B(E)3M33UI: Competencies

Petr Pošík

March 20, 2017

Abstract

List of competencies students shall gain after completing course *Artificial Intelligence* taught at Czech Technical University in Prague, Dept. of Cybernetics, primarilly for (but not limited to) students of Cybernetics and Robotics master study programme, branch Robotics.

1 Bayesian decision tasks. Non-bayesian tasks. Empirical learning.

After this lecture, a student shall be able to ...

- explain various view on AI and describe the differences of their personal view of AI;
- describe the fields of science with the greatest effect on AI;
- define Bayesian decision task and all its components (decision strategy, risk, penalty function, observation, hidden state, joint probability distribution);
- solve simple instances of Bayesian decision task by hand, write a computer program solving Bayesian decision tasks;
- explain features of Bayesian strategy;
- recognize special cases of Bayesian decision task (minimization of error probability when estimating hidden state, strategy with "dontknow" decision);
- describe reasons and examplify situations when the Bayesian approach cannot be used;
- define and describe examples of non-Bayesian tasks which can be solved to some extent without learning (Neyman-Pearson, minimax, Wald);
- solve simple instances of the above non-Bayesian decision tasks by hand, write a computer program solving them;
- define the decision strategy design as a learning from data;
- describe the differences between Bayesian decision tasks, non-Bayesian decision tasks and decision tasks solved by learning;
- define the types of learning (supervised, unsupervised, semisupervised, reinforcement) and describe conceptual differences between them;
- define classification and regression types of problems, recognize them in practical situations;
- describe 2 approaches to learning (as parameter estimation, as direct optimal strategy design) and give examples of surrogate criteria used in them.

2 Linear methods for regression and classification.

After this lecture, a student shall be able to ...

- define and recognize linear regression model (with scalar parameters, in scalar product form, in matrix form, non-homogenous and homogenous coordinates);
- define the loss function suitable for fitting a regression model;
- explain the least squares metod, draw an illustration;
- compute coefficients of simple (1D) linear regression by hand, write a computer program computing coefficients for multiple regression;
- explain the concept of discrimination function for binary and multinomial classification;

- define a loss function suitable for fitting a classification model;
- describe a perceptron algorithm, perform a few iterations by hand;
- explain the characteristics of perceptron algorithm;
- describe logistic regression, the interpretation of its outputs, and why we classify it as a linear model;
- define loss functions suitable for fitting logistic regression;
- define optimal separating hyperplane, explain in what sense it is optimal;
- define what a margin is, what support vectors are, and explain their relation;
- compute the margin given the parameters of separating hyperplane for which $\min_{i:y^{(i)}=+1}(x^{(i)}w^T + w_0) = 1$ and $\max_{i:y^{(i)}=-1}(x^{(i)}w^T + w_0) = -1$;
- formulate the primary quadratic programming task which results in the optimal separating hyperplane (including the soft-margin version);
- compute the parameters of optimal hyperplane given the set of support vectors and their weights.

3 Non-linear models. Basis expansion. Overfitting. Regularization.

After this lecture, a student shall be able to ...

- explain the reason for doing basis expansion (feature space straightening), and describe its principle;
- show the effect of basis expansion with a linear model on a simple example for both classification and regression settings;
- implement user-defined basis expansions in certain programming language;
- list advantages and disadvantages of basis expansion;
- explain why the error measured on the training data is not a good estimate of the expected error of the model for new data, and whether it under- or overestimates the true error;
- explain basic methods to get unbiased estimate of the true model error (testing data, k-fold crossvalidation, LOO crossvalidation);
- describe the general form of dependency of the model training and testing errors on the model complexity/flexibility/capacity;
- define overfitting;
- discuss high bias and high variance problems of models;
- explain how to proceed if a suitable model complexity must be chosen as part of the training process;
- list 2 basic methods of overfitting prevention;
- describe the principles of ridge (Tikhonov) and lasso regularizations and their effects on the model parameters.

4 Nearest neighbors. Kernels, SVM. Decision trees.

After this lecture, a student shall be able to ...

- explain, use, and implement method of k nearest neighbors for both classification and regression;
- explain the influence of k to the form of the final model;
- describe advantages and disadvantages of k-NN, and suggest a way hot to find a suitable value of k;
- show how to force the algorithm for learning the optimal separating hyperplane to find a nonlinear model using basis expansion, and using a kernel function;
- explain the meaning of kernels, and their advantages compared to basis expansion;
- explain the principle of support vector machine;
- describe the structure of classification and regression tree, and the way it is used to determine a prediction;
- know a lower bound on the number of Boolean decision trees for a dataset with n attributes;
- describe TDIDT algorithm and its features, and know whether it will find the optimal tree;
- explain how to choose the best attribute for a split, and be able to manually perform the choice for simple examples;

- describe 2 methods to prevent tree overfitting, and argue which of them is better;
- explain how a decision tree can handle missing data during training and during prediction;
- describe what happens and what to do if the dataset contains an attribute with unique value for each observation;
- explain how to handle continuous input and output variables (as opposed to the discrete attributes).

5 Bagging. Random forests. Boosting.

After this lecture, a student shall be able to ...

- describe the basic principle behind all committee/ensemble methods;
- list and conceptually compare several methods to achieve diversity among models trained on the same data, and know which of these methods are used in which ensemble algorithms;
- explain the purpose and the basic principle of stacking;
- explain how a bootstrap sample is created from the available data, and describe its properties;
- describe features of bagging;
- explain how to compute out-of-bag error estimate when using bagging;
- explain the principle of random forests and describe their difference to bagging with trees;
- explain how to compute a score of variable importance using random forest;
- explain the hypothesis boosting problem, and define a weak and a strong classifier in this context;
- explain the basic principle of AdaBoost.M1 algorithm;
- relate the training error of the AdaBoost algorithm to the number of constituent models and to the errors of individual models;
- describe the relations of AdaBoost.M1, L2Boost, and Gradient Boosting.