Statistical Machine Learning (BE4M33SSU)
Lecture 1.

Czech Technical University in Prague

Course format

Teachers: Jan Drchal, Boris Flach, Vojtech Franc and Jakub Paplham

Format: 1 lecture & 1 tutorial per week (6 credits), tutorials of two types

- seminars: discussing solutions of theoretical assignments (published a week before the class). You are expected to work on them in advance.
- practical labs: explaining and discussing practical homeworks, i.e. implementation of selected methods in Python (or Matlab). You have to submit
  1. a report in PDF format (typeset preferably in LaTeX). Exception: if necessary, you may include lengthy formula derivations as handwritten scans.
  2. your code either as source file or as python notebook. The code must be executable.

Grading: 40% homeworks + 60% written exam = 100% (+ bonus points)

Prerequisites:

- probability theory and statistics (A0B01PSI)
- pattern recognition and machine learning (AE4B33RPZ)
- optimisation (AE4B33OPT)

More details: https://cw.fel.cvut.cz/wiki/courses/be4m33ssu/start

Goals

The aim of statistical machine learning is to develop systems (models and algorithms) for solving prediction tasks given a set of examples and some prior knowledge about the task.

Machine learning has been successfully applied e.g. in areas

- text and document classification,
- speech recognition and natural language processing,
- computational biology (genes, proteins) and biological imaging & medical diagnosis
- computer vision,
- fraud detection, network intrusion,
- and many others

You will gain skills to construct learning systems for typical applications by successfully combining appropriate models and learning methods.

Characters of the play

- **object features** $x \in \mathcal{X}$ are observable; x can be:
  a categorical variable, a scalar, a real valued vector, a tensor, a sequence of values, an image, a labelled graph, . . .
- **state of the object** $y \in \mathcal{Y}$ is usually hidden; y can be: see above
- **prediction strategy** (a.k.a. inference rule) $h\colon \mathcal{X} \to \mathcal{Y}$; depending on the type of $\mathcal{Y}$:
  - y is a categorical variable $\Rightarrow$ classification
  - y is a real valued variable $\Rightarrow$ regression
- **training examples** $\mathcal{T} = \{(x, y) \mid x \in \mathcal{X}, y \in \mathcal{Y}\}$
- **loss function** $\ell\colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ penalises wrong predictions,
  i.e. $\ell(y, h(x))$ is the loss for predicting $y' = h(x)$ when y is the true state

Goal: optimal prediction strategy $h\colon \mathcal{X} \to \mathcal{Y}$ that minimises the loss

> Q: give meaningful application examples for combinations of different $\mathcal{X}$, $\mathcal{Y}$ and related loss functions

Statistical machine learning

Main assumption:

- X, Y are random variables,
- X, Y are related by an unknown joint p.d.f. $p(x, y)$,
- we can collect examples (x, y) drawn from $p(x, y)$.

Typical concepts:

- regression: $Y = f(X) + \epsilon$, where f is unknown and ϵ is a random error,
- classification: $p(x, y) = p(y)p(x \mid y)$, where $p(y)$ is the prior class probability and $p(x \mid y)$ the conditional feature distribution.

Consequences and problems

- the inference rule $h(X)$ and the loss $\ell(Y, h(X))$ become random variables.
- risk of an inference rule $h(X) \Rightarrow$ expected loss

$$R(h) = \mathbb{E}[\ell(Y, h(X))] = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y)\ell(y, h(x))$$

- how to estimate $R(h)$ if $p(x, y)$ is unknown?
- how to choose an optimal predictor $h(x)$ if $p(x, y)$ is unknown?

Statistical machine learning

Estimating $R(h)$:

collect an i.i.d. test sample $\mathcal{S}^m = \big\{(x^i, y^i) \in \mathcal{X} \times \mathcal{Y} \mid i = 1, \ldots, m\big\}$ drawn from the distribution $p(x, y)$,

estimate the risk $R(h)$ of the strategy h by the empirical risk

$$R(h) \approx R_{\mathcal{S}^m}(h) = \frac{1}{m} \sum_{i=1}^{m} \ell(y^i, h(x^i))$$

Q: how strong can they deviate from each other? (see next lectures)

$$\mathbb{P}\Big(|R_{\mathcal{S}^m}(h) - R(h)| > \epsilon \Big) \leq ??$$

Choosing an optimal inference rule $h(x)$

If $p(x, y)$ is known:

The smallest possible risk is

$$R^* = \inf_{h \in \mathcal{Y}^\mathcal{X}} R(h) = \inf_{h \in \mathcal{Y}^\mathcal{X}} \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \ell(y, h(x)) = \sum_{x \in \mathcal{X}} p(x) \inf_{y' \in \mathcal{Y}} \sum_{y \in \mathcal{Y}} p(y \,|\, x) \ell(y, y')$$

The corresponding best possible inference rule is the Bayes inference rule

$$h^*(x) = \arg\min_{y' \in \mathcal{Y}} \sum_{y \in \mathcal{Y}} p(y \,|\, x) \ell(y, y')$$

But $p(x, y)$ is not known and we can only collect examples drawn from it. We need:

Learning algorithms that use training data and prior assumptions/knowledge about the task

Learning types

Training data:

- if $\mathcal{T}^m = \{(x^i, y^i) \in \mathcal{X} \times \mathcal{Y} \mid i = 1, \ldots, m\}$ $\Rightarrow$ supervised learning
- if $\mathcal{T}^m = \{x^i \in \mathcal{X} \mid i = 1, \ldots, m\}$ $\Rightarrow$ unsupervised learning
- if $\mathcal{T}^m = \mathcal{T}_l^{m_1} \bigcup \mathcal{T}_u^{m_2}$, with labelled training data $\mathcal{T}_l^{m_1}$ and unlabelled training data $\mathcal{T}_u^{m_2}$ $\Rightarrow$ semi-supervised learning

Prior knowledge about the task:

- **Discriminative learning:** assume that the optimal inference rule h^* is in some class of rules $\mathcal{H}$ $\Rightarrow$ replace the true risk by empirical risk

$$R_{\mathcal{T}}(h) = \frac{1}{|\mathcal{T}|} \sum_{(x,y) \in \mathcal{T}} \ell(y, h(x))$$

and minimise it w.r.t. $h \in \mathcal{H}$, i.e. $h^*_{\mathcal{T}} = \arg\min_{h \in \mathcal{H}} R_{\mathcal{T}}(h)$.

Q: How strong can $R(h^*_{\mathcal{T}})$ deviate from $R(h^*)$? How does this deviation depend on $\mathcal{H}$?

$$\mathbb{P}\Big(|R(h^*_{\mathcal{T}}) - R(h^*)| > \epsilon\Big) \leq ??$$

Learning types

- **Generative learning:** assume that the true p.d. $p(x, y)$ is in some parametrised family of distributions, i.e. $p = p_{\theta^*} \in \mathcal{P}_\Theta \Rightarrow$ use the training set $\mathcal{T}$ to estimate $\theta \in \Theta$:

  1. $\theta^*_\mathcal{T} = \arg\max_{\theta \in \Theta} \log p_\theta(\mathcal{T})$, i.e. maximum likelihood estimator,
  2. set $h^*_\mathcal{T} = h_{\theta^*_\mathcal{T}}$, where h_θ denotes the Bayes inference rule for the p.d. p_θ.

  Q: How strong can $\theta^*_\mathcal{T}$ deviate from θ^*? How does this deviation depend on $\mathcal{P}_\Theta$?

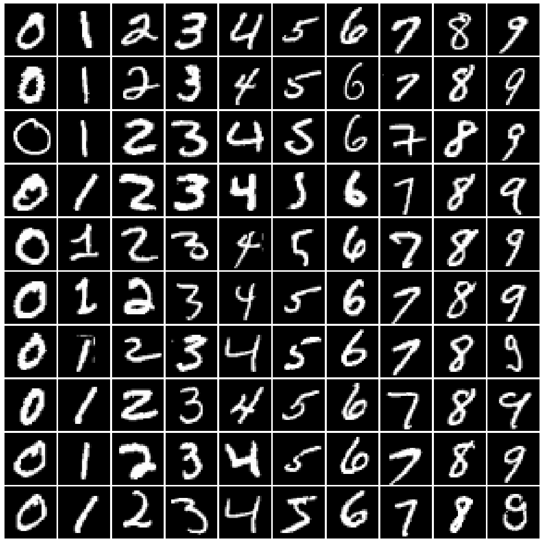
Possible combinations (training data vs. learning type)

	discr.	gener.
superv.	yes	yes
semi-sup.	(yes)	yes
unsuperv.	no	yes

In this course:

- discriminative: Support Vector Machines, Deep Neural Networks
- generative: mixture models, Hidden Markov Models
- other: Bayesian learning, Ensembling

Example: Classification of handwritten digits



$x \in \mathcal{X}$ - grey valued images, 28x28, $y \in \mathcal{Y}$ - categorical variable with 10 values

- **discriminative:** Specify a class of strategies $\mathcal{H}$ and a loss function $\ell(y, y')$. How would you estimate the optimal inference rule $h^* \in \mathcal{H}$?
- **generative:** Specify a parametrised family $p_\theta(x, y)$, $\theta \in \Theta$ and a loss function $\ell(y, y')$. How would you estimate the optimal θ^* by using the MLE? What is the Bayes inference rule for p_{θ^*}?