Machine Learning and Data Mining

Machine Learning, lecture

Maxim Borisyak

September 15, 2016

National Research University
Higher School of Economics

Before we start

Thank you for participating in the survey!

Deep Learning by Alexander Panin (Fedor Ratnikov)?

Machine Learning

What for?

Machine Learning - approximation of algorithms:

- to solve hard problems:
  - complex distributions:
    - physics, chemistry, ...;
  - unknown distributions:
    - cat images;
- to speed up algorithms:
  - also physics, chemistry;
  - games (AlphaGO).

What for?

How it is different from statistics or Computer Science?

Unlike statistics:

- we usually don't care much about underlying distributions,
- only about solution.

Unlike Computer Science:

- we usually learn from real data;
- all improvements and speeding up comes from adjusting to data.

Machine Learning problems

Machine Learning problem consists of:

- set of samples: $x \in X$:
    - for benchmark problems it is usually split into *training* and *test* sets;
- targets: $y \in Y$;
- quality metric: $Q : \mathcal{A} \times X \times Y \to \mathbb{R}$
- restrictions:
    - by speed,
    - by scalability,
    - invariance to some parameters,
    - etc.

Machine Learning algorithms generic recipe

Machine Learning algorithm:

- model - a set of algorithms:

$$\mathcal{A} \subseteq \{A : X \to Y\}$$

- learning procedure:

$$P : X \times Y \to A$$

usually:

$$P = \arg\min_{A \in \mathcal{A}} \mathcal{L}(X, Y, A)$$

where: $\mathcal{L} : \mathcal{A} \times X \times Y \to \mathbb{R}$ - a loss unction.

Model

Model is a set of <u>algorithms</u>.

Usually, algorithms are parametrized by some vector $\theta \in \mathbb{R}^n$. But nothing stops you from considering simply a bunch of algorithms.

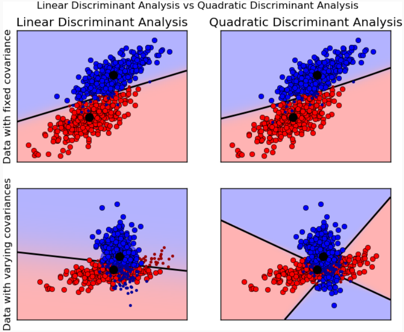


Figure 1: Separating hyperplane is a model too...

Hyper-parameters

Model can be parametrized itself, e.g. maximal depth of trees, regularization coefficients. These parameters are called **hyper-parameters** since they are not selected in learning procedure.

Learning procedures:

- brute force;
- random guessing;
- greedy methods
- continuous optimization:
  - gradient descent and Co.;
  - second order methods, e.g. Newton-Raphson method;
  - gradient-free optimization, e.g. genetics;
- discrete optimization.

Machine Learning = Model + Optimization.

No free lunch

Settings

- binary classification (for simplicity);
- Loss: accuracy:

$$L(A, X, y) = \frac{1}{|y|} \sum_i \mathbb{I}[A(x_i) = y_i]$$

- classifier (hypothesis) A, true relation $y = F(x)$:
- dataset D with $n = |D|$ and learning procedure $P_k(A \mid D)$;
- error for learning algorithm k:

$$E_k(F, n) = \sum_{x \notin D} P(x) L(A, x, y) P_k(A \mid D)$$

No free lunch

No free lunch theorem

1. Uniformly averaged over all target functions F:

$$E_1(F, n) - E_2(F, n) = 0$$

2. For any fixed trainingset D, uniformly averaged over F:

$$E_1(F, D) - E_2(F, D) = 0$$

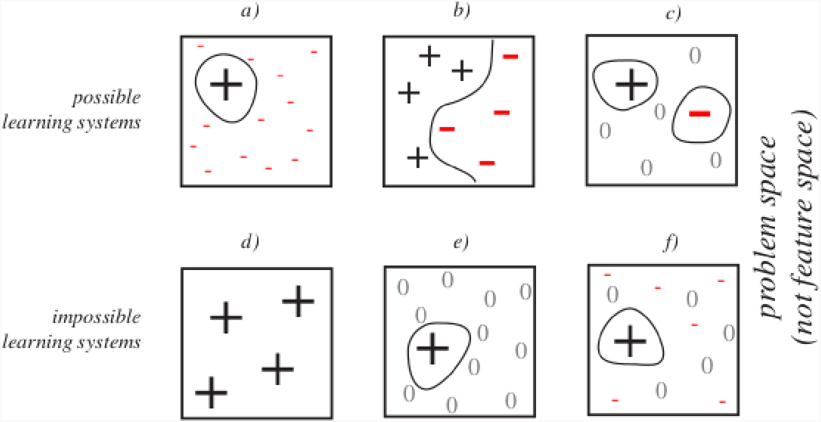
3. Uniformly averaged over all priors P(F):

$$E_1(n) - E_2(n) = 0$$

4. For any fixed trainingset D, uniformly averaged over P(F):

$$E_1(D) - E_2(D) = 0$$

No free lunch



Interpretation

> *The theorem tells that there is no universal learning algorithm. To successfully learn one type of problems you need to sacrifice generalization ability on the rest.*

Caveats:

- problems in the real world are not uniformly distributed;
- solving problem you have some prior knowledge about it;

Machine Learning algorithms generic recipe

Machine Learning algorithm:

- implicit or explicit assumptions about data;
- model;
- learning procedure.

Examples

Vanilla classification/regression:

- usually, assumptions on 'smoothness' of the functions;
- restrictions on number of interactions between variables.

Reinforcement learning:

- depends on sensor type;
- Markov properties;
- we can estimate 'goodness' of current state;
- assumptions on opponent/environment.

Examples

Vision:

- objects on image are connected region;
- strong correlation between neighbor pixels;
- position (or angle) of an object are not relevant.

Speech recognition:

- signal can be split into short samples;
- underlying language model;
- tone does not make any difference;
- a lot of samples correspond to the same class.

Examples

Recommendation systems:

- users with similar interest like similar items;
- latent variable models

Natural Language Processing:

- there is a lot of synonyms;
- words are highly correlated in a short term;
- almost uncorrelated in a long term;
- texts from the same author have similar statistical properties.