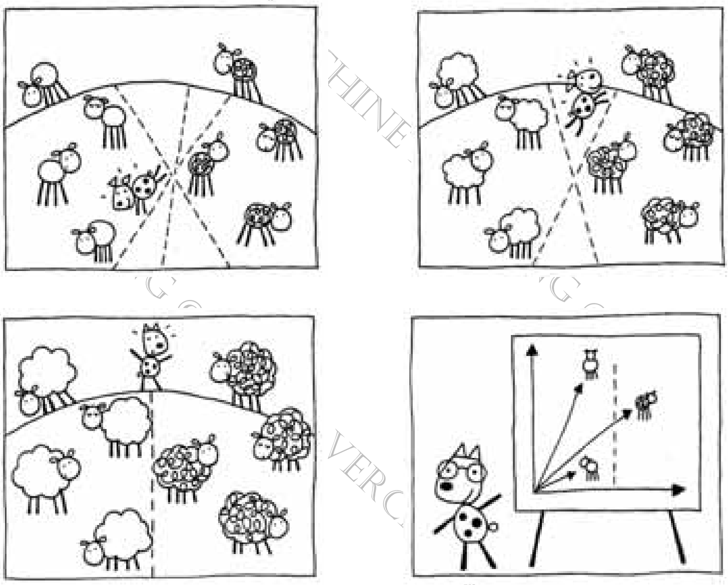
- The VC dimension of the lines in the plane is 3
- the VC dimension of the hyperplanes in the *n*-dimensional space is *n+1*

Classification and statistical learning theory



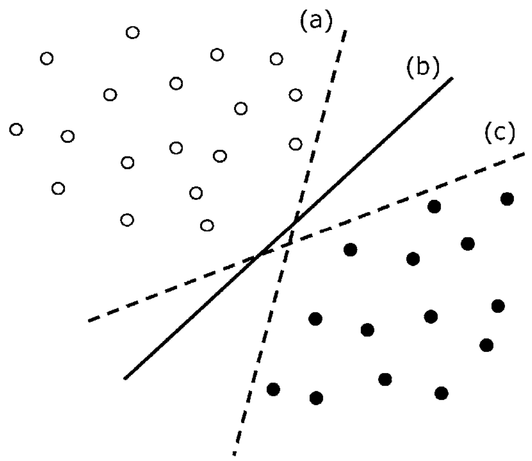
63

Accuracy vs. generalization

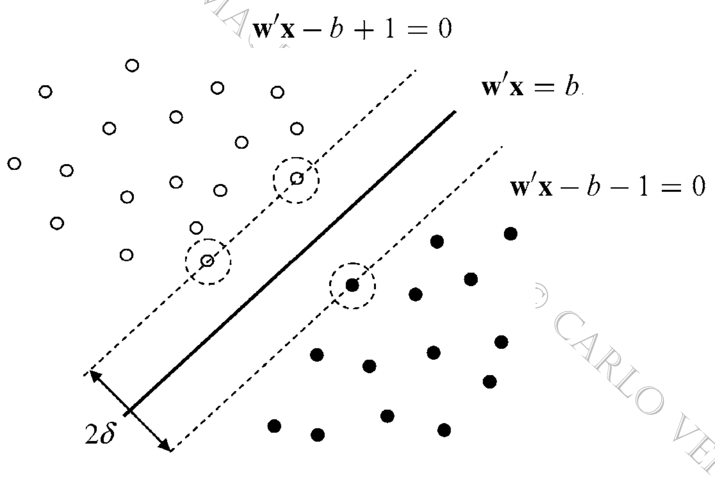


64

Support vector machines



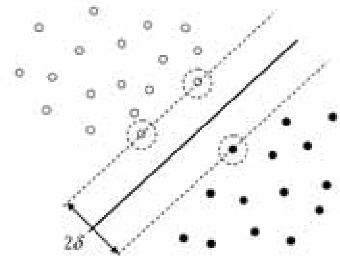
Support vector machines



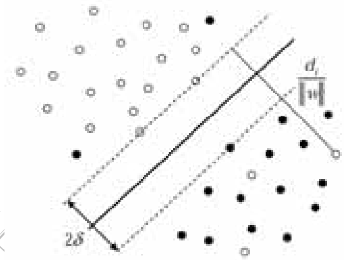
$$\delta = \frac{2}{\|\mathbf{w}\|}, \quad \text{dove} \quad \|\mathbf{w}\| = \sqrt{\sum_{j \in \mathcal{N}} w_j^2}$$

Support vector machines: SVM formulation

$$
\begin{aligned}
&\min_{\mathbf{w},b} \quad \frac{1}{2}\|\mathbf{w}\|_2^2 \\
&\text{s.t.} \quad y_i(\mathbf{w}'\mathbf{x}_i - b) \geq 1 \quad i \in \mathcal{M}
\end{aligned}
$$



$$
\begin{aligned}
&\min_{\mathbf{w},b,d} \quad \frac{1}{2}\|\mathbf{w}\|_2^2 + \lambda \sum_{i=1}^{m} d_i \\
&\text{s.t.} \quad y_i(\mathbf{w}'\mathbf{x}_i - b) \geq 1 - d_i \quad i \in \mathcal{M} \\
&\qquad\;\; d_i \geq 0 \quad i \in \mathcal{M}
\end{aligned}
$$



Support vector machines

$$
L(\mathbf{w}, b, \mathbf{d}, \boldsymbol{\alpha}\, \boldsymbol{\mu}) = \frac{1}{2}\|\mathbf{w}\|^2 + \lambda \sum_{i=1}^{m} d_i
$$

$$
- \sum_{i=1}^{m} \alpha_i [y_i(\mathbf{w}'\mathbf{x}_i - b) - 1 - d_i] - \sum_{i=1}^{m} \mu_i d_i
$$

$$
\frac{\partial L(\mathbf{w}, b, \mathbf{d}, \boldsymbol{\alpha}\, \boldsymbol{\mu})}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i = \mathbf{0},
$$

$$
\frac{\partial L(\mathbf{w}, b, \mathbf{d}, \boldsymbol{\alpha}\, \boldsymbol{\mu})}{\partial b} = \lambda - \alpha_i - \mu_i = 0,
$$

$$
\frac{\partial L(\mathbf{w}, b, \mathbf{d}, \boldsymbol{\alpha}\, \boldsymbol{\mu})}{\partial \mathbf{d}} = \sum_{i=1}^{m} \alpha_i y_i = 0,
$$

Support vector machines

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i,$$

$$\lambda = \alpha_i + \mu_i,$$

$$\sum_{i=1}^{m} \alpha_i y_i = 0.$$

$$L(\mathbf{w}, b, \mathbf{d}, \boldsymbol{\alpha}\ \boldsymbol{\mu}) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{k=1}^{m} y_i y_k \alpha_i \alpha_k \mathbf{x}_i' \mathbf{x}_k$$

Support vector machines

Lagrangean function

$$L(\mathbf{w}, b, \mathbf{d}, \boldsymbol{\alpha}\ \boldsymbol{\mu}) = \frac{1}{2}\|\mathbf{w}\|^2 + \lambda \sum_{i=1}^{m} d_i - \sum_{i=1}^{m} \alpha_i [y_i(\mathbf{w}'\mathbf{x}_i - b) - 1 + d_i] - \sum_{i=1}^{m} \mu_i d_i$$

Representation form

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i$$

Lagrangean function expressed as

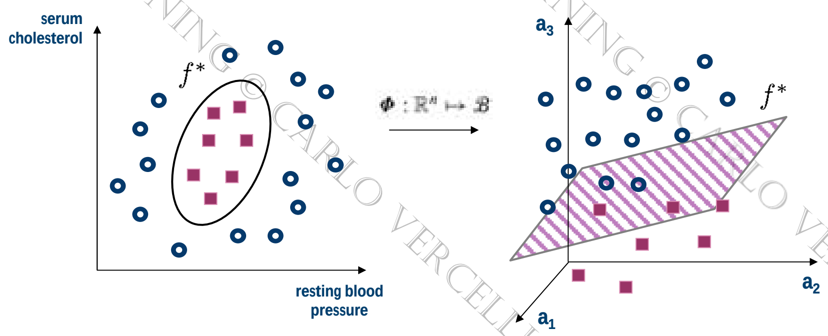
$$L(\mathbf{w}, b, \mathbf{d}, \boldsymbol{\alpha}\ \boldsymbol{\mu}) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{h=1}^{m} y_i y_h \alpha_i \alpha_h \mathbf{x}_i' \mathbf{x}_h$$

Prediction

$$f(\mathbf{x}) = \mathrm{sgn}(\mathbf{w}'\mathbf{x} - b)$$

Nonlinear separation with SVM

- Kernel methods: projection of the original input data into a higher dimensional Hilbert space, in an efficient way due to kernels
  - Vapnik, 1998, Statistical learning theory
  - Scholkopf & Smola, 2002, Learning with kernels
  - Shawe-Taylor & Cristianini, 2006, Kernel methods for pattern analysis



Nonlinear separation with SVM

Kernel

$$\Phi(\mathbf{x}_i)'\Phi(\mathbf{x}_h) = k(\mathbf{x}_i, \mathbf{x}_h)$$

Lagrangean function

$$L(\mathbf{w}, b, \mathbf{d}, \boldsymbol{\alpha}\, \boldsymbol{\mu}) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{h=1}^{m} y_i y_h \alpha_i \alpha_h k(\mathbf{x}_i, \mathbf{x}_h)$$

- Existence of meaningful kernel functions is guaranteed by Mercer Theorem
  - polynomial kernels $k(\mathbf{x}_i, \mathbf{x}_h) = (\mathbf{x}_i'\mathbf{x}_h + 1)^d$
  - radial basis kernels $k(\mathbf{x}_i, \mathbf{x}_h) = \exp\left(\frac{-\|\mathbf{x}_i - \mathbf{x}_h\|^2}{2\sigma^2}\right)$
  - neural networks kernels $k(\mathbf{x}_i, \mathbf{x}_h) = \tanh(\kappa \mathbf{x}_i'\mathbf{x}_h - \delta)$

Prediction

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{m} \alpha_i y_i \Phi(\mathbf{x})'\Phi(\mathbf{x}_i) - b\right)$$

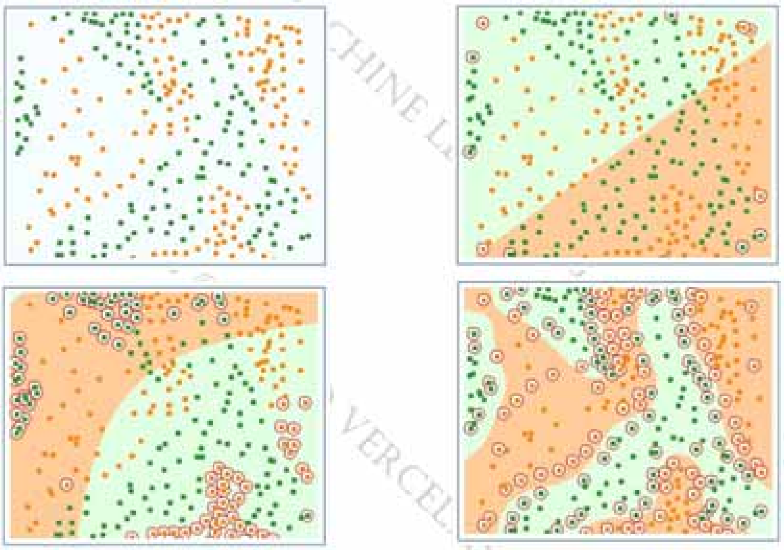
Support vector machines

- complementarity conditions (KKT theorem)

$$\alpha_i[y_i(\mathbf{w}'\mathbf{x}_i - b) - 1 + d_i] = 0, \quad i \in \mathcal{M}$$
$$\mu_i(\alpha_i - \lambda) = 0, \quad i \in \mathcal{M}$$

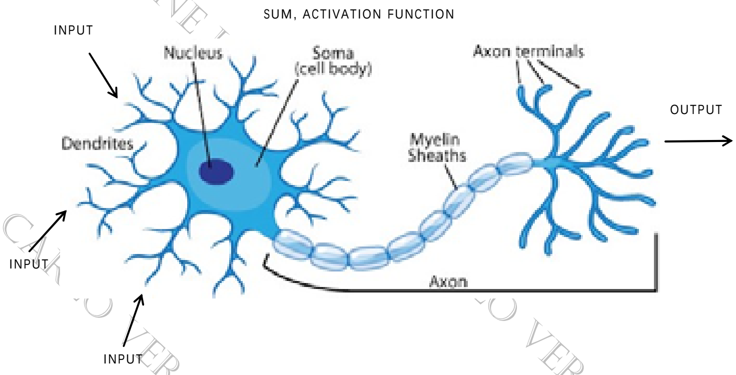
- the observations for which $0 < \alpha < \lambda$ are at distance $\frac{1}{\|w\|}$ from separating hyperplane and therefore lie on the canonical hyperplanes: support vectors

SVM: linear and nonlinear learning with kernels



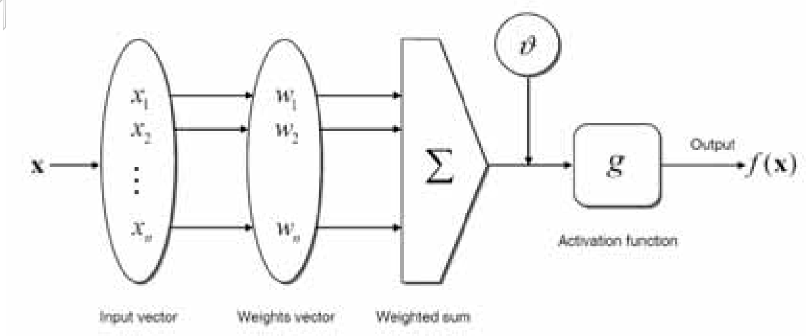
Neural networks: brain analogy

- It takes inputs (signals) from its dendrites (i.e. other neurons).
- The signals received from the dendrites are joined in the soma (a weighted sum of these inputs is computed).
- The result is passed on to the axon hillock (a specialized part of the soma that controls the firing of the neuron).
- If the weighted sum (i.e. total strength of the signal) is larger than a threshold limit, the neuron fires, otherwise it stays at rest.
- The state of the neuron (on/off) propagates through its axon and is passed on to the other connected neurons via its synapses.



75

Neural networks (artificial): perceptron

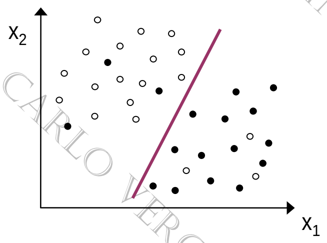


76

Neural networks

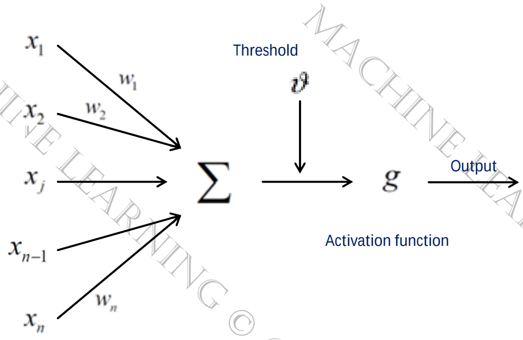
- separation by a hyperplane

$$z = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n - \vartheta = \mathbf{w}'\mathbf{x} - \vartheta.$$



77

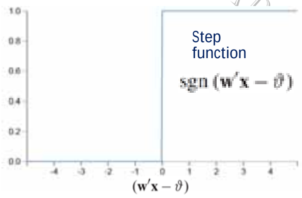
Rosenblatt's Perceptron

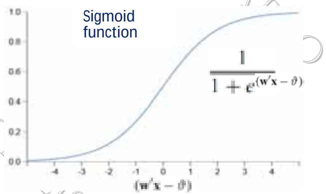


1. Initialize weights and threshold
2. Input an example and compute its output
3. Update weights according to the entity of the error made

   $w_1 = w_{1,old}$ + error x_1

4. Repeat steps 2 and 3 until the error at a given iteration is less than a threshold or a number of iterations have been reached

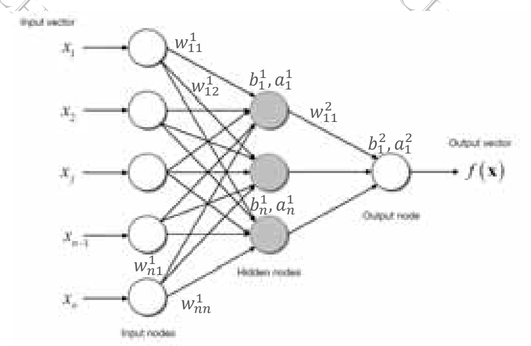




78

Multi-layer feed-forward neural networks

Each hidden unit can be considered as a single perceptron.

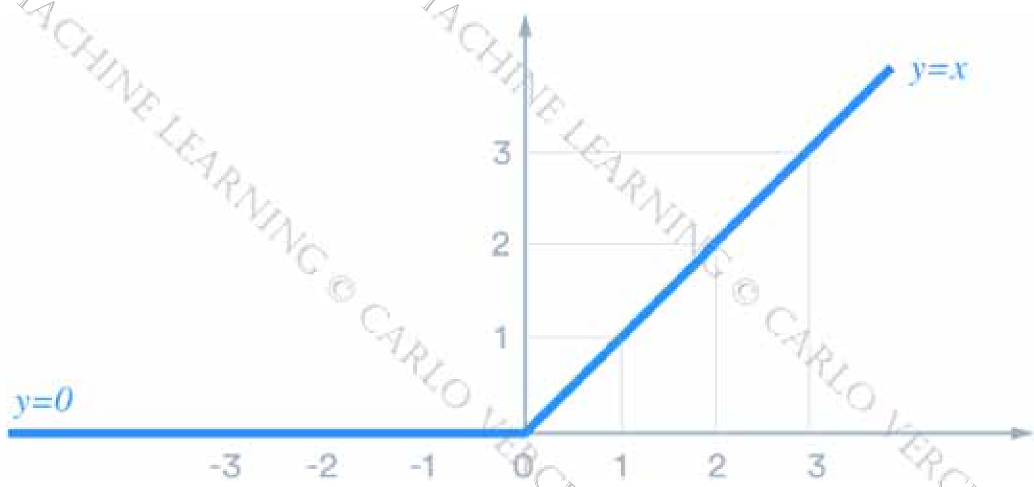


1. Initialize weights and threshold.
2. Input an example, compute its output and the corresponding error.
3. Starting with the output layer propagate the error back to the previous layer and update the weights according to the "strength" of the connection.
4. Repeat steps 2 and 3 for all the training examples.

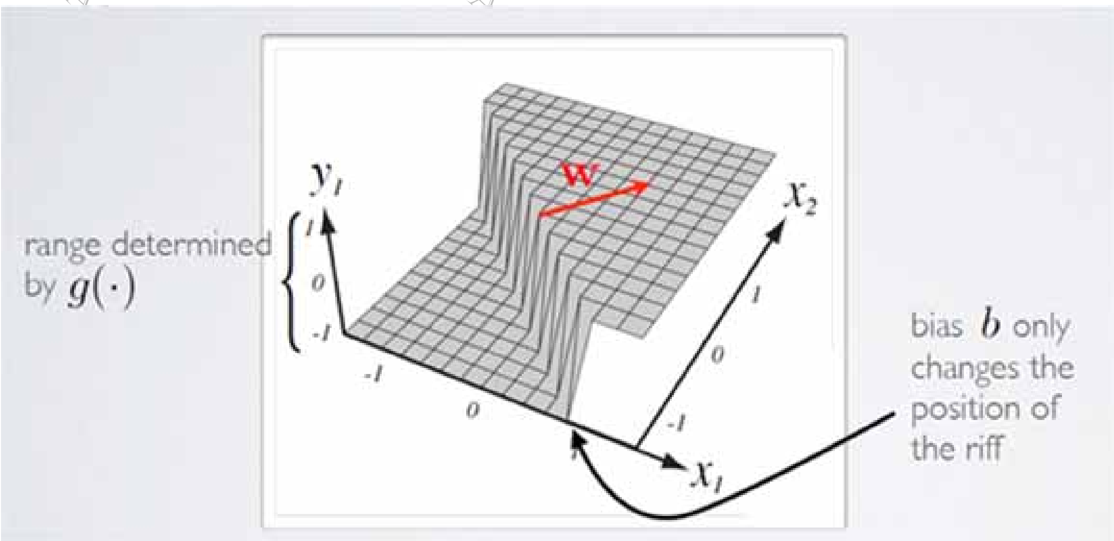
$$\boxed{w_j^{(k+1)} = w_j^{(k)} + \lambda(y_i - \hat{y}_i^{(k)})x_{ij}}$$

$a_i^l = \phi(\sum_n w_{ni}^l a_n^{l-1} + b_i^l)$, where ϕ is the element-wise activation function

ReLu activation function

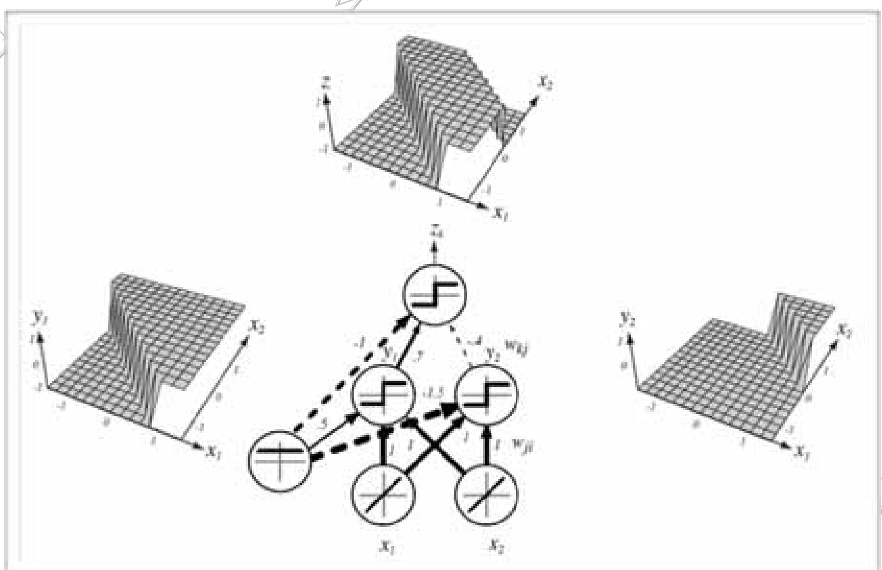


Nonlinear learning



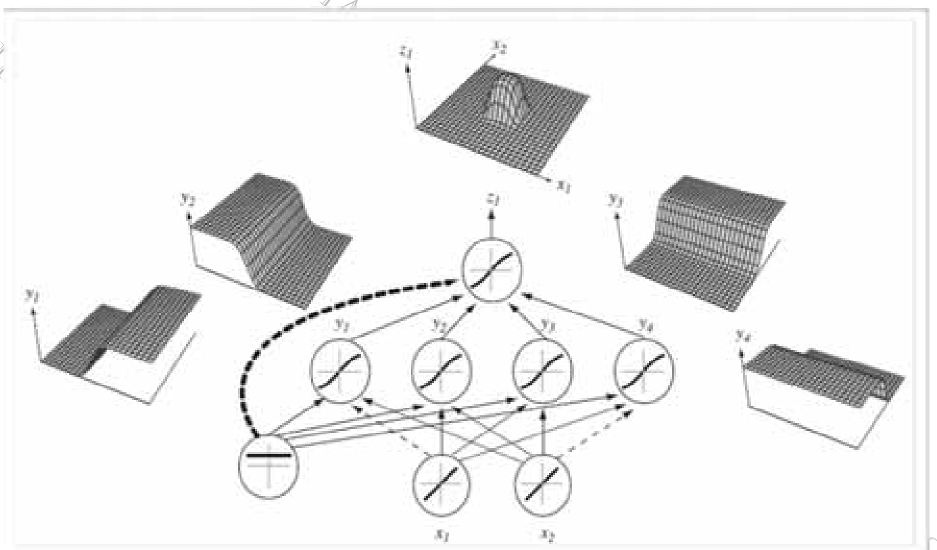
81

Nonlinear learning



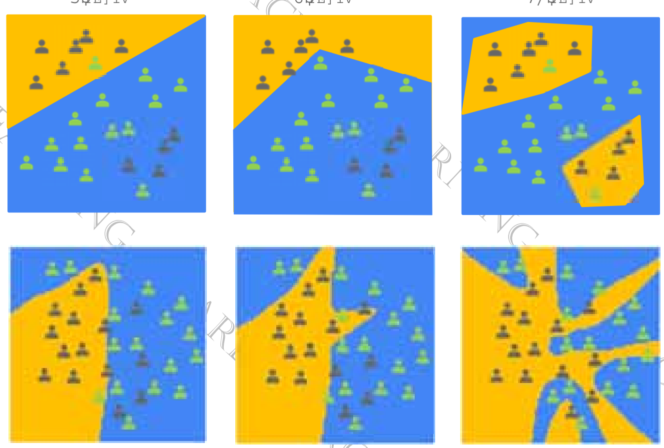
82

Nonlinear learning



83

Nonlinear learning



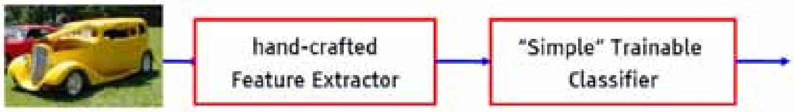
84

Deep Learning as Hierarchical Feature Learning

Deep learning can be summarized as learning both the representation and the classifier out of it

- Fixed engineered features (or kernels) + trainable classifier

Machine Learning



VS.

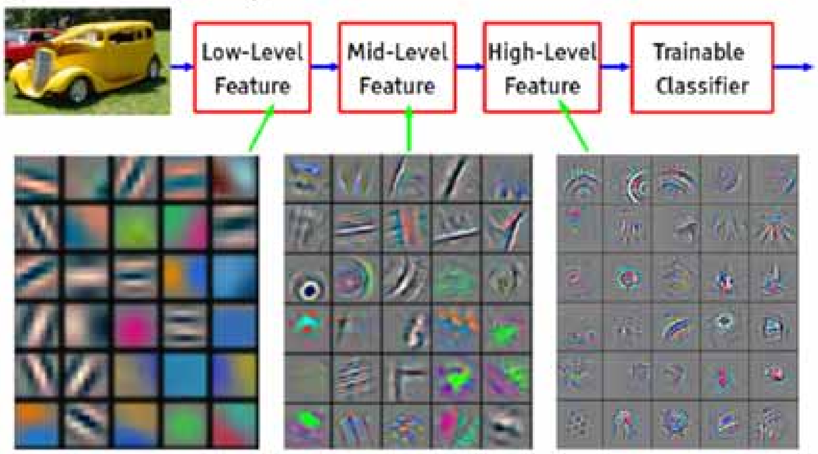
- End-to-end learning / feature learning / deep learning

Deep Learning

85

Deep Learning as Hierarchical Feature Learning

In deep learning we have multiple stages of non linear feature transformation

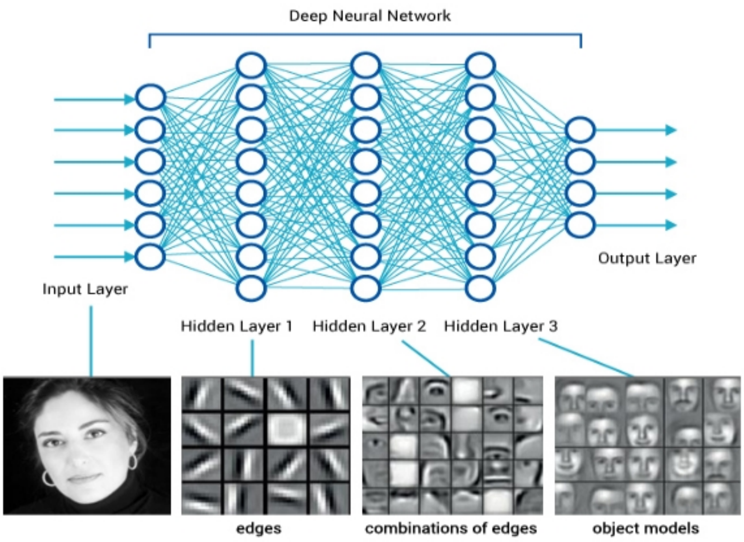


Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

Machine Learning © Carlo Vercellis POLITECNICO MILANO 1863

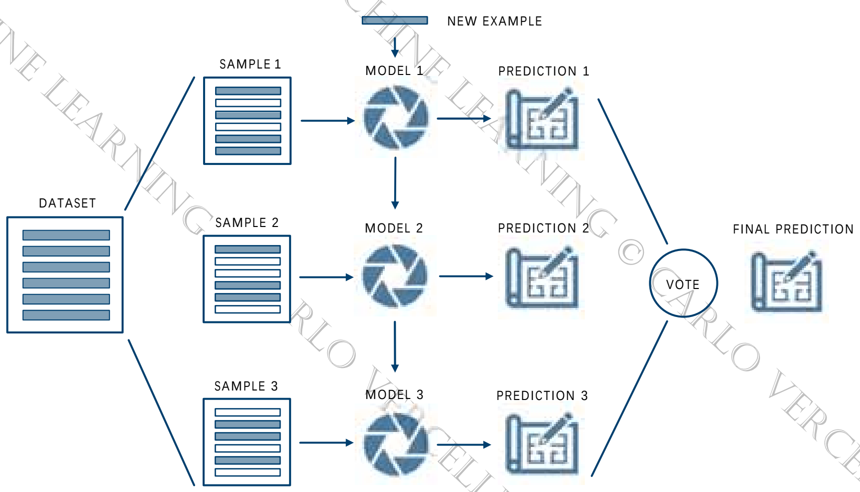
86

Image analysis



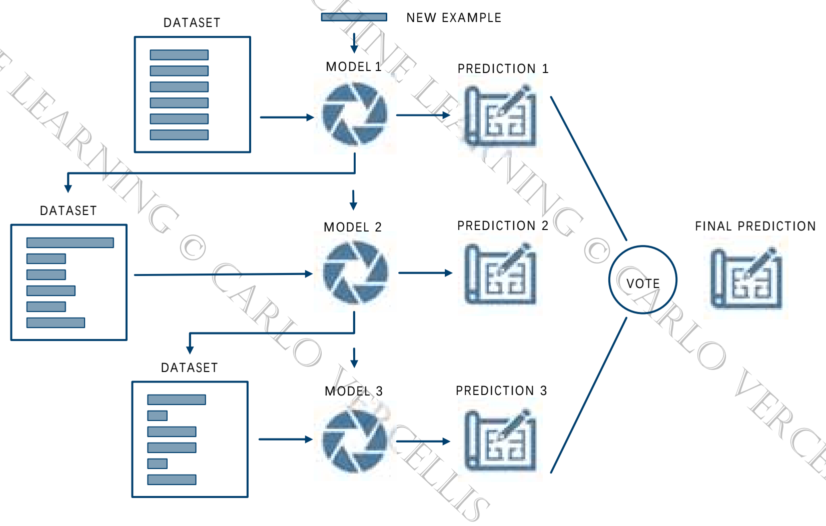
Ensemble framework: Bagging

Bagging (bootstrap aggregating) is an ensemble framework aimed at improving the stability and accuracy of a classifier. It helps to avoid overfitting.



Ensamble framework: Adaptive Boosting

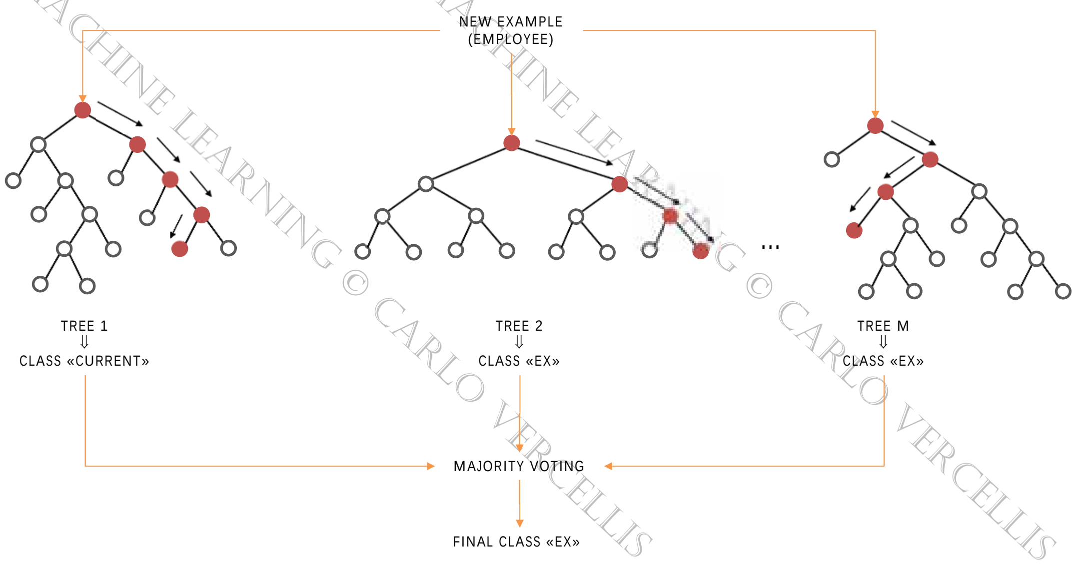
Adaptive Boosting (AdaBoost) is another ensemble framework. It focuses on "harder-to-classify" examples.



89

Random Forest

Each classifier is a classification tree and is generated using a random selection of attributes at each node to determine the split.



90

Adaboost

- Initially, all the weights of tuples are set the same (1/d)
- Generate k classifiers in k rounds. At round i,
  - Tuples from D are sampled (with replacement) to form a training set D_i of the same size
  - Each tuple's chance of being selected is based on its weight
  - A classification model M_i is derived from D_i and its error rate is calculated
  - If a tuple is misclassified, its weight is increased, otherwise it is decreased
- Error rate: $err(X_j)$ is the misclassification error of tuple X_j. Classifier M_i error rate is the sum of the weights of the misclassified tuples:

$$error(M_i) = \sum_{j}^{d} w_j \times err(\mathbf{X_j})$$

- The weight of classifier M_i's vote is $\log \frac{1-error(M_i)}{error(M_i)}$