

Introduction to machine learning

Victor Kitov

v.v.kitov@yandex.ru



Course information

- Instructor - Victor Vladimirovich Kitov
- Tasks of the course
- Structure:
 - lectures, seminars
 - assignments: theoretical, labs, competitions
 - exam
- Tools
 - python
 - ipython notebook
 - numpy, scipy, pandas
 - matplotlib, seaborn
 - scikit-learn.

Recommended materials

- **The Elements of Statistical Learning: Data Mining, Inference, and Prediction.** Trevor Hastie, Robert Tibshirani, Jerome Friedman, 2nd Edition, Springer, 2009.
- **Data Mining: The Textbook.** Charu C. Aggarwal, Springer, 2015.
- **Statistical Pattern Recognition.** 3rd Edition, Andrew R. Webb, Keith D. Copsey, John Wiley & Sons Ltd., 2011.
- Vorontsov's SHAD video lectures (Russian).
- Vorontsov's textual lectures (Russian).
- Any additional public sources:
 - wikipedia, articles, tutorials, video-lectures.
- Practical questions:
 - [StackOverflow](#), [scikit-learn documentation](#), [kaggle forum](#).

Table of Contents

- 1 Problem statement
- 2 Visual examples
- 3 Function class
- 4 Function estimation
- 5 Notation used in the course

Formal definitions of machine learning

- Machine learning is a field of study that gives computers the ability to learn without being explicitly programmed.
- A computer program is said to learn from **experience E** with respect to some **class of tasks T** and **performance measure P**, if its performance P at tasks in T improves with experience E .

Examples

- Spam filtering
 - if sender belongs to black-list -> spam
 - if contains phrase 'buy now' and sender is unknown -> spam
 - ...
- Part-of-speech tagger.
 - if ends with 'ed' -> verb
 - if previous word is 'the' -> noun
 - ...
- ML finds decision rules automatically with labelled data!

Formal problem statement

- Set of objects O
- Each object is described by a vector of known characteristics $\mathbf{x} \in \mathcal{X}$ and predicted characteristics $y \in \mathcal{Y}$.

$$o \in O \longrightarrow (\mathbf{x}, y)$$

- Task: find a mapping f , which could accurately approximate $\mathcal{X} \rightarrow \mathcal{Y}$.
 - using a finite **known set** of objects.
 - apply model for objects from the **test set**.
- test set way be known or not.

Specification of known/test sets

Known set:

- **supervised learning:** $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots (\mathbf{x}_N, y_N)$
 - e.g. regression, classification.
- **unsupervised learning:** $\mathbf{x}_1, \mathbf{x}_2, \dots \mathbf{x}_N$ -
 - e.g. dimensionality reduction, clustering, outlier analysis
- **semi-supervised learning:**
 $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots (\mathbf{x}_N, y_N), \mathbf{x}_{N+1}, \mathbf{x}_{N+2}, \dots \mathbf{x}_{N+M}$

If test set objects $\mathbf{x}'_1, \mathbf{x}'_2, \dots \mathbf{x}'_K$ are known in advance, then this is **transductive learning**.

Reinforcement learning

- **Reinforcement learning** setup:
 - a set of environment and agent states S ;
 - a set of actions A , of the agent
 - $P(s_{t+1} = s' | s_t = s, a_t = a)$ is the probability of transition from state s to state s' under action a .
 - $R_a(s, s')$ is the (expected) immediate reward after transition from s to s' with action a .
 - rules that describe what the agent observes
 - full / partial observability
- Well-suited to problems which include a long-term versus short-term reward trade-off
- Applications: robot control, elevator scheduling, games (chess, go), etc.

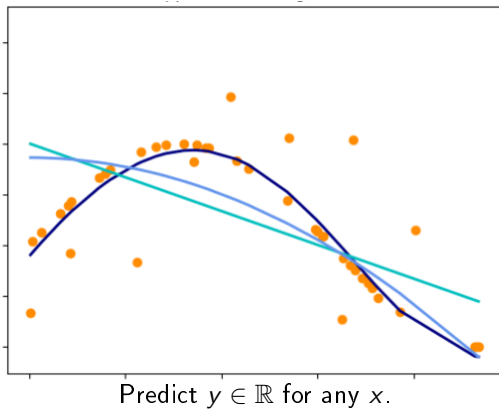
Table of Contents

- 1 Problem statement
- 2 Visual examples
 - Supervised learning
 - Unsupervised learning
- 3 Function class
- 4 Function estimation
- 5 Notation used in the course

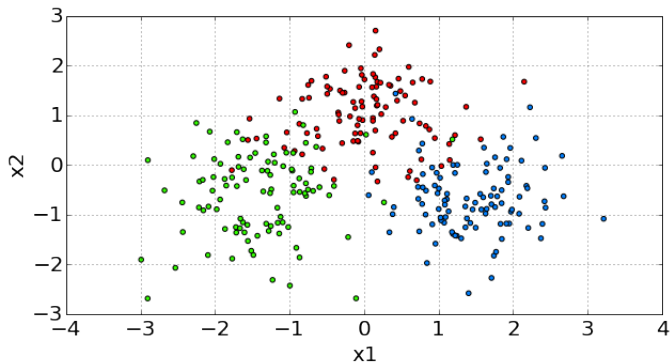
2 Visual examples

- Supervised learning
- Unsupervised learning

Regression

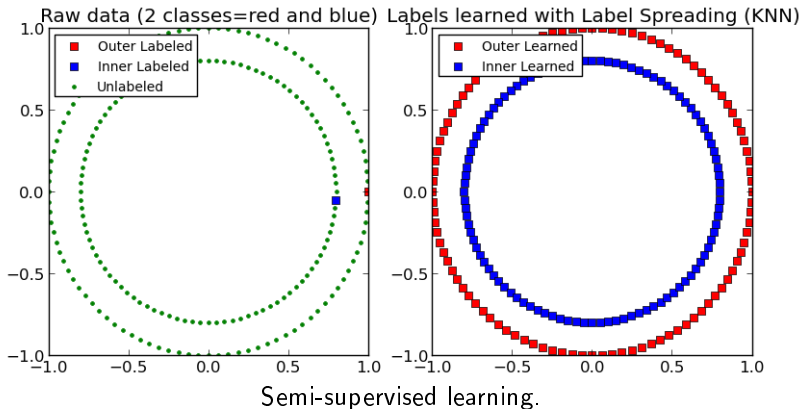


Classification



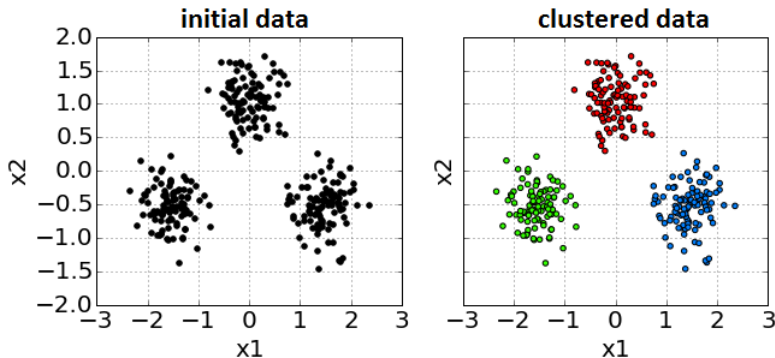
Predict class y shown with color for any point.

Semi-supervised classification



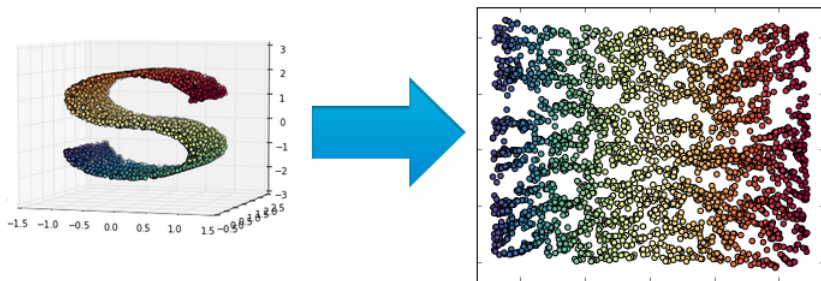
- 2 Visual examples
 - Supervised learning
 - Unsupervised learning

Clustering



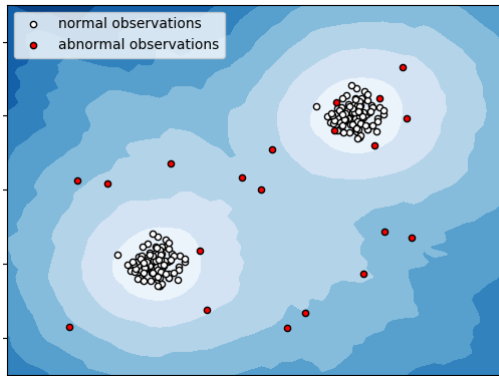
Cluster points into distinct similarity groups.

Dimensionality reduction



Reduce dimension from 3D to 2D with minimal distortion.

Outlier detection



Detect untypical observations.

General problem statement

- We want to find $f(x) : X \rightarrow Y$.
- How it may be used:
 - prediction of Y
 - qualitative analysis, understanding of $X \rightarrow Y$ dependency
 - untypical objects detection (where model fails)
- Questions solved in ML:
 - what target y we are predicting?
 - how to select object descriptors (features) x ?
 - what is the kind of mapping f ?
 - in what sense a mapping f should approximate true relationship?
 - how to tune f ?

Types of target variable (supervised learning)²

- $\mathcal{Y} = \mathbb{R}$ - regression
 - e.g. flat price
- $\mathcal{Y} = \mathbb{R}^M$ - vector regression
 - e.g. stock price dynamics
- $\mathcal{Y} = \{\omega_1, \omega_2, \dots, \omega_C\}$ - classification.
 - $C=2$: binary classification.
 - e.g. spam / not spam
 - $C>2$: multi-class classification
 - e.g. identity recognition, activity recognition
- \mathcal{Y} - set of all sets of $\{\omega_1, \omega_2, \dots, \omega_C\}$ - labeling.¹
 - e.g. news categorization

¹How to solve labeling using classification?

²Actually any type is possible. Listed are most common types.

Types of features³

- Full object description $\mathbf{x} \in \mathcal{X}$ consists of individual features $x^i \in \mathcal{X}_i$
- Types of feature (e.g. for credit scoring):
 - $\mathcal{X}_i = \{0, 1\}$ - binary feature
 - e.g. marital status
 - $|\mathcal{X}_i| < \infty$ - categorical (nominal) feature
 - e.g. occupation
 - $|\mathcal{X}_i| < \infty$ and \mathcal{X}_i is ordered - ordinal feature
 - e.g. education level
 - $\mathcal{X}_i = \mathbb{R}$ - real feature
 - e.g. age

³Actually any type is possible. Listed are most common types.

Table of Contents

- 1 Problem statement
- 2 Visual examples
- 3 Function class**
- 4 Function estimation
- 5 Notation used in the course

Function class. Linear example.

- **Function class** - parametrized set of functions
 $F = \{f_\theta, \theta \in \Theta\}$, from which the true relationship $\mathcal{X} \rightarrow \mathcal{Y}$ is approximated.

⁴Are discriminant functions uniquely defined for fixed mapping $X \rightarrow Y$?

Function class. Linear example.

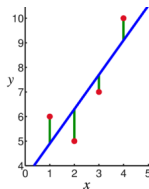
- **Function class** - parametrized set of functions
 $F = \{f_\theta, \theta \in \Theta\}$, from which the true relationship $\mathcal{X} \rightarrow \mathcal{Y}$ is approximated.
- **Regression**: $\hat{y} = f(x|\theta)$,
- **Classification**: $\hat{y} = f(x|\theta) = \arg \max_c \{g_c(x|\theta)\}$,
 $c = 1, 2, \dots, C$.
 - $c = 1, 2, \dots, C$: possible classes, $g_c(x)$ - score of class c , given x called *discriminant function*⁴.

⁴Are discriminant functions uniquely defined for fixed mapping $X \rightarrow Y$?

Examples

linear regression $y \in \mathbb{R}$:

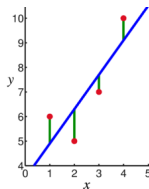
$$f(x|\theta) = \theta_0 + \theta_1 x$$



Examples

linear regression $y \in \mathbb{R}$:

$$f(x|\theta) = \theta_0 + \theta_1 x$$



linear classification $y \in \{1, 2\}$:

$$g_c(\mathbf{x}|\theta) = \theta_c^0 + \theta_c^1 x^1 + \theta_c^2 x^2, \quad c = 1, 2.$$

$$f(\mathbf{x}|\theta) = \arg \max_c g_c(\mathbf{x}|\theta)$$

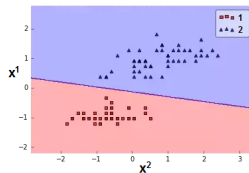


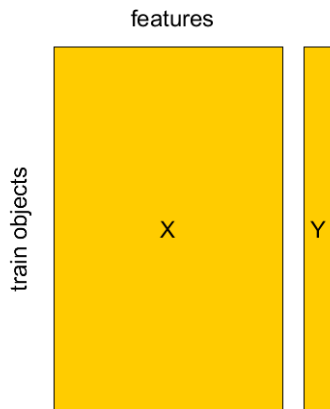
Table of Contents

- 1 Problem statement
- 2 Visual examples
- 3 Function class
- 4 Function estimation**
 - Separate validation set
 - Cross-validation
 - A/B testing
- 5 Notation used in the course

Known set

Known set: $(\mathbf{x}_1, y_1), \dots (\mathbf{x}_M, y_M)$,

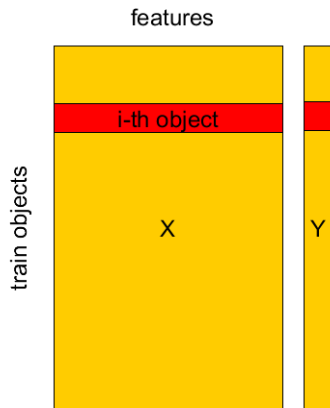
design matrix $X = [\mathbf{x}_1, \dots \mathbf{x}_M]^T$, $Y = [y_1, \dots y_M]^T$.



Known set

Known set: $(\mathbf{x}_1, y_1), \dots (\mathbf{x}_M, y_M)$,

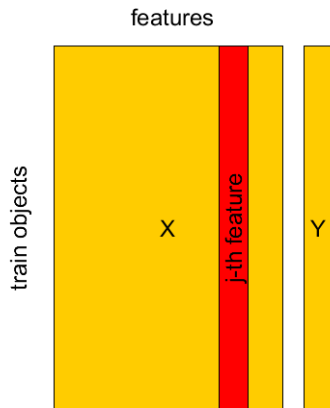
design matrix $X = [\mathbf{x}_1, \dots \mathbf{x}_M]^T$, $Y = [y_1, \dots y_M]^T$.



Known set

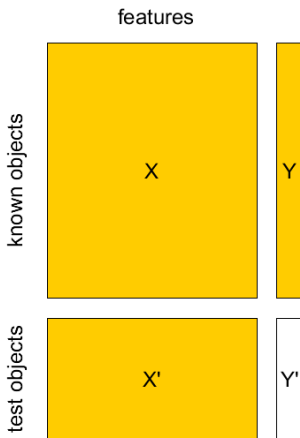
Known set: $(\mathbf{x}_1, y_1), \dots (\mathbf{x}_M, y_M)$,

design matrix $X = [\mathbf{x}_1, \dots \mathbf{x}_M]^T$, $Y = [y_1, \dots y_M]^T$.



Known set, test set

- Known sample $X, Y: (\mathbf{x}_1, y_1), \dots, (\mathbf{x}_M, y_M)$
- Test sample $X', Y': (\mathbf{x}'_1, y'_1), \dots, (\mathbf{x}'_K, y'_K)$



Score versus loss

- In machine learning predictions, functions, objects can be assigned:
 - **score, rating** - this should be maximized
 - **loss, cost** - this should be minimized⁵

⁵how can one convert score \leftrightarrow loss?

Loss function $\mathcal{L}(\hat{y}, y)$ ⁶

- Examples:
 - **classification:**
 - misclassification rate

$$\mathcal{L}(\hat{y}, y) = \mathbb{I}[\hat{y} \neq y]$$

- **regression:**
 - MAE (mean absolute error):

$$\mathcal{L}(\hat{y}, y) = |\hat{y} - y|$$

- MSE (mean squared error):

$$\mathcal{L}(\hat{y}, y) = (\hat{y} - y)^2$$

⁶Selecting realistic loss is not trivial. Consider e.g. demand forecasting.

Empirical risk

- Want to minimize *expected risk*:

$$\int \int \mathcal{L}(f_{\theta}(\mathbf{x}), y) p(\mathbf{x}, y) d\mathbf{x} dy \rightarrow \min_{\theta}$$

⁷We assume that objects are i.i.d.

Empirical risk

- Want to minimize *expected risk*:

$$\int \int \mathcal{L}(f_{\theta}(\mathbf{x}), y) p(\mathbf{x}, y) d\mathbf{x} dy \rightarrow \min_{\theta}$$

- Can minimize only *empirical risk*⁷:

$$L(\theta|X, Y) = \frac{1}{N} \sum_{n=1}^N \mathcal{L}(f_{\theta}(\mathbf{x}_n), y_n)$$

- Method of empirical risk minimization:

$$\hat{\theta} = \arg \min_{\theta} L(\theta|X, Y)$$

⁷We assume that objects are i.i.d.

Estimation of empirical risk

- What is the relationship between $L(\hat{\theta}|X, Y)$ and $L(\hat{\theta}|X', Y')$?

Estimation of empirical risk

- What is the relationship between $L(\hat{\theta}|X, Y)$ and $L(\hat{\theta}|X', Y')$?
- Typically

$$L(\hat{\theta}|X, Y) < L(\hat{\theta}|X', Y')$$

- How to get realistic estimate of $L(\hat{\theta}|X', Y')$?

Estimation of empirical risk

- What is the relationship between $L(\hat{\theta}|X, Y)$ and $L(\hat{\theta}|X', Y')$?
- Typically

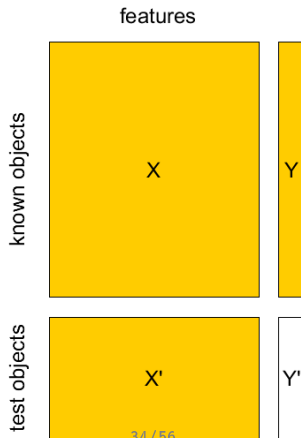
$$L(\hat{\theta}|X, Y) < L(\hat{\theta}|X', Y')$$

- How to get realistic estimate of $L(\hat{\theta}|X', Y')$?
 - separate **validation set**
 - **cross-validation**
 - **leave-one-out** method

- 4 Function estimation
 - Separate validation set
 - Cross-validation
 - A/B testing

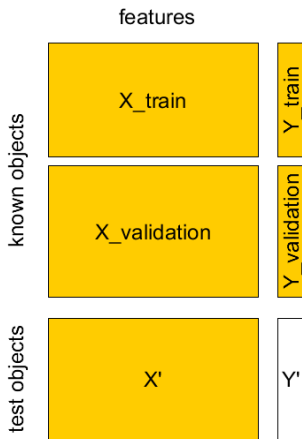
Separate validation set

- Known sample $X, Y: (\mathbf{x}_1, y_1), \dots, (\mathbf{x}_M, y_M)$
- Test sample $X', Y': (\mathbf{x}'_1, y'_1), \dots, (\mathbf{x}'_K, y'_K)$



Separate validation set

Divide known set randomly or randomly with stratification:



- 4 Function estimation
 - Separate validation set
 - Cross-validation
 - A/B testing

4-fold cross-validation example

X	Y
1	1
2	2
3	3
4	4

Divide training set into K parts, referred as «folds» (here $K = 4$).

Variants:

- randomly
- randomly with stratification (w.r.t target value or feature value).

4-fold cross validation example

X	Y
1	1
2	2
3	3
4	4

Use folds 1,2,3 for model estimation and fold 4 for model evaluation.

4-fold cross validation example

X	Y
1	1
2	2
3	3
4	4

Use folds 1,2,4 for model estimation and fold 3 for model evaluation.

4-fold cross validation example

X	Y
1	1
2	2
3	3
4	4

Use folds 1,3,4 for model estimation and fold 2 for model evaluation.

4-fold cross validation example

X	Y
1	1
2	2
3	3
4	4

Use folds 2,3,4 for model estimation and fold 1 for model evaluation.

4-fold cross validation example

- Denote
 - $k(n)$ - fold to which observation (\mathbf{x}_n, y_n) belongs to: $n \in I_k$.
 - $\hat{\theta}^{-k}$ - parameter estimation using observations from all folds except fold k .

⁸will samples be correlated?

4-fold cross validation example

- Denote
 - $k(n)$ - fold to which observation (\mathbf{x}_n, y_n) belongs to: $n \in I_k$.
 - $\hat{\theta}^{-k}$ - parameter estimation using observations from all folds except fold k .

Cross-validation empirical risk estimation

$$\hat{L}_{total} = \frac{1}{N} \sum_{n=1}^N \mathcal{L}(f_{\hat{\theta}^{-k(n)}}(\mathbf{x}_n), y_n)$$

⁸will samples be correlated?

4-fold cross validation example

- Denote
 - $k(n)$ - fold to which observation (\mathbf{x}_n, y_n) belongs to: $n \in I_k$.
 - $\hat{\theta}^{-k}$ - parameter estimation using observations from all folds except fold k .

Cross-validation empirical risk estimation

$$\hat{L}_{total} = \frac{1}{N} \sum_{n=1}^N \mathcal{L}(f_{\hat{\theta}^{-k(n)}}(\mathbf{x}_n), y_n)$$

- For K -fold CV we have:
 - K parameters $\hat{\theta}^{-1}, \dots, \hat{\theta}^{-K}$
 - K models $f_{\hat{\theta}^{-1}}(\mathbf{x}), \dots, f_{\hat{\theta}^{-K}}(\mathbf{x})$.
 - can use ensembles
 - K estimations of empirical risk:

$$\hat{L}_k = \frac{1}{|I_k|} \sum_{n \in I_k} \mathcal{L}(f_{\hat{\theta}^{-k}}(\mathbf{x}_n), y_n), \quad k = 1, 2, \dots, K.$$
 - can estimate variance & use statistics!⁸

⁸will samples be correlated?

Comments on cross-validation

- When number of folds K is equal to number of objects N , this is called **leave-one-out method**.
- Cross-validation uses the i.i.d.⁹ property of observations
- Stratification by target y helps for imbalanced/rare classes.

⁹i.i.d.=independent and identically distributed

- 4 Function estimation
 - Separate validation set
 - Cross-validation
 - A/B testing

A/B testing

- Observe test set **after the models were built.**
- A/B testing procedure:
 - 1 divide test objects randomly into two groups - A and B.
 - 2 apply base model to A
 - 3 apply modified model to B
 - 4 compare final results

A/B testing

- Observe test set **after the models were built.**
- A/B testing procedure:
 - 1 divide test objects randomly into two groups - A and B.
 - 2 apply base model to A
 - 3 apply modified model to B
 - 4 compare final results

Cross-validation vs. A/B testing

Comparison of cross-validation and A/B test:

	cross-validation	A/B test
realism	use retrospective analysis, rely on i.i.d. assumption	full realism
overfitting	possible (when use it multiple times)	almost impossible (possible if A/B split is inadequate)
costs	uses available data, only computational costs	requires time and resources for collecting & evaluating feedback from objects of groups A and B

- When forecast affects true outcome (e.g. in recommender system) A/B test is more adequate.

General modelling pipeline¹⁰

- Understand business problem
- Problem formalization
- Data collection
- **Data preprocessing**
- **Modelling**
- **Model evaluation**
- Deployment
- Maintenance

¹⁰Bold - steps covered in kaggle competitions.

Major niches of ML

- hard to formulate explicit rules
 - complex inter-relationships
 - e.g. image recognition
 - too many attributes
 - e.g. text categorization
- fine-tuning performance on huge datasets
 - e.g. threshold for credibility in credit scoring
- fast adaptation to changing conditions
 - e.g. stock prices/volatility prediction
- further adaptation to usage conditions is required
 - e.g. voice detection

Examples of ML applications by domain

- WEB
 - Web-page ranking
 - Spam filtering
 - e-mails, web pages in search results
- Computer networks
 - Authentication systems
 - by voice, face, fingerprint
 - by behavior
 - Intrusion detection
- Business
 - Fraud detection
 - Churn prediction
- Banking
 - Credit scoring
 - Stock prices forecasting
 - Risks estimation

Examples of ML applications by data type

- Texts
 - Document classification
 - POS tagging, semantic parsing,
 - named entities detection
 - sentimental analysis
 - automatic summarization
- Images
 - Handwriting recognition
 - Face detection, pose detection
 - Person identification
 - Image classification
 - Image segmentation
 - Adding artistic style
- Other
 - Target detection / classification
 - Particle classification

Connection of ML with other fields

- Pattern recognition
 - recognize patterns and regularities in the data
- Computer science
- Artificial intelligence
 - create devices capable of intelligent behavior
- Time-series analysis
- Theory of probability, statistics
 - when relies upon probabilistic models
- Optimization methods
- Theory of algorithms

Table of Contents

- 1 Problem statement
- 2 Visual examples
- 3 Function class
- 4 Function estimation
- 5 Notation used in the course**

Notation used in the course¹¹

- **Objects and outputs:**

- x - vector of known input characteristics of an object
- y - predicted target characteristics of an object specified by x
- x_i - i -th object of a set, y_i - corresponding target characteristic
- x^k - k -th feature of object specified by x
- x_i^k - k -th feature of object specified by x_i

- **General definitions:**

- D - dimensionality of the feature space: $x \in \mathbb{R}^D$
- N - the number of objects in the training set
- C - total number of classes in classification.
- Possible classes: $\{1, 2, \dots, C\}$ or $\{\omega_1, \omega_2, \dots, \omega_C\}$

¹¹If this corresponds the context and there are no redefinitions

Notation used in the course

- **Training set:**

- X - design matrix, $X \in \mathbb{R}^{N \times D}$
- $Y \in \mathbb{R}^N$ - target characteristics of a training set

- **Optimization:**

- $\mathcal{L}(\hat{y}, y)$ - loss function for 1 object
 - y is the true value and \hat{y} is the predicted value.
- $L(\theta) = \sum_{n=1}^N \mathcal{L}(f_{\theta}(x_n), y_n)$ loss function for the whole the training set.

Notation used in the course

- **Special functions:**

- $[x]_+ = \max\{x, 0\}$
- $\mathbb{I}[\text{condition}] = \begin{cases} 1, & \text{if condition is satisfied} \\ 0, & \text{if condition is not satisfied} \end{cases}$
- $\text{sign}(x) = \begin{cases} 1, & x \geq 0 \\ -1, & x < 0 \end{cases}$

- **Other definitions:**

- \hat{z} defines an estimate of z , based on the training set: for example, $\hat{\theta}$ is the estimate of θ , \hat{y} is the estimate of y , etc.
- r.v.=random variable, w.r.t.=with respect to, e.g.=for example.
- $A \succcurlyeq 0$ means that A is a square positive semi-definite matrix.
- All vectors are vectors-columns, e.g. if $x \in \mathbb{R}^D$ its dimensions are $D \times 1$.

Summary

- Machine learning algorithms reconstruct relationship between features x and outputs y .
- Relationship is reconstructed by optimal function $\hat{y} = \hat{f}_{\hat{\theta}}(x)$ from function class $\{f_{\theta}(x), \theta \in \Theta\}$.
- θ is particular controls model complexity, models may be too simple and too complex.
- $\hat{\theta}$ selected to minimize empirical risk $\frac{1}{N} \sum_{n=1}^N \mathcal{L}(f_{\theta}(x_n), y_n)$ for some loss function $\mathcal{L}(\hat{y}, y)$.
- Overfitting - non-realistic estimate of expected loss on the training set.
- To avoid overfitting - use validation sets, cross-validation, A/B test.