

Introduction to Machine Learning

NPFL 054

<http://ufal.mff.cuni.cz/course/npfl054>

Barbora Hladká

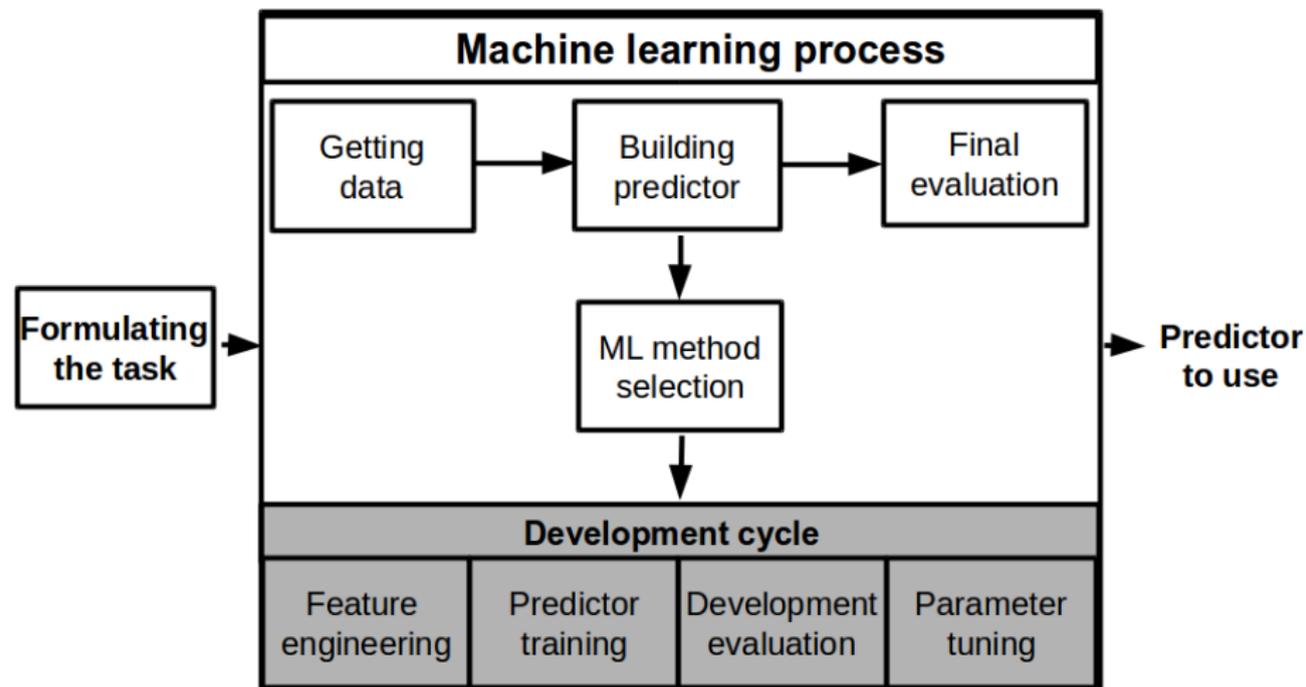
Martin Holub

{Hladka | Holub}@ufal.mff.cuni.cz

Charles University,
Faculty of Mathematics and Physics,
Institute of Formal and Applied Linguistics

Lecture #12

Machine learning overview



Machine learning overview

machine learning = representation + evaluation + optimization

representation	evaluation	optimization
instances k-NN	evaluation function accuracy/error rate precision, recall ROC curve	combinatorial greedy search
decision trees		
hyperplanes Naïve Bayes	objective function generative (conditional probability)	continuous <i>unconstrained</i> gradient descent,
Logistic regression	discriminative (conditional probability)	maximum likelihood estimation
SVM	margin	<i>constrained</i>
Perceptron	mean square error	quadratic programming
graphical models Bayesian networks		
neural networks		

Machine learning overview

Task and data management

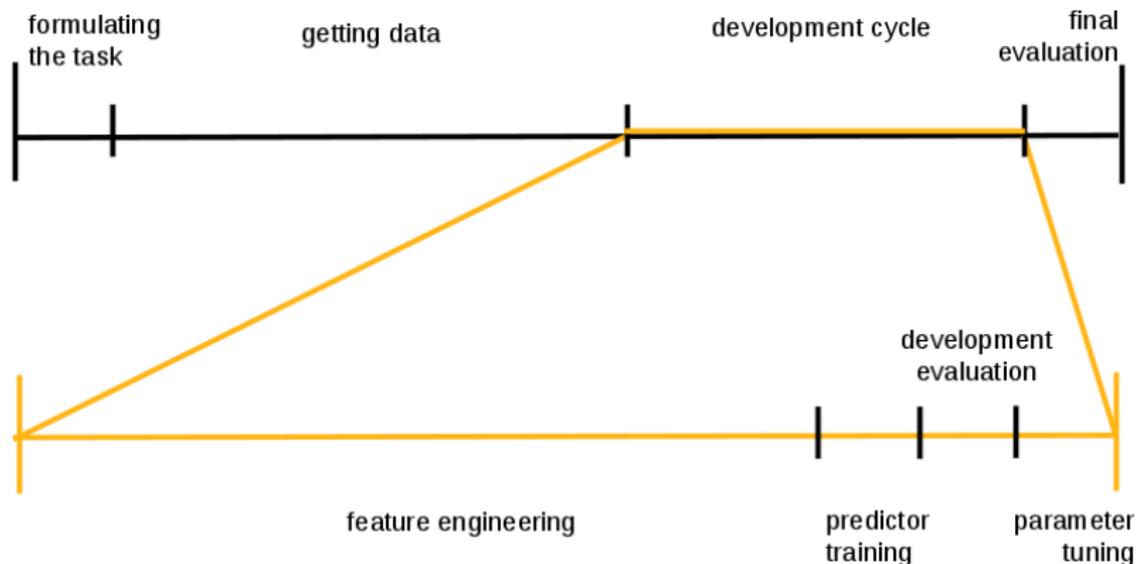
- ① Time management
- ② Formulating the task
- ③ Getting data
- ④ The more data, the better
- ⑤ Feature engineering
- ⑥ Curse of dimensionality

Methods and evaluation

- ⑦ Learning algorithms
- ⑧ Development cycle
- ⑨ Evaluation
- ⑩ Optimizing learning parameters
- ⑪ Overfitting
- ⑫ The more classifiers, the better
- ⑬ Theoretical aspects of ML

(1) Time management

How much time do particular steps take?



(2) Formulating the task

- Precise formulation of the task
- What are the objects of the task?
- What are the target values of the task?

(3) Getting data

- Gather data
- Assign true prediction
- Clean it
- Preprocess it
- Analyse it

(4) The more data, the better

If we don't have enough data

- **cross-validation** – The data set $Data$ is partitioned into subsets of equal size. In the i -th step of the iteration, the i -th subset is used as a test set, while the remaining parts from the training set.
- **bootstrapping** – New data sets $Data_1, \dots, Data_k$ are drawn from $Data$ with replacement, each of the same size as $Data$. In the i -th iteration, $Data_i$ forms the training set, the remaining examples in $Data$ form the test set

(5) Feature engineering

- Understand the properties of the objects
 - How they interact with the target value
 - How they interact each other
 - How they interact with a given ML algorithm
 - Domain specific
- Feature selection manually
- Feature selection automatically: generate large number of features and then filter some of them out

(6) Curse of dimensionality

- A lot of features \rightarrow high dimensional spaces
- The more features, the more difficult to extract useful information
- Dimensionality increases \rightarrow predictive power of predictor reduces
- The more features, the harder to train a predictor
- **Remedy:** feature selection, dimensionality reduction

(7) Learning algorithms

Which one to choose?

First, identify appropriate learning paradigm

- Classification? Regression?
- Supervised? Unsupervised? Mix?
- If classification, are class proportions even or skewed?

In general, **no learning algorithm dominates all others on all problems.**

(8) Development cycle

- Test developer's expectation
- What does it work and what doesn't?

(9) Evaluation

Model assessment

- **Metrics** and **methods** for performance evaluation
How to evaluate the performance of a predictor?
How to obtain reliable estimates?
- **Predictor comparison**
How to compare the relative performance among competing predictors?
- **Predictor selection**
Which predictor should we prefer?

(10) Optimizing learning parameters

Searching for the best predictor, i.e.

- adapting ML algorithms to the particulars of a training set
- optimizing predictor performance

Optimization techniques

- Grid search
- Gradient descent
- Quadratic programming
- ...

(11) Overfitting

- bias
- variance

To avoid overfitting using

- cross-validation
- feature engineering
- parameter tuning
- regularization

(12) The more classifiers, the better

- **Build an ensemble of classifiers** using
 - different learning algorithm
 - different training data
 - different features
- **Analyze** their performance: complementarity implies potential improvement
- **Combine** classification results (e.g. majority voting).

Examples of ensemble techniques

- **bagging** works by taking a bootstrap sample from the training set
- **boosting** works by changing weights on the training set

(13) Theoretical aspects

Computational learning theory (CLT) aims to understand fundamental issues in the learning process. Mainly

- How computationally hard is the learning problem?
- How much data do we need to be confident that good performance on that data really means something? I.e., accuracy and generalization in more formal manner
- CLT provides a formal framework to formulate and address questions regarding the performance of different learning algorithms. Are there any general laws that govern machine learners? Using statistics, we compare learning algorithms empirically

- Pedro Domingos. A Few Useful Things to Know about Machine Learning. 2012.
– <https://homes.cs.washington.edu/~pedrod/papers/cacm12.pdf>
- Pedro Domingos. Ten Myths About Machine Learning. 2016.
– <https://medium.com/@pedromdd/ten-myths-about-machine-learning-d888b48334a3>