University of Wrocław 30.08.2019

Methods of classification and dimensionality reduction

Syllabus

Paweł Lorek

Course description and objectives

The course provides a survey of dimensionality reduction (feature extraction) and classification methods. Dimensionality reduction enhances the performance of computer vision and machine learning-based approaches, it allows to represent the data in a more efficient way, it allows to visualise high-dimensional data. Among others, we will study principal component analysis (PCA) and singular value decomposition (SVD), non-negative matrix factorization (NMF), independent component analysis (ICA), t-distributed Stochastic Neighbour Embedding (t-SNE). Concerning classification methods, we study many classical "shallow-learning" classifiers, e.g., nearest neighbours, naive Bayes, support vector machine (SVM), linear and quadratic discriminant analysis (LDA and QDA), decision trees, ensemble methods (bagging, AdaBoost, Random Forest). We will also present (together with several applications) Gaussian Mixture Models and Hidden Markov Models, the models include the Expectation-Maximization (EM) algorithm for learning the parameters. We will also shortly discuss Monte Carlo Markov Chain based methods for "recovering" broken images. We will introduce Markov chains from scratch, however, some basic knowledge and intuition on these chains is advisable.
Though all details are provided for most methods, we put a strong emphasis on an intuition and practical applications: we discuss (and apply the acquired knowledge to various practical problems in lab classes), e.g., classification of multidimensional data (including time series, images and texts), image compression, topic recovery, recommendation systems. This course will familiarize students with a broad cross-section of machine learning algorithms. Students will be prepared for research or industry application of acquired methods.

Teaching methods

Lecture, individual work at the computer, project, presentation of a project, consulting ideas for solving tasks

Requirements

Courses

- Algebra
- Probability theory

Other requirements

- Basic (at least) Python

Course content

- Introduction to feature selection and feature extraction (aka dimension reduction)
- Principal component analysis (PCA), singular value decomposition (SVD), independent component analysis (ICA). Application to image compression.
- Kernel trick and Kernel PCA.
- Nonnegative matrix factorization (NMF). Applications to recommendation systems.
- Dimensionality reduction for data visualisation: t-distributed Stochastic Neighbour Embedding (t-SNE), UMAP.
- Classical classifiers: nearest neighbours, naive Bayes, Lagrange multipliers – primal and dual problems, support vector machine (SVM), linear discriminant analysis (LDA) as both, dimension reduction technique and a classifier, creating multiclass classifiers from binary ones.
- Clustering algorithms. K-means clustering, Gaussian Mixture Model (GMM). Maximum likelihood estimation, Expectation-Maximization (EM) algorithm. Applications to generating images of hand-written digits.
- Working with text, bag of words, vector representations of words (e.g., word2vec)
- Hidden Markov models (HMM) with discrete and continuous observations, the forward and backward procedure, the Baum-Welch algorithm, the Viterbi algorithm. Applications to error corrections, classifying cells in fluorescent microscopy and to classification of time-series.
- Introduction to Monte Carlo Markov Chain (MCMC) methods. Gibbs sampler. Application to "recovering" broken (in a specific, known way) images.

Learning outcomes

Knowledge

- Student knows several dimension reduction techniques
- Student knows fundamental "shallow-learning" classification algorithms
- Student knows basic clustering algorithms
- Student knows how to analyze the complexity of given problems and how to come up with some optimizations

Skills

- Student is able to design and implement solutions involving machine learning algorithms
- Student is familiar with application of machine learning algorithms to real-life problems, e.g., recommendation systems, text classification, image compression, data visualisation
- Student is able to structure and prepare scientific and technical documentation describing project activities
- Student is able to autonomously extend the knowledge acquired during the course by reading and understanding scientific and technical documentation

Verification methods

two projects, implementation and presentation of computer programs, oral exam (presentation and questions), written reports

Rules and conditions

- *Laboratories*: There will be points for 2 projects. To pass, the minimum number of points must be collected. Handing project = handing report (in .pdf) and a working program.
- *Oral exam*: Presentation of own projects and answering questions

Student workload

- Classes with the teacher:
  - lectures - 30 hours

  - laboratories - 30 hours
  - oral exam (including presentation) - 45 minutes
- Student's own work:
  - preparing for classes - 15 hours
  - reading additional material - 30 hours
  - implementing algorithms, writing reports - 45 hours

Bibliography

[1] Andreas C. Muller; Sarah Guido *Introduction to machine learning with Python: a guide for data scientists*, Sebastopol O'Reilly Media, Inc., 2017

[2] Matthew Kirk *Thoughtful Machine Learning With Python*, Sebastopol O'Reilly Media, Inc., 2017

[3] Olle Häggström. *Finite Markov Chains and Algorithmic Applications*, Cambridge University Press, 2002. (Chapters 1–7)