Data Mining algorithms

2017-2018 spring

02.07-09.2018

Overview

Classification vs. Regression

Evaluation I

Basics

Bálint Daróczy

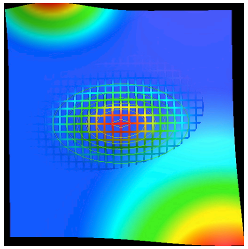
daroczyb@ilab.sztaki.hu

Basic reachability: MTA SZTAKI, Lágymányosi str. 11

Web site:

http://cs.bme.hu/~daroczyb/DM_2018_spring

(slides will be uploaded after class)



Requirements

Lectures: 2x(2x45) min., wed and fri 12pm – 2pm

Where? IB134

Can we start at 12:15 with a 5 min. break and finish at 13:50?

Project work: challenge?

Tests: midterm (7th week?) + exam

References

1. Tan, Steinbach, Kumar (TSK): Introduction to Data Mining

    Addison-Wesley, 2006, Cloth; 769 pp, ISBN-10: 0321321367, ISBN-13: 9780321321367

    http://www-users.cs.umn.edu/~kumar/dmbook/index.php

2. Leskovic, Rajraman, Ullmann: Mining of Massive Datasets

    http://infolab.stanford.edu/~ullman/mmds.html

3. Devroye, Győrfi, Lugosi: A Probabilistic Theory of Pattern Recognition, 1996

4. Rojas: Neural Networks, Springer-Verlag, Berlin, 1996

5. Hopcroft, Kannan: Computer Science Theory for the Information Age
http://www.cs.cmu.edu/~venkatg/teaching/CStheory-infoage/hopcroft-kannan-feb2012.pdf

+ papers

Main topics

- Evaluation of classifiers: cross-validation, bias-variance trade-off
- Supervised learning (classification): nearest neighbour methods, decision trees, logistic regression, non-linear classification, neural networks, support vector networks timeseries classification and dynamic time warping
- Linear and polynomial, one and multidimensional regression and optimization: gradient descent and least squares
- Advanced classification methods: semi-supervised learning, multi-class classification, multi-task learning, ensemble methods: bagging, boosting, stacking, ensemble
- Clustering: k-means (k-medoid, FurthestFirst), hierarchical clustering, Kleinberg's impossibility theorem, internal and external evaluation, convergence speed
- Principal component analysis, low-rank approximation, collaborative filtering and applications (recommender systems, drug-target prediction)
- Density estimation and anomaly detection
- Frequent itemset mining
- Additional applications and problems: preprocessing, scaling, overfitting, hyperparameter optimization, imbalanced classification

Tools

Scikit (mainly)

Chainer

Tensorflow

Keras

Weka (some)

DATO (opt.)

Underlying: python (numpy), R etc.

Server: at SZTAKI (unfortunately w/o GPU)

Projects

Some ideas:

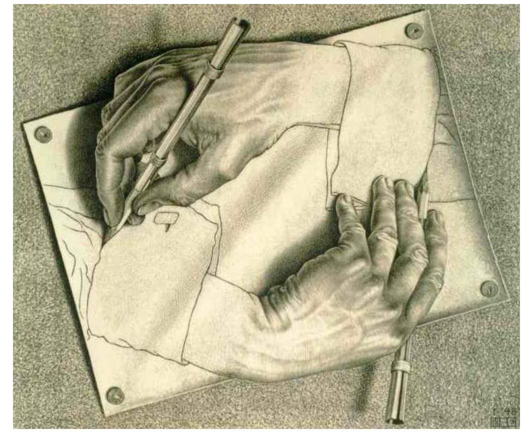
Text mining/classification
- trust and bias
- embeddings
- network?

Recommendation system:
- item-to-item recommendation
- regular explicit

Image:
- classification/reconstruction
- medical image classification

Team work would be preferable

Presentation at the end of the semester



User/Movie	Napoleon Dynamite	Monster RT.	Cindarella	Life on Earth
David	1	?	?	3
Dori	5	3	5	5
Peter	?	4	3	?

Representation

Dataset: set of objects, with some known attributes

Hypothesis: the attributes represent and differentiate the objects

E.g. attribute types:

- binary
- nominal
- numerical
  - string
  - date

Attributes, “features”

“records”

Tid	**Refund**	**Marital Status**	**Taxable Income**	**Cheat**
1	Yes	Single	125K	**No**
2	No	Married	100K	**No**
3	No	Single	70K	**No**
4	Yes	Married	120K	**No**
5	No	Divorced	95K	**Yes**
6	No	Married	60K	**No**
7	Yes	Divorced	220K	**No**
8	No	Single	85K	**Yes**
9	No	Married	75K	**No**
10	No	Single	90K	**Yes**

Representation

Attributes, "features"

Structure:

- sequential
- spatial

Sparse or dense

We presume that the set of attributes are previously known and fixed

Missing values?

"records"

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Machine learning

Let be a finite set $X=\{x_1,..,x_T\}$ in R^d and for each point a label $y=\{y_1,..,y_T\}$ usually in $\{-1,1\}$. The problem of binary classification is to find a particular f(x) which approximate y over X.

How to measure the performance of the approximation?
How to choose the function class?
How to find a particular element in the chosen function class?
How to generalize?

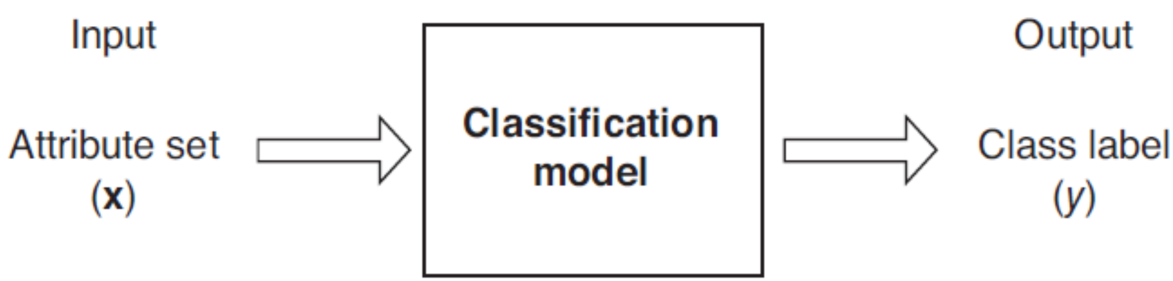
Classification vs. regression?

Classification

E.g. the problem of learning a half-space or a linear separator. The task is to find a d-dimensional vector w, if one exists, and a threshold b such that

$$w \cdot x_i > b \text{ for each } x_i \text{ labelled} +1$$
$$w \cdot x_i < b \text{ for each } x_i \text{ labelled } -1$$

A vector-threshold pair, (w, b), satisfying the inequalities is called a linear separator -> dual problem: high dimensional learning via kernels (inner products)



Classification

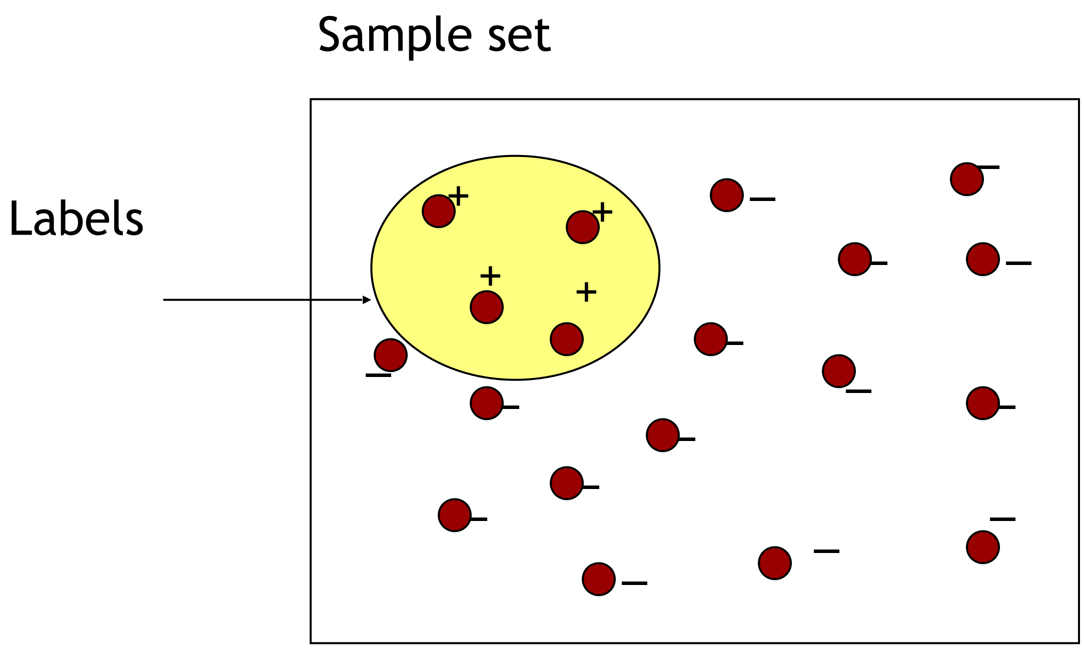


Fig.: TSK

Clustering, is it regression?

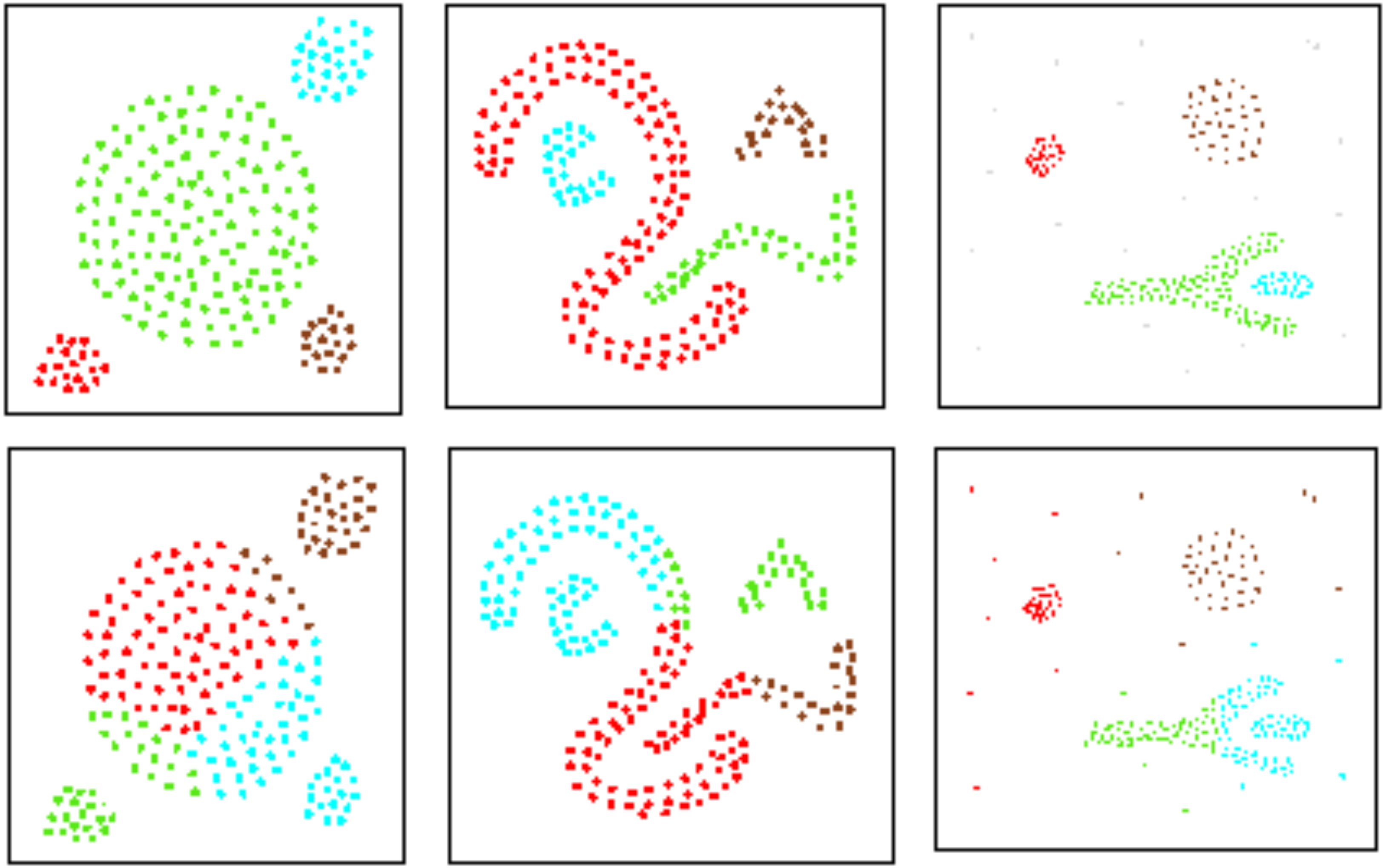


Fig.: TSK

K-Means

Presumption: our data points are in a vector space.

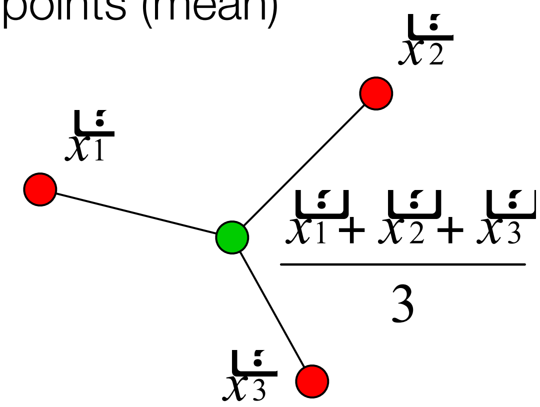
K-means (D, k)

Init: Let C_1, C_2,... , C_k be the centroids of the clusters

While the centroids change:

assign every point in D to the cluster with the closest centroid

Update the centroids according to the assigned points (mean)



$$\frac{\vec{x}_1 + \vec{x}_2 + \vec{x}_3}{3}$$

The initial centroids are:

a) random points from D

b) random vectors

When do we stop?

a) the centroids are not changing

b) the approximation error is below a threshold

c) we reach the maximal number of allowed iterations

K-means

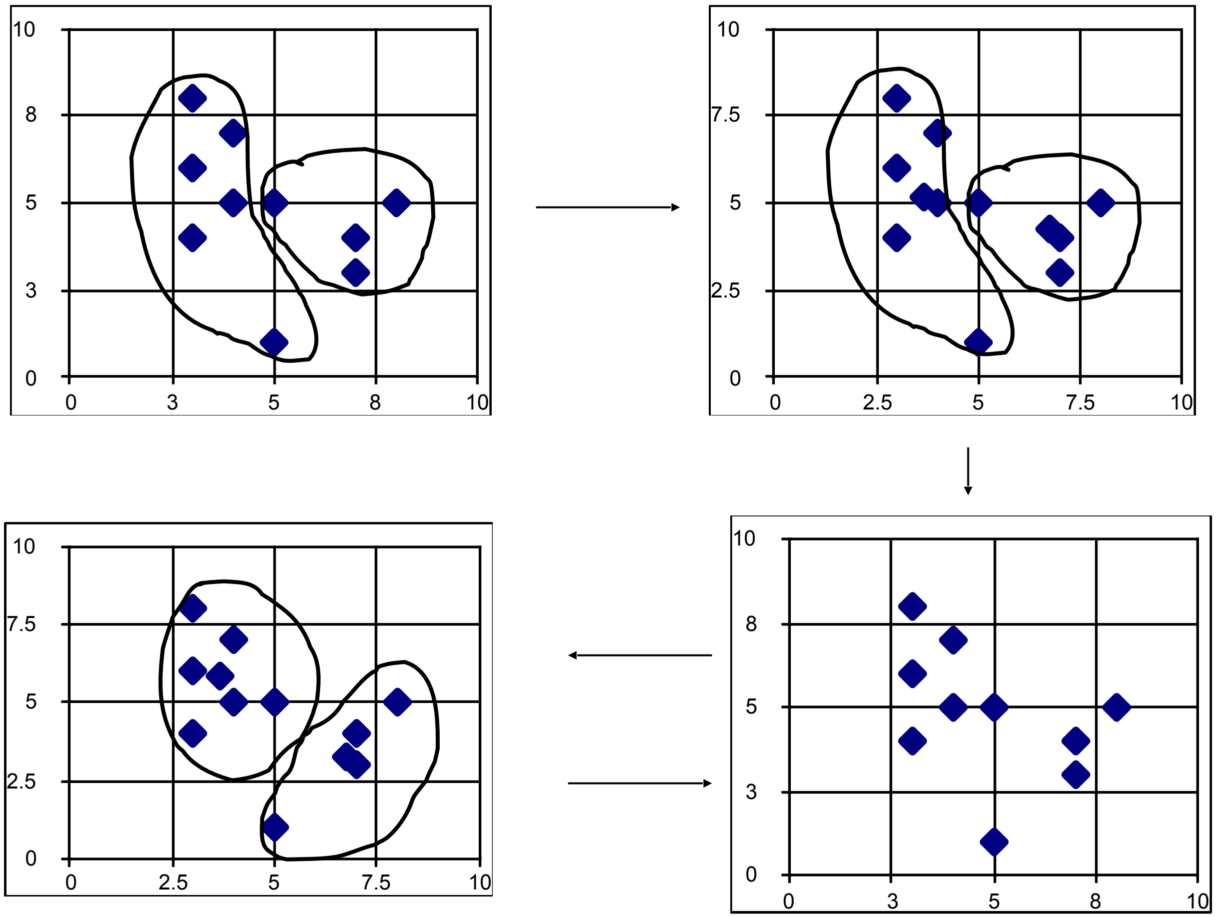
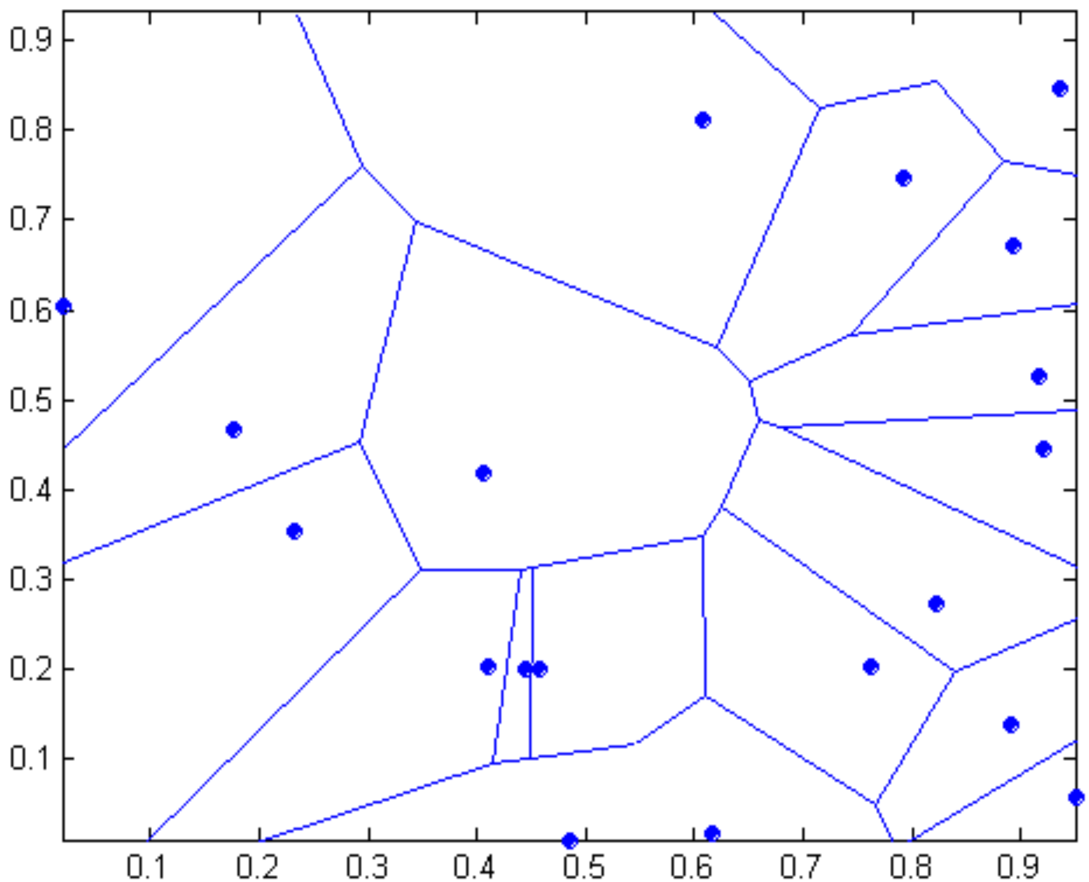


Fig.: TSK

K-means



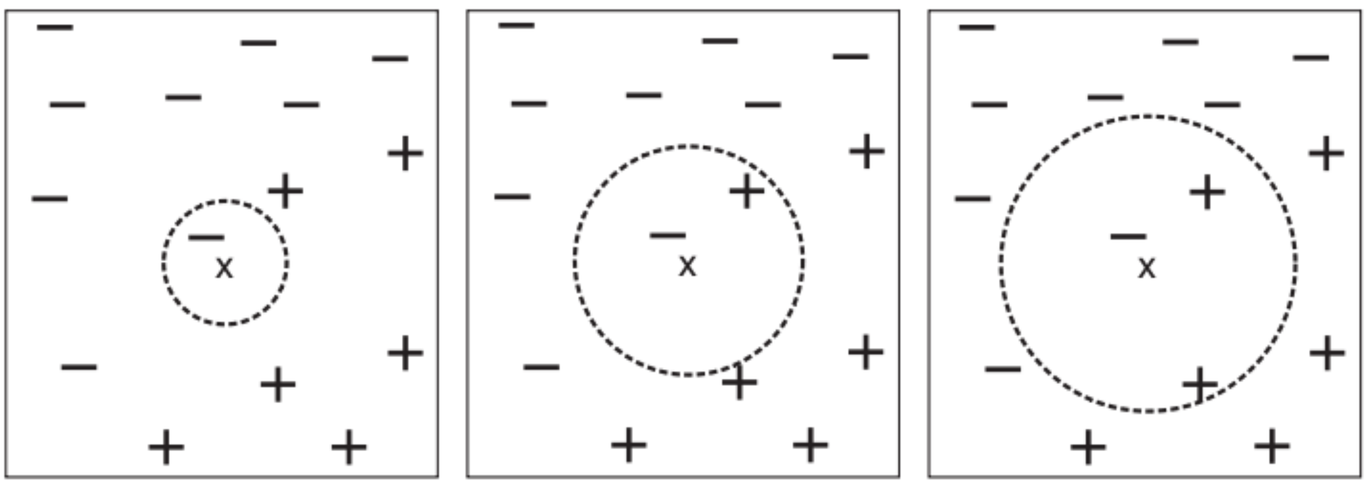
K- nearest neighbor (K-NN)

Hypothesis:

“If it walks like a duck, swim like a duck, eat like a duck than it is a duck!”

1. Find k nearest training points
2. Majority vote

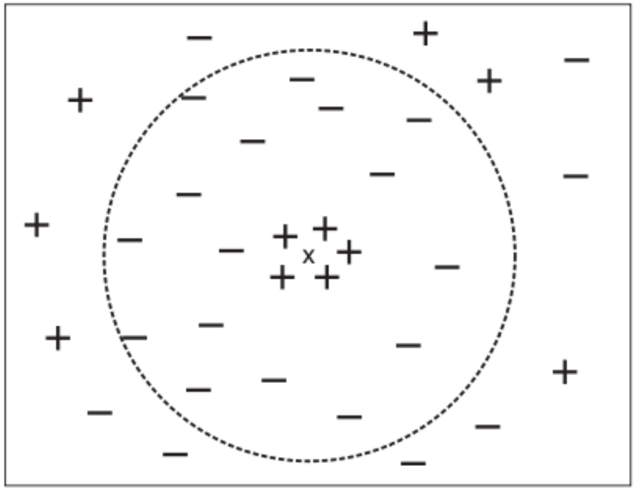
E.g.:



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

Fig.: TSK

K- nearest neighbor (K-NN)



Why it is not a good classifier?

Machine learning algorithms are either

Eager: the algorithm builds a model and predicate using only the model
or
Lazy: the algorithm use the training set during prediction

kNN is lazy
Complexity? Generalization?

Fig.: TSK

K- nearest neighbor (K-NN)

E.g. Distance/divergence metrics:

- Minkowski
- Mahalanobis $D_M(x) = \sqrt{(x-\mu)^T S^{-1}(x-\mu)}.$
- Cosine, Jaccard, Kullback-Leibler, Jensen-Shannon etc.

Notes:

- scale
- normalization

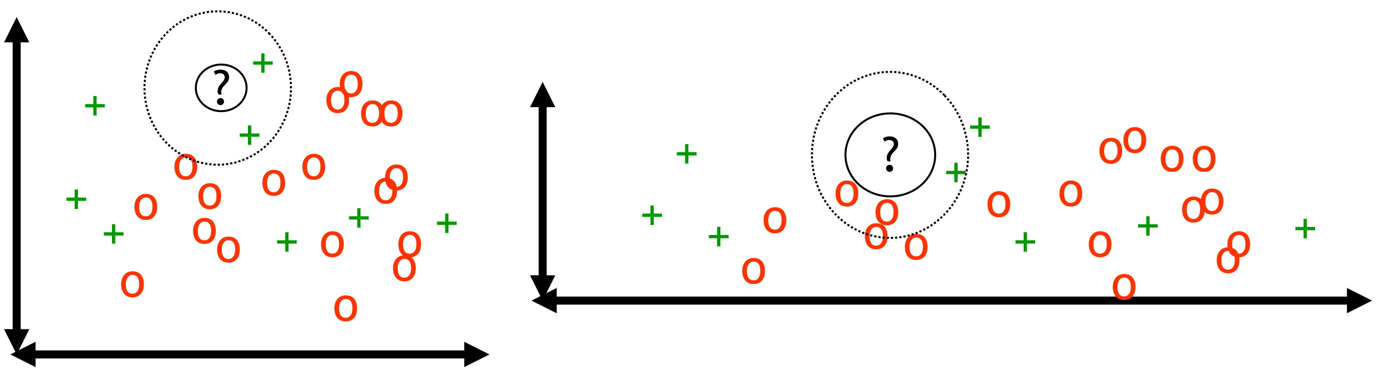


Fig.: TSK

Johnson-Lindenstrauss lemma (1984)

Theorem

Any finite set of points $X = \{x_1, ..., x_n\}$ *in* $\mathbb{R}^d$ *can be projected into a* $k = O(\epsilon^{-2}\log(n))$ *dimensional space while preserving the pairwise L2 distances with some distortion:*

$$\sqrt{\frac{k}{d}}\|x_i - x_j\|_2(1-\epsilon) \leq \|\theta(x_i) - \theta(x_j)\|_2 \leq \sqrt{\frac{k}{d}}\|x_i - x_j\|_2(1+\epsilon) \quad (1)$$

Main questions:

1. what is the constant in k? e.g. n=1M, $\epsilon = 0.01$, then $k \approx c * 120000$
2. what is the transformation? JL transform: random orthogonal unit vectors uniformly chosen from S^{d-1} (the unit sphere in $\mathbb{R}^d$)

Johnson-Lindenstrauss lemma

The JL transform satisfies three main properties:

1. Spherical symmetry: for any orthogonal matrix A, the transformed A and the transformation θ has the same distribution
2. Orthogonality
3. Normality

Indyk & Motwani (1998):

No orthogonality, no normality
Independently draw each entry in θ from $\mathcal{N}(0, \frac{1}{d})$ yet it satisfies JL.
On expectation the normality and the orthogonality are satisfied:

$$\mathbf{E}[< \theta_i, \theta_j >] = 0, \mathbf{E}[< \theta_i, \theta_i >] = 1 \tag{2}$$

$O(\epsilon^{-2} d log(n))$ in time and $n^{O(\epsilon^{-2})}$ in space

Johnson-Lindenstrauss lemma

Achlioptas (2003):

No spherical symmetry

For all unit vectors x, the $\theta(x)_i^2$ concentrated around mean $\frac{1}{d}$

Distribution 1: Choose each entry in θ uniformly from $\{-\frac{1}{\sqrt{d}}, \frac{1}{\sqrt{d}}\}$

Distribution 2: Choose each entry independently as:

$$\theta_{ij} = \begin{cases} (\frac{d}{3})^{-1/2}, & \text{w.p. } 1/6 \\ 0, & \text{w.p. } 2/3 \\ -(\frac{d}{3})^{-1/2}, & \text{w.p. } 1/6 \end{cases}$$

Sparse: 2/3 of the entries are zero, going lower may distort the sparse vectors

Johnson-Lindenstrauss lemma

Ailon & Chazelle (2009):

Heisenberg principle:

A singal and its spectrum cannot be both concentrated.

Key idea:

Preprocess the vectors with Fourier (actually with Walsh-Hadamard)

$O(d\log(d) + \epsilon^{-2} log^3(n))$ in time (if d is large enough)

They assume that $d = 2^m > k$ (because of FFT) and $d = \Omega(\epsilon^{-1/2})$ and $n \geq d$:(

The final transformation is $\theta = PHD$:)

Johnson-Lindenstrauss lemma

Ailon & Chazelle (2009):

The final transformation is $\theta = PHD$:)

$$\begin{pmatrix} & & \\ & JL & \\ & & \end{pmatrix} \begin{pmatrix} & & \\ & Walsh - Hadamard & \\ & & \end{pmatrix} \begin{pmatrix} \pm 1 & & & & \\ & \pm 1 & & & \\ & & \ddots & & \\ & & & \pm 1 & \\ & & & & \pm 1 \end{pmatrix}$$

- P: kxd, with probability $q = \min\{O(\frac{\log^2(n)}{d}), 1\}$ i.i.d from $\mathcal{N}(0, 1/q)$ otherwise zero
- H: Walsh-Hadamard
- D: diagonal, flipping coins with prob. $1/2$ with values $\{-1, +1\}$

Johnson-Lindenstrauss lemma

For same certain type of points $\|Px\|_2$ has high variance, especially if a point is very sparse (e.g. one non-zero element)

However if we precondition with HD the vectors will be suitable to be transformed with P while satisfy JL (with a certain prob., see lemma 1 in Ailon & Chazelle (2009)) aka it densifies the sparse input vectors

Johnson-Lindenstrauss lemma

OK, we should stop, since the next step is a bit far away. Yet.

But wait …

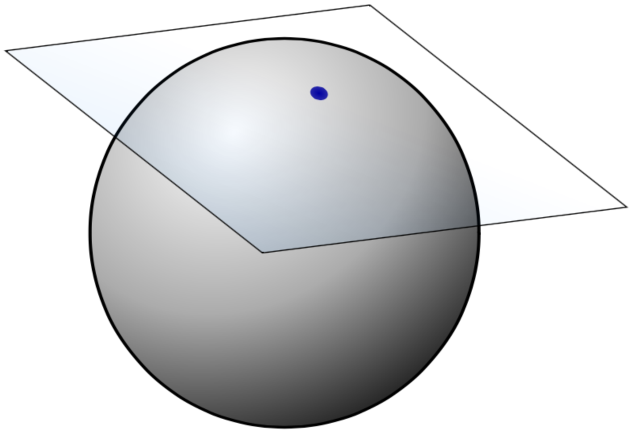
What may be the next step?

Are there any other methods to approximate distance or approximate NN?

e.g. Riemannian Manifold

Given a smooth (or differentiable) n-dimensional manifold M, a Riemannian metric on M (or TM) is a family of inner products $(\langle \bullet, \bullet \rangle_p)_{p \in M}$ on each tangent space T_pM, such that the inner product depends smoothly on p.

A smooth manifold M, with a Riemannian metric is called a Riemannian manifold.



Riemannian Metric

Let γ: [x, y] be a continuously differentiable curve in M.

The length of a curve γ on M is defined as integrating the length of the tangent vector dγ (d is a differential operator).

Example: $g_{11}\ dx_1^2 + g_{12}\ dx_1 dx_2 + g_{22}\ dx_2^2 \ \ldots.$

If g_{ij} is the Kronecker delta it will be the Euclidean.

The distance d(x,y) is the shortest among the curves between x and y.

OK, at this point we should really stop! Do not worry, we will come back.

Evaluation

Confusion matrix

(binary classification):

Ground truth / predicted class	pos	neg	Total
pos	True Positive (TP)	False Negative (FN)	TP+FN
neg	False Positive (FP)	True Negative (TN)	FP+TN
Total	TP+FN	FP+TN	

Evaluation

Accuracy: proportion of correctly classified instances

TP+TN/(TP+FP+TN+FN)

Precision (p): proportion of correctly classified positive instances in the set of instances with positive predicted label

TP/(TP+FP)

Recall (r): proportion of correctly classified positive instances

TP/(TP+FN)

F-measure: harmonic mean of precision and recall

(2*p*r/(p+r))

Evaluation

False-Positive Rate (FPR) = FP/(FP+TN)

True-Positive Rate (TPR) = TP/(TP+FN)

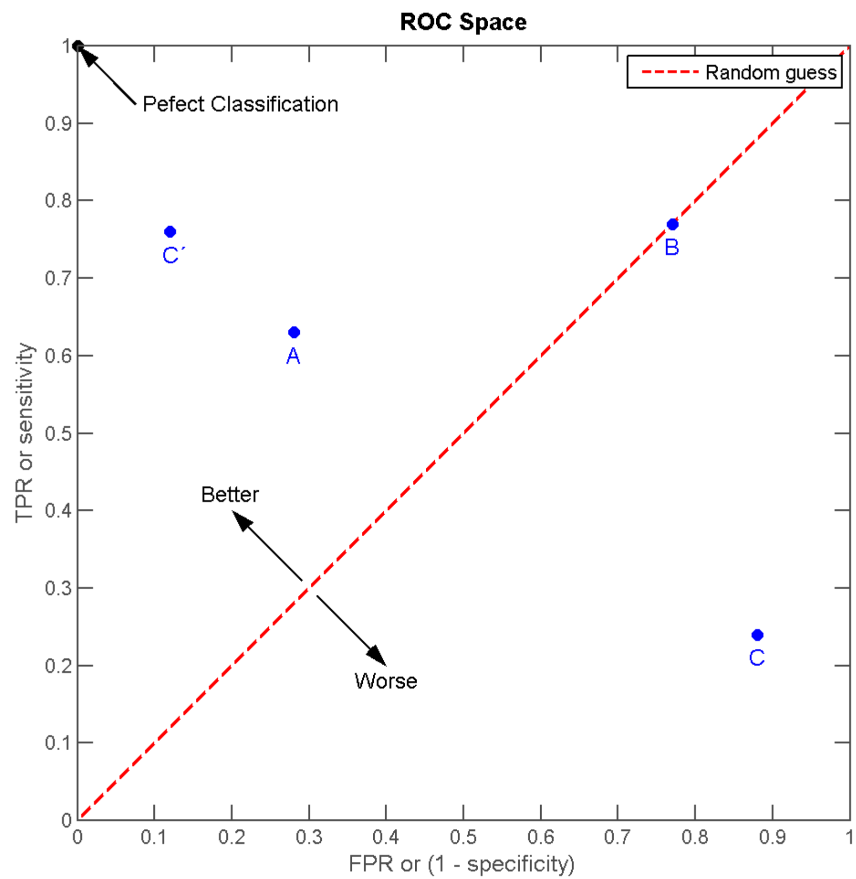
ROC: Receiver Operating Characteristic

MAP: Mean Average Precision (Friday)

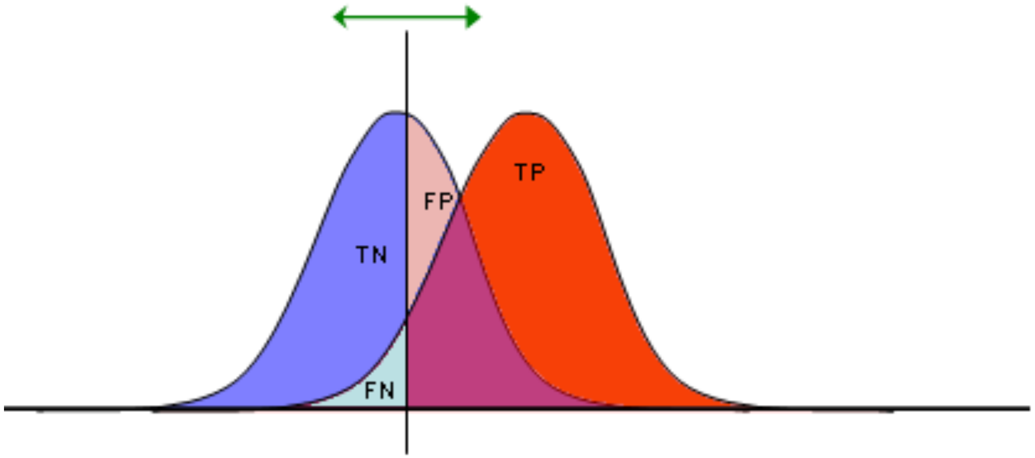
$$\text{AveP} = \frac{\sum_{r=1}^{N} (P(r) \times \text{rel}(r))}{\text{number of relevant documents}}$$

nDCG: normalized Discriminative Cummulative Gain (later)

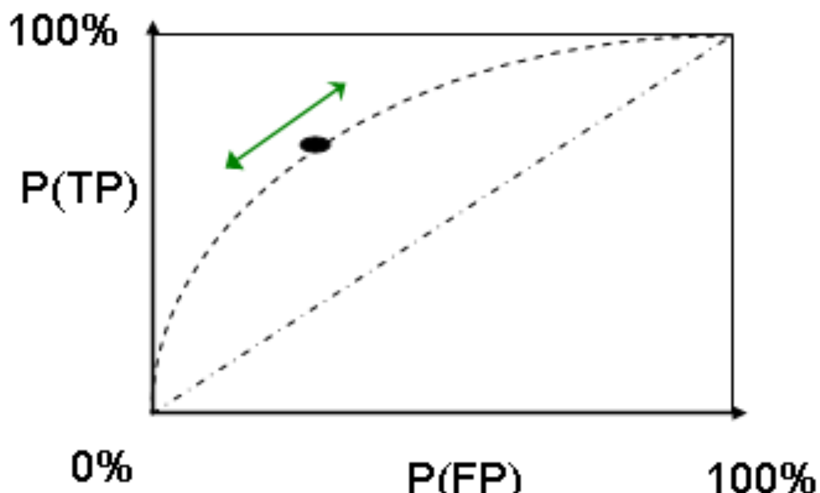
$$\mathrm{DCG_p} = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{\log_2(i)}$$



Evaluation, tradeoff



TP	FP
FN	TN
1	1



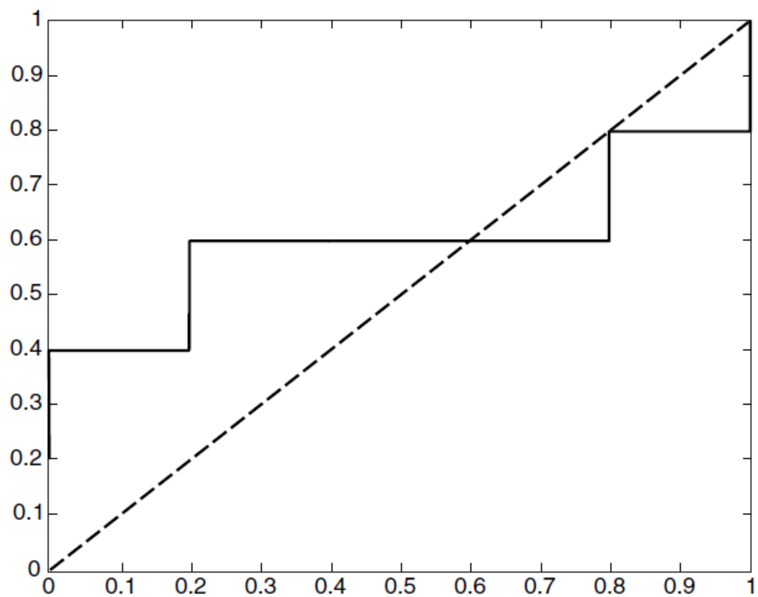
ROC: Receiver Operating Characteristic

- Only for binary classification
- Area Under Curve: prop. with the probability of correct separation
- threshold independent
- Presumption: available scores (ties?)

Class	+	−	+	−	−	−	+	−	+	+	
	0.25	**0.43**	**0.53**	**0.76**	**0.85**	**0.85**	**0.85**	**0.87**	**0.93**	**0.95**	**1.00**
TP	5	4	4	3	3	3	3	2	2	1	0
FP	5	5	4	4	3	2	1	1	0	0	0
TN	0	0	1	1	2	3	4	4	5	5	5
FN	0	1	1	2	2	2	2	3	3	4	5
TPR	1	0.8	0.8	0.6	0.6	0.6	0.6	0.4	0.4	0.2	0
FPR	1	1	0.8	0.8	0.6	0.4	0.2	0.2	0	0	0

ROC: Receiver Operating Characteristic

Class	+	−	+	−	−	−	+	−	+	+	
	0.25	0.43	0.53	0.76	0.85	0.85	0.85	0.87	0.93	0.95	1.00
TP	5	4	4	3	3	3	3	2	2	1	0
FP	5	5	4	4	3	2	1	1	0	0	0
TN	0	0	1	1	2	3	4	4	5	5	5
FN	0	1	1	2	2	2	2	3	3	4	5
TPR	1	0.8	0.8	0.6	0.6	0.6	0.6	0.4	0.4	0.2	0
FPR	1	1	0.8	0.8	0.6	0.4	0.2	0.2	0	0	0



AUC=?

+	+	-	+	-	-	+	+	-	+	
0.16	0.32	0.42	0.44	0.45	0.51	0.78	0.87	0.91	0.93	Score
										TP
										FN
										TN
										FP
										TPR
										FPR

Some exercise:
How to compare models?
AUC?

						+	-	+	+	
0.43	0.56	0.62	0.78	0.79	0.86	0.89	0.89	0.91	0.96	Score
										TP
										FN
										TN
										FP
										TPR
										FPR