Lecture 2
Non-parametric methods

Machine Learning

Andrey Filchenkov

13.09.2019

Lecture plan

- Validation techniques
- Similarity-based classification and regression
- One nearest neighbor
- *k*NN and hyperparameter tuning
- Generalized distance-based classifiers
- Nonparametric regression
- Prototype selection
- Performance measures

- The presentation is prepared with materials of the K.V. Vorontsov's course "Machine Learning".
- Slides are available online: **bit.ly/2lynu3L**

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

2

Lecture plan

- Validation techniques
- Similarity-based classification and regression
- One nearest neighbor
- kNN and hyperparameter tuning
- Generalized distance-based classifiers
- Nonparametric regression
- Prototype selection
- Performance measures

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

3

Example from the previous lecture

$$y(x) = \frac{1}{1 + 25x^2}; \quad a(x) - \text{polynomial of degree } n = 38$$
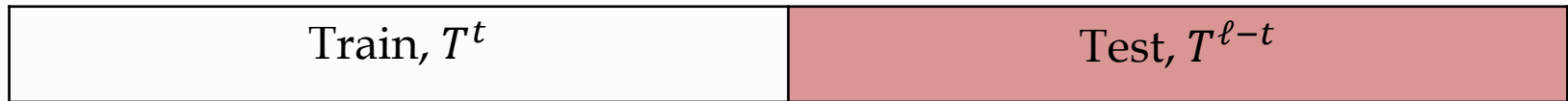
Machine learning. Lecture 2. Non-parametric. 13.09.2019.

4
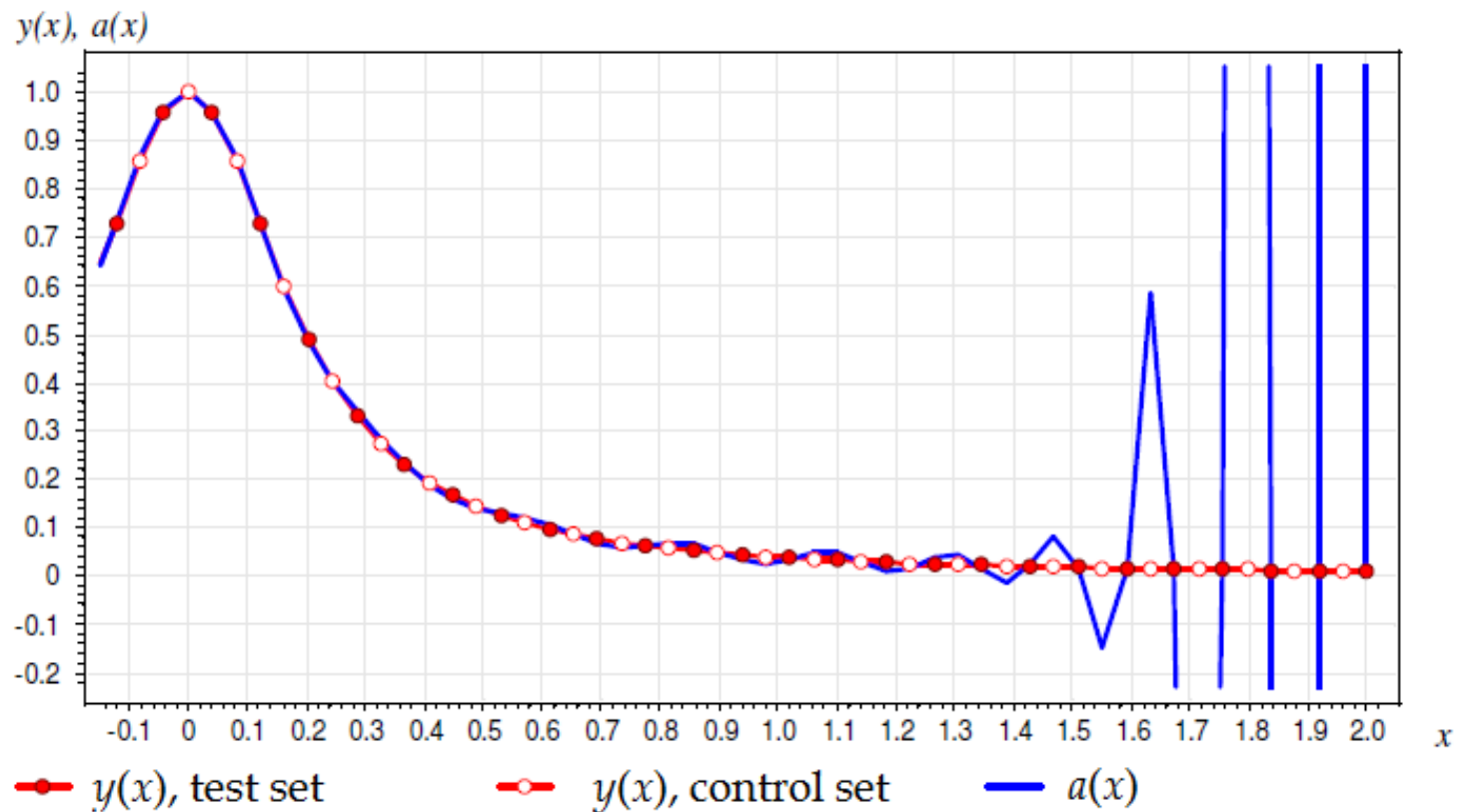
Hold-out validation

Hold-out validation, HO

Split training sample into two parts:
$$T^\ell = T^t \cup T^{\ell-t}$$

Train, T^t	Test, $T^{\ell-t}$

Solve the optimization problem:
$$\mathrm{HO}\big(\mu, T^t, T^{\ell-t}\big) = Q\big(\mu(T^t), T^{\ell-t}\big) \to \min$$

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

5

Complete cross-validation

Choose value of t.

Split the sample with all the possible ways on T^t and $T^{\ell-t}$.

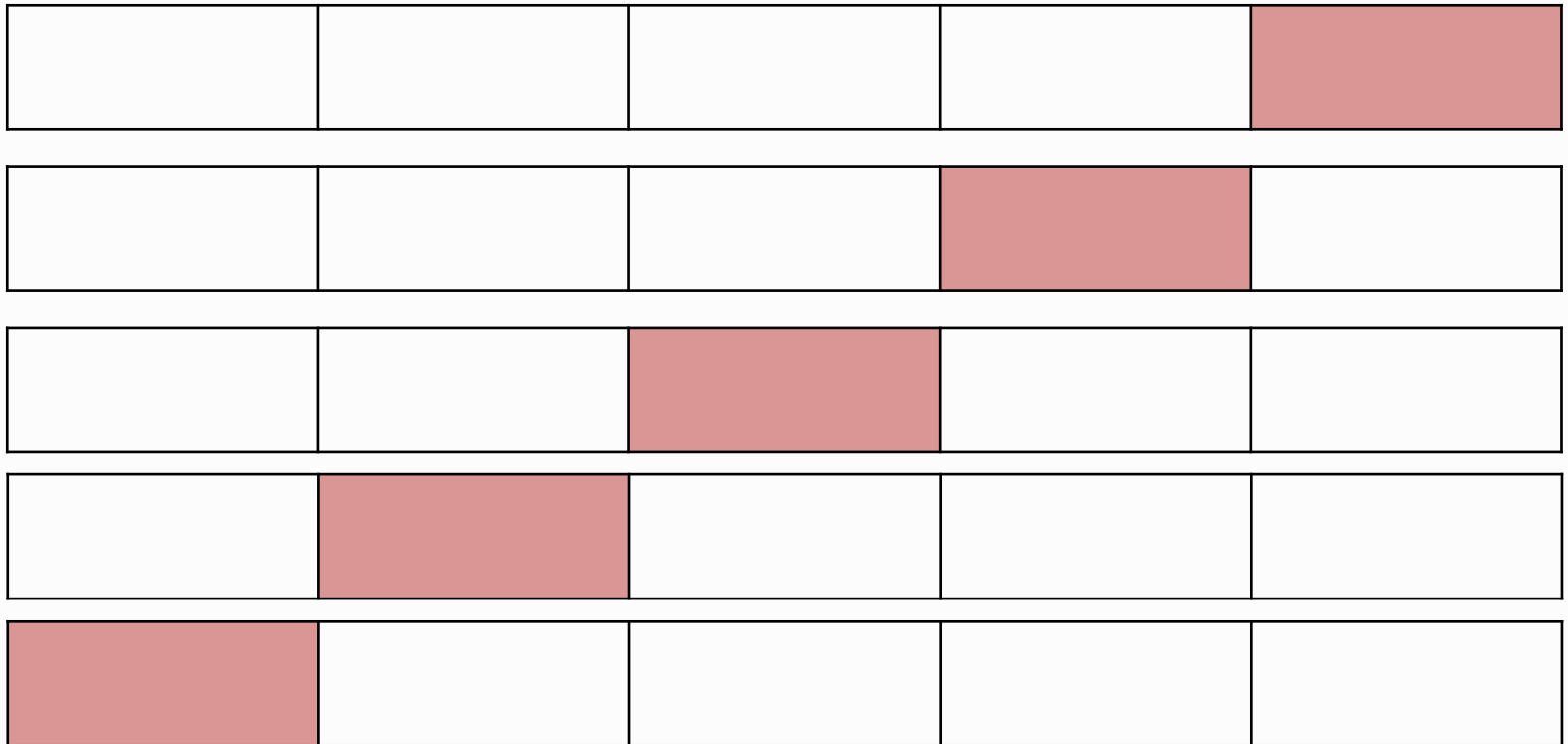| Train, T^t	Test, $T^{\ell-t}$

Solve the optimization problem:

$$\text{CVV}_t = \frac{1}{C_\ell^{\ell-t}} \sum_{T^\ell = T^{\ell-t} \cup T^t} Q\big(\mu(T^t), T^{\ell-t}\big) \to \min$$

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

6

Cross-validation

Split sample to k parts k times

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

7

k-fold cross-validation

k-fold cross-validation

Each of k blocks is a test sample once.

k is usually 10 (5 is small sample size).

Split $T^\ell = F_1 \cup \cdots \cup F_k$, $|F_i| \approx \dfrac{\ell}{k}$.

Solve the optimization problem:

$$\mathrm{CV}_k = \frac{1}{k} \sum_{i=1}^{k} Q\big(\mu(T^\ell \backslash F_i), F_i\big) \to \min.$$

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

8

$t \times k$-fold cross-validation

Repeat t times: split sample on k blocks, each of k blocks is a test sample once.

k usually equals 10, t usually equals 10 or less.

Split T^ℓ randomly t times :

$$T^\ell = F_{(1,1)} \cup \cdots \cup F_{(k,1)} = \cdots = F_{(1,t)} \cup \cdots \cup F_{(k,t)},$$

$$\left| F_{(i,j)} \right| \approx \frac{\ell}{k}.$$
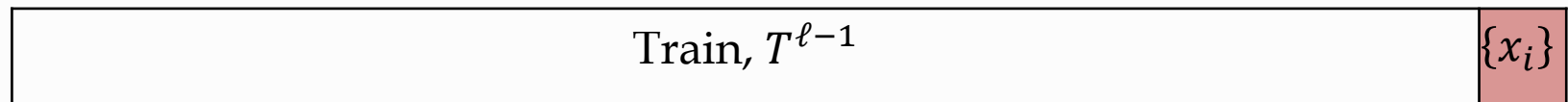
Solve the optimization problem:

$$\mathrm{CV}_{t \times k} = \frac{1}{tk} \sum_{j=1}^{t} \sum_{i=1}^{k} Q\left(\mu\left(T^\ell \backslash F_{(i,j)}\right), F_{(i,j)}\right) \to \min.$$

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

9

Leave one out

Leave-one-out cross-validation, LOO

Split sample into $\ell - 1$ and 1 objects ℓ times.

Train, $T^{\ell-1}$	$\{x_i\}$

Solve the optimization problem:

$$\text{LOO} = \frac{1}{\ell} \sum_{i=1}^{\ell} Q\big(\mu(T^\ell \backslash p_i), p_i\big) \to \min.$$

where $p_i = (x_i, y_i)$.

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

10

Lecture plan

- Validation techniques
- **Similarity-based classification and regression**
- One nearest neighbor
- kNN and hyperparameter tuning
- Generalized distance-based classifiers
- Nonparametric regression
- Prototype selection
- Performance measures

Problem formulation

X is an object set, Y is an answer set,

$y : X \rightarrow Y$ is an unknown dependency, $|Y| \ll \infty$

$X^\ell = \{x_1, \dots, x_n\}$ is a training sample,

$T^\ell = \{(x_1, y_1), \dots, (x_\ell, y_\ell)\}$ is a set of examples.

Task: return an algorithm $a : X \rightarrow Y$.

What is this task?

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

12

Classification problem formulation

X is an object set, Y is an answer set,

$y : X \rightarrow Y$ is an unknown dependency, $|Y| \ll \infty$

$X^{\ell} = \{x_1, \ldots, x_n\}$ is a training sample,

$T^{\ell} = \{(x_1, y_1), \ldots, (x_{\ell}, y_{\ell})\}$ is a set of examples.

Task: return an algorithm $a : X \rightarrow Y$.

What is this task?

Classification, because $|Y| \ll \infty$.

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

13

Duck test:

If it looks like a duck, swims like a duck, and quacks like a duck, then it probably is a duck.

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

14

Duck test

Duck test: *if it looks like a duck, swims like a duck, and quacks like a duck, then it probably is a duck.*

	Looks	Swims	Quacks	A duck?
	like a duck	like a duck	like a duck	**Probably, a duck**
	totally not like a duck	**can be a duck**	**not like a duck**	**Probably, not a duck**

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

15

How is the classifier formalized?

What is the training sample?

Many ducks, many non-ducks (unducks).

What is classification procedure?

1. Ducks were described with **key features**.

2. **Similarity concept** was used.

3. Logical separator was used for classification.

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

16

Main idea

Key hypothesis: similar objects belong to same class / may have similar target function values.

Main idea: given an object, we should find a class, objects of which are the most similar to the given one.

- Reasoning by analogy (case-based)
- Lazy learning

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

17

Lecture plan

- Validation techniques
- Similarity-based classification and regression
- **One nearest neighbor**
- kNN and hyperparameter tuning
- Generalized distance-based classifiers
- Nonparametric regression
- Prototype selection
- Performance measures

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

18

Formalization of "similarity"

"Similarity" is a distance between objects. We will talk about **metrics**.

Distance: $\rho: X \times X \to [0; +\infty)$.

Metric space is a set with a metric $\rho(x, y)$, defined on it.

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

19

Commonly used metrics

Minkowski distance:

$$p(x, y) = \left(\sum_i |x_i - y_i|^p \right)^{\frac{1}{p}},$$

when $p = 2$, it is the Euclidian distance;
when $p = 1$, it is the Manhattan distance.

Mahalanobis distance:

$$p(x, y) = \sqrt{(x - y)^\top S^{-1} (x - y)},$$

where S is covariance matrix for x and y.

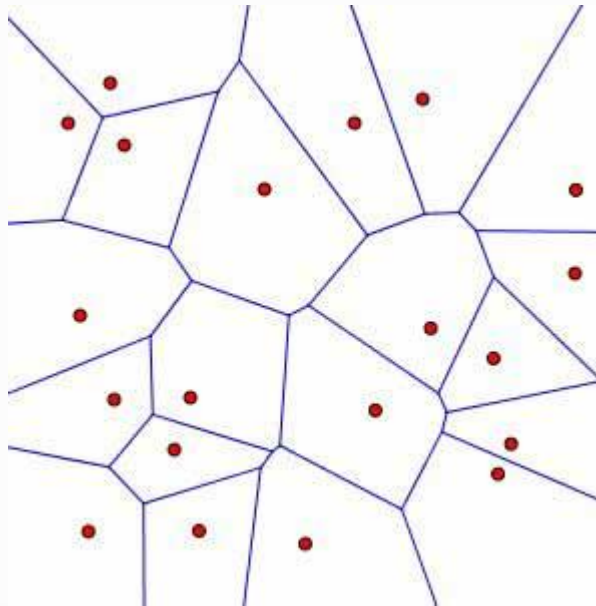Machine learning. Lecture 2. Non-parametric. 13.09.2019.

20

Nearest neighbor method (1NN)

$x_{(u,1)}$ is **nearest neighbor of** u: $x_{(u,1)} = \text{argmin}_{x \in X^\ell} \rho(u, x)$.

Classifier:

$$a\left(u, T^\ell\right) = y\left(x_{(u,1)}\right).$$

Voronoi diagram:

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

21

1NN discussion

Advantages:
- simplicity;
- lucidity;
- interpretability.

Disadvantage:
- sensibility to noise;
- low efficacy;
- no parameters (explicitly);
- necessity to store all the examples.

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

22

Could you suggest 1NN regressor?

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

23

1NN for regression

Could you suggest 1NN regressor?

Assign a value that the nearest neighbor has

Is this somehow useful?

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

24

1NN for regression

Could you suggest 1NN regressor?

Assign a value that the nearest neighbor has

Is this somehow useful?

It is not so bad as it could be, but none uses it

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

25

Lecture plan

- Validation techniques
- Similarity-based classification and regression
- One nearest neighbor
- kNN and hyperparameter tuning
- Generalized distance-based classifiers
- Nonparametric regression
- Prototype selection
- Performance measures

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

26

kNN

Choose a distance ρ.

Sort objects:
$$\rho\big(u, x_{(u,1)}\big) \leq \rho\big(u, x_{(u,2)}\big) \leq \cdots \leq \rho\big(u, x_{(u,\ell)}\big).$$

Algorithm kNN:

$$a\big(u; T^\ell\big) = \text{argmax}_{y \in Y} \sum_{i=1}^{\ell} \big[y(x_{(u,i)}) = y\big]\big[i \leq k\big],$$

$$a\big(u; T^\ell\big) = \text{argmax}_{y \in Y} \sum_{i=1}^{k} \big[y(x_{(u,i)}) = y\big].$$

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

27

Hyperparameters and parameters

We can work with any value of k.

If an algorithm can work with any value of k but cannot answer a question, which value of k is better, k is a **hyperparameter** for this algorithm.

If it can choose best value of k, then
k is a **parameter** of this algorithm.

How to turn k in kNN to be a parameter?

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

28

It is equal to the problem of LOO quality functional minimization:

$$\text{LOO}\big(k, T^\ell\big) = \sum_{i=1}^{\ell} \big[a\big(x_i; T^\ell \backslash \{(x_i, y_i)\}, k\big) \neq y_i\big] \rightarrow$$
$$\rightarrow \min_k$$

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

29

Choosing hyperparameters

You cannot choose hyperparameters on a test set.
Why?

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

30

Choosing hyperparameters

You cannot choose hyperparameters on a test set, because it will lead to overfitting. You will choose the best hyperparameters that will improve performance on this dataset, but not in general.

How to fix it?

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

31

Choosing hyperparameters

You cannot choose hyperparameters on a test set, because it will lead to overfitting. You will choose the best hyperparameters that will improve performance on this dataset, but not in general.

You should split your train set. You will learn parameters on one part, and then tune hyperparameters on another part. Therefore, datasets are typically split to train, validation and test subsets.

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

32

Lecture plan

- Validation techniques
- Similarity-based classification and regression
- One nearest neighbor
- kNN and hyperparameter tuning
- **Generalized distance-based classifiers**
- Nonparametric regression
- Prototype selection
- Performance measures

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

33

How can *k*NN be improved?

- More complicated model (more parameters)

- Distance choosing
- Dimensionality reduction

- Usage of good structures for storing data

- Object set thinning
- Noise filtering
- Prototype selection

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

34

Generalized metric classifier

$$a\left(u; T^{\ell}\right) = \text{argmax}_{y \in Y} \sum_{i=1}^{\ell} \left[y(x_{(u,i)}) = y\right] w_{(i,u)},$$

where $w_{(i,u)}$ is a function representing importance of ith neighbor of u.

$C_y(u) = \sum_{i=1}^{\ell} \left[y(x_{(u,i)}) = y\right] w_{(i,u)}$ is estimation of object u closeness to class y.

$$a\left(u; T^{\ell}\right) = \text{argmax}_{y \in Y} \sum_{i=1}^{\ell} C_y(u).$$

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

35

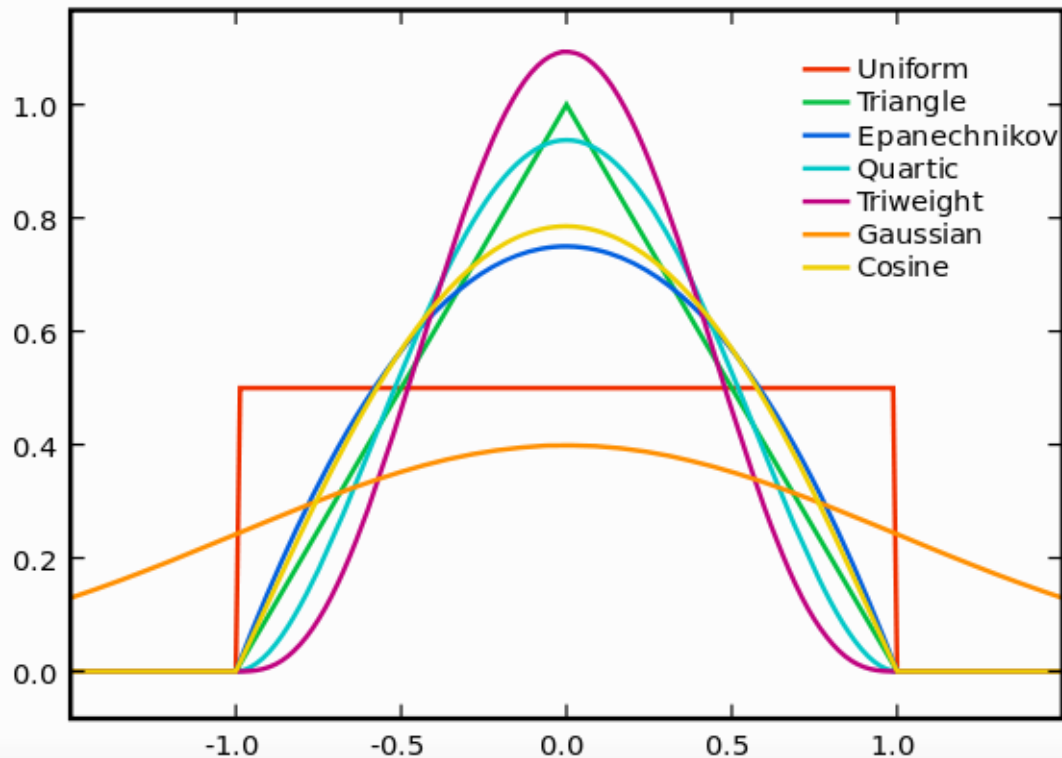# What can be chosen as *w*?

$w_{(i,u)}$:

- linearly decreasing functions;
- exponentially decreasing functions;
- kernel functions.

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

36

Kernel function

Kernel function $K(x)$ is symmetric non-negative function, $\int_{-\infty}^{+\infty} K(x)\,dx = 1$.

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

37

Parzen-Rosenblatt window

With fixed window width:
$$a\left(u; T^{\ell}; {\color{red}h}; K\right) =$$

$$= \mathrm{argmax}_{y \in Y} \sum_{i=1}^{\ell} \left[y(x_{(u,i)}) = y \right] K \left(\frac{\rho(u, x_{(u,i)})}{\color{red}h} \right),$$

With variable window width:
$$a\left(u; T^{\ell}; {\color{red}k}; K\right) =$$

$$= \mathrm{argmax}_{y \in Y} \sum_{i=1}^{\ell} \left[y(x_{(u,i)}) = y \right] K \left(\frac{\rho\left(u, x_{(u,i)}\right)}{\color{red}\rho\left(u, x_{(u,k+1)}\right)} \right).$$

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

38

Distance selection (learning)

Distance can be learned.

Example (weighted Minkowski):

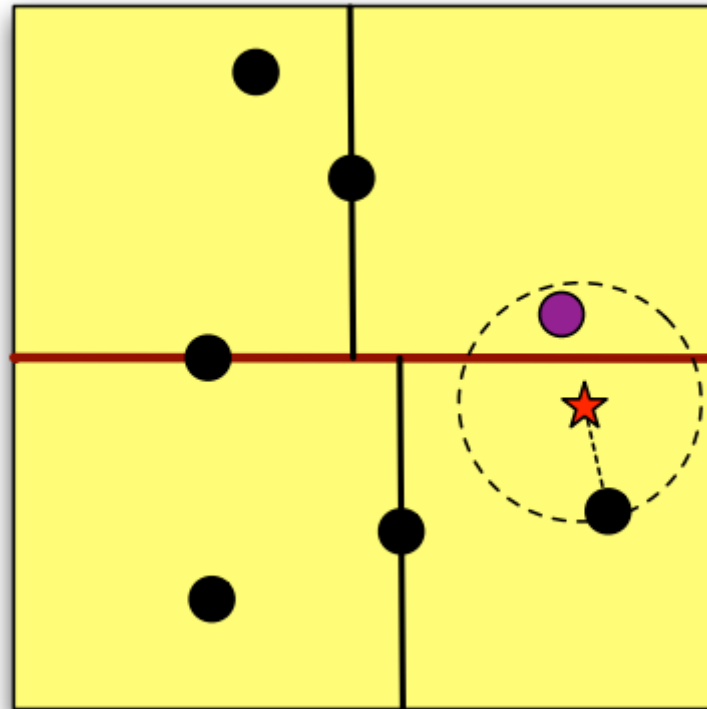$$p(x, y) = \left(\sum_i w_i |x_i - y_i|^p \right)^{\frac{1}{p}}.$$

Now the problem is how to choose coefficients w_i.

When $w_i = 0$, the corresponding feature is thrown away (feature selection).

A kernelization can be applied.

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

39

Structure for storing data

Different greed-like structures can be used. The most effective is *k-d-trees*:

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

40

Lecture plan

- Validation techniques
- Similarity-based classification and regression
- One nearest neighbor
- kNN and hyperparameter tuning
- Generalized distance-based classifiers
- **Nonparametric regression**
- Prototype selection
- Performance measures

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

41

Problem formalization

X is an object set, Y is an answer set,

$y: X \rightarrow Y$ is an unknown dependency, $Y \in \mathbb{R}$

$X^{\ell} = \{x_1, \dots, x_{\ell}\}$ is training sample,

$T^{\ell} = \{(x_1, y_1), \dots, (x_{\ell}, y_{\ell})\}$ set of instances.

Problem: find $a: X \rightarrow Y$.

`

$a(x) = f(x, \theta)$ is dependency model, $\theta \in \mathbb{R}^t$.

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

42
42

Main idea

Basic idea: let think that $\theta(x) = \theta$ nearby $x \in X$:

$$Q\left(\theta, T^{\ell}\right) = \sum_{i=1}^{\ell} w_i(x)(\theta - y_i)^2 \rightarrow \min_{\theta \in \mathbb{R}}.$$

Main idea: use kernel smoothing:

$$w_i(x) = K\left(\frac{\rho(x_i, x)}{h}\right),$$

where h is window width.

Kernel smoothing

Nadaraya-Watson kernel smoothing:

$$a_h\big(x, T^\ell\big) = \frac{\sum_{i=1}^{\ell} y_i w_i(x)}{\sum_{i=1}^{\ell} w_i(x)} = \frac{\sum_{i=1}^{\ell} y_i K\left(\frac{\rho(x_i, x)}{h}\right)}{\sum_{i=1}^{\ell} K\left(\frac{\rho(x_i, x)}{h}\right)}.$$

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

44
44

Basic theorem

Theorem. If

1) sample T^ℓ is simple, distributed with $p(x, y)$;

2) $\int_0^\infty K(r) dr < \infty$, $\lim_{r \to \infty} r K(r) = 0$;

3) $\mathrm{E}(y^2 | x) < \infty \ \forall x \in X$;

4) $\lim_{\ell \to \infty} h_\ell = 0$, $\lim_{\ell \to \infty} \ell h_\ell = \infty$,

then $a_h(x, T^\ell) \to^P \mathrm{E}(y|x)$ in any $x \in X$,

when $E(y|x), p(x), D(y|x)$ are continuing, $p(x) > 0$.

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

45
45

Method discussion

- kernel function has impact on smoothness;
- kernel function has small impact on approximation quality;
- h impacts on approximation quality;
- k can be tuned;
- sensitive to noise.

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

46
46

Lecture plan

- Validation techniques
- Similarity-based classification and regression
- One nearest neighbor
- kNN and hyperparameter tuning
- Generalized distance-based classifiers
- Nonparametric regression
- **Prototype selection**
- Performance measures

Margin

Margin of object x_i with respect to algorithm $a(u)$:

$$M(x_i) = C_{y_i}(x_i) - \max_{y \in Y \setminus \{y_i\}} C_y(x_i).$$

Margin is the measure of object "typicalness" for its class. The higher margin is, the more typical the object is.
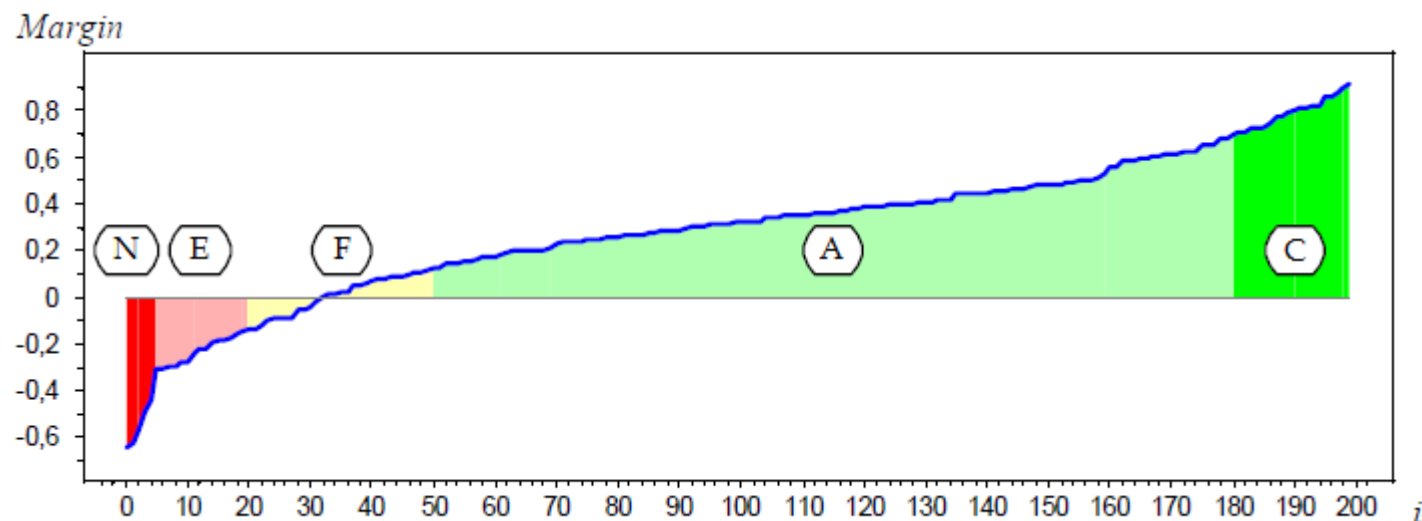
Ranking based on margin

C − **core** (base for classification)

A − **accompaniment** (can be deleted)

F − **frontier** (classification is instable)

E − **erroneous** (misclassification because of bad model)

N − **noisy** (misclassification because of bad data)

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

49

Prototype selection problem

Prototype selection problem is how to choose optimal subset of object $\Omega \subseteq X^{\ell}$.

Distance-based classifier:

$$a(u; \Omega) = \operatorname{argmax}_{y \in Y} \sum_{x_i \in \Omega} \left[y(x_{(u,i)}) = y \right] w_{(u,i)} .$$

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

50

Prototype selection solutions

The problem is NP-hard, so there are lots of approximate solutions.

Two main approaches:

1) linear programming relaxation;

2) greedy algorithms.

Plenty of heuristic approaches.

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

51

Lecture plan

- Validation techniques
- Similarity-based classification and regression
- One nearest neighbor
- kNN and hyperparameter tuning
- Generalized distance-based classifiers
- Nonparametric regression
- Prototype selection
- **Performance measures**

Contingency table

	Positive	Negative
Classified as positive	**TP** = True Positive	**FP** = False Positive
Classified as negative	**FN** = False Negative	**TN** = True Negative

FN in math. stat. is I type error

FP in math. stat. is II type error

P = **TP** + **FN** is the number of positive examples

N = **FP** + **TN** is the number of negative examples

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

53
53

Some definitions

Sensitivity or **Recall**:

$$\text{Recall} = \text{TPR} = \frac{\text{TP}}{\text{P}}$$

Specificity:

$$\text{SPC} = \frac{\text{TN}}{\text{N}}$$

Precision:

$$\text{Precision} = \text{PPV} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

Accuracy:

$$\text{Accuracy} = \text{ACC} = \frac{\text{TP} + \text{TN}}{\text{P} + \text{N}}$$

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

54
54

F-measure

We will not lose much in accuracy performing badly on small classes.

F_β-measure:

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{Precision} \cdot \text{Recall}}{\beta^2 \cdot \text{Precision} + \text{Recall}}$$
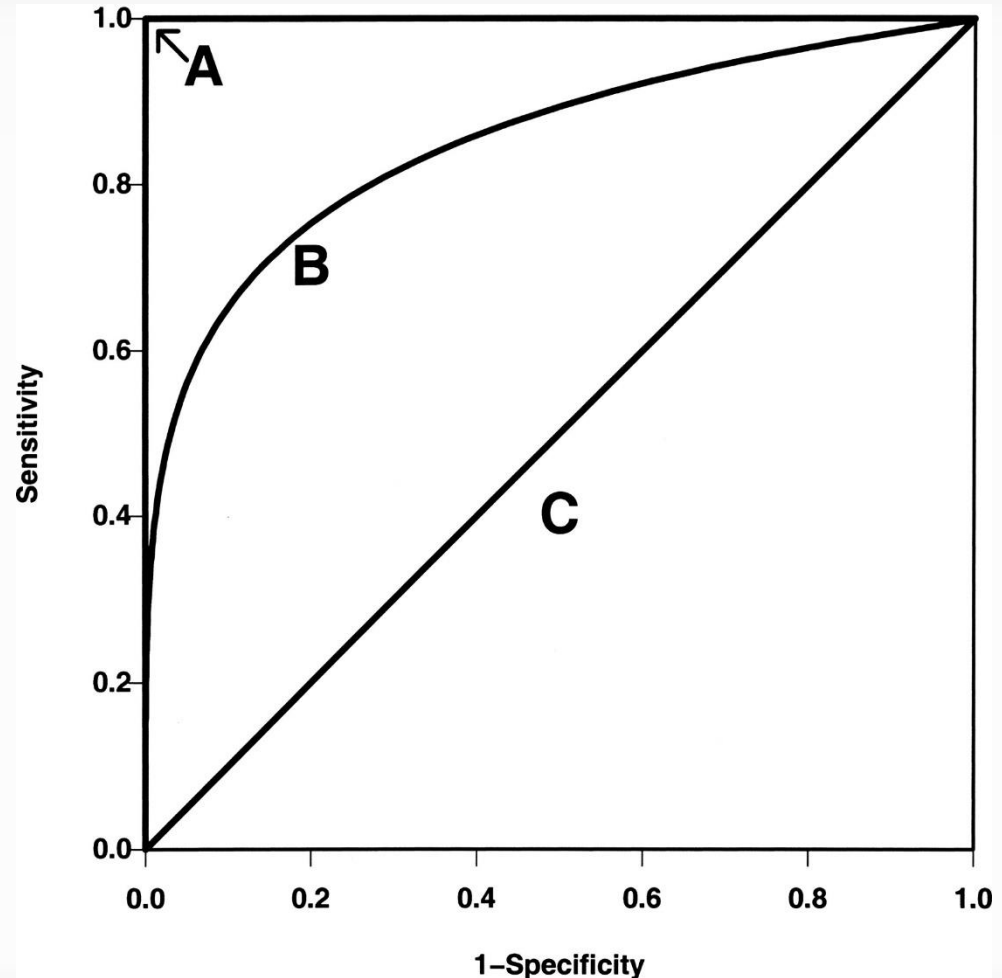
F_1-measure:

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

55
55

ROC-curve

A is the best algorithm

B is a typical algorithm

C is the worst algorithm

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

56
56

AUC

Area under the curve (AUC) is area under the ROC-curve.

Connected with Mann-Whitney U.

Can be expressed with Gini-index.

Out of date measure. However, it is widespread, and people still use it.

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

57
57

Multiclass case

- One vs one classification
- One vs all (one vs rest) classification
- Hierarchical classification
- Confusion matrix

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

58
58

Usual regression errors

- Root mean squared error (RMSE)
- Mean absolute error (MAE)
- Mean squared error (MSE)
- Symmetric mean absolute percentage error (SMAPE):

$$\text{SMAPE} = \frac{1}{\ell} \sum_{i=1}^{\ell} \frac{2 \cdot |a(x_i) - y_i|}{|a(x_i)| + |y_i|}.$$

Machine learning. Lecture 2. Non-parametric. 13.09.2019.

59
59