

Price: € 2,700.00 excl. VAT
Duration: 5 consecutive days
Contact: training@hightechinstitute.nl, +31 85 401 3600

Intro

The course is a set of lectures each followed by practical exercises with Machine Learning Software and Matlab. Emphasis is put on the "how-to-do-it" approach. The hands-on sessions are on real-world industrial case studies. The participants are encouraged to bring their own data to get help and advice on how to deal with it.

The teachers have extensive experience with design of industrial machine learning systems in different application areas, and a rich academic and teaching background from Delft University of Technology with respectively 10 and 15 years experience in machine learning training for industry.

The course is structured to be useful also without the perClass Machine Learning software. Included in the course price is a license and support fee of the perClass software. The license and support are valid starting at the moment of installation and remain valid until one month.

Objective

After the course, the participant:

- has a solid and state-of-the-art knowledge on machine learning and its applications;
- knows how to make a good labelled set;
- knows how to train and assess a classifier;
- knows to develop an efficient classifier;
- knows how to extract a useful data set from a raw data set,
- knows how to use the perClass software;
- knows the basics of deep learning and when to use it;
- knows how to build a robust Machine Learning system suitable to solve real-world industrial projects on a step-by-step approach;
- knows how to avoid common pitfalls;

Intended for

Engineers from R&D, practitioners and teachers interested in machine learning and deep learning. The course is suitable both for those who are new to machine learning and who are already familiar. The first group can concentrate on the basic concepts and learn to apply these in projects. The last group can improve their skills to design accurate and robust systems.

Education: At least BSc.

Basic knowledge of MATLAB is recommended.

For the hands-on session the participants need to bring their laptop with Matlab installed (version 7.5 or newer). A few days before the course starts the perClass software is distributed to be installed on the laptop.



Certification

Participants will receive a High Tech Institute course certificate for attending this training.

Course leader

Hans Vink MSc

Trainers

Carmen Lai PhD
Pavel Paclik PhD

Methods

Classroom lectures, classroom demonstrations and hands-on sessions on the perClass software.

Course material: lecture notes with a copy of the slides and textual documents.

Program

Module 1: Introduction to Machine Learning

Overview of what is needed to design a machine learning system. Supervised and unsupervised classification. Training from examples. Concept of a class, feature, data sample. Examples of several typical scenarios.

Module 2: Data and labels

How to annotate my data with multiple types of meta-data? Why this helps classifier design? Tools: sddata object; constructing data sets; using properties; labels, categories and lists; working with data subsets; visualisation using scatter and image views; normalization issues; data visualization and its relevance.

Module 3: Supervised Learning

What is a classifier? How to train a classifier? How to choose a good classifier for my problem? Bayes theorem; generative and discriminative classifiers; parametric and non-parametric models; naive Bayes; linear, quadratic, and mixture models; Parzen density estimation; linear discriminant analysis; nearest-neighbor rules; support-vector machines; perceptron; neural networks; decision trees; random forests.

Module 4: Evaluation and model selection

How to reliably estimate classifier performance? How to choose a good performance measure? How to test a classifier on an unseen object / patient? Tools: Error and performance measures; confusion matrix; learning curves; overtraining; classifier complexity; cross-validation.

Module 5: Dimensionality reduction

Why more features aren't always giving better classifiers? How to choose or create smaller feature subset? What features are useful? Tools: visualizing feature distributions; measures of overlap; feature selection with individual, greedy, and floating search strategies; genetic search, feature extraction; PCA, LDA, non-linear extraction methods.

Module 6: Classifier optimisation

How to make sure we meet performance requirements? How to change behavior of already trained classifier? How to deal with skewed data sets (one class much smaller than others)? How to protect classifier from outliers and concepts unknown in training? Tools: Target detection, one-class classification, ROC analysis for two-class and multi-class problems; class imbalance; performance constraints; cost-sensitive optimization; handling of prior probabilities; rejection of outliers, rejection of low-confidence regions (to find areas of overlap = difficult samples).

Module 7: Advanced data handling

How to get from raw files to data sets? How to clean raw data? How to learn from (multi-band) image data? Defining the machine learning problem; importing images with annotation; computing local image features in regions; representation for texture and appearance classification; working with high-resolution imagery; extracting local features on a sparse grid, passing labels and classifier decisions between sparse and original image data; training from data extracted from multiple images; dealing with multi-band and hyper-spectral images; extracting spectral bands; importing data from databases using SQL queries; handling data sets that don't fit in memory; handling data validity; working with missing data (removal and imputation).

Module 8: Deep Learning

What is Deep Learning? What problems does it solve better than other approaches? How to build a reliable Deep learning classifier? Tools: Building blocks of convolutional neural networks (CNNs). Strengths and

weaknesses of deep learning. How to build reliable CNNs? How to integrate with other machine learning tools (ROC, cascading with other classifiers).

Module 9: Clustering, similarity representations, classifier fusion

How to define groups of similar observations? How to interpret clustering results? How to combine multiple classifiers? How to incorporate prior knowledge in custom similarity measures and learn from them.

Using clusters to quickly label data or build better classifiers in multi-modal problems; Visualizing clustering solutions; Leveraging clustering as a tool to understand the source of classification errors; Deciding on the number of clusters; Dissimilarity measures; k-means; mixture models, EM algorithm; hierarchical clustering; evidence accumulation, Representing measurements by proximities; building classifier in dissimilarity spaces; Classifier fusion; crisp and trained combiners; Robust combining system based on unbiased estimation of second-stage soft outputs; Cascading of classifiers (solving difficult problems with different features/models than simple ones).

Module 10: System design

How to build robust systems? Why may the optimization of a single component (classifier) not yield a good system performance? System design work-flow.

Tools: Role of meta-data, how to setup robust and realistic system evaluation, custom algorithms, automatic selection of operating points, local and object-level classification, cross-validation over objects.

Module 11: classifier deployment, embedded classifiers in production

How to move from a research prototype to a production machine? Is my classifier fast enough? How to speed up classifier execution? How to directly test research ideas real-time in production machine?

Execution complexity of classifiers; how to measure speed; Performance vs speed characteristics; Classifier speedup strategies; cascading for faster execution; Practical real-time embedding out of Matlab with perClass Runtime; linking perClass Runtime to a custom application; API walkthrough; accessing decision names; using multiple pipelines; changing operating points in production; strategies to speed up classifier execution.

Trainers

Carmen Lai PhD

Pavel Paclik PhD