Statistical Machine Learning (BE4M33SSU)
Lecture 1.

Czech Technical University in Prague

Course format

Teachers: Jan Drchal, Vojtech Franc and Jakub Paplhám

Format: 1 lecture & 1 seminar per week (6 credits)

Seminars: Solving theoretical assignments; explaining and discussing homeworks.

Homeworks:

1. Automaticaly evaluated: You have to submit a Python code.
2. Manually evaluated: You have to submit i) PDF report and ii) a Python code.

Grading:

- Thresholds for passing: at least 50% of the regular points in the practical labs and at least 50% of the regular points in the exam.
- 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

- email spam detection,
- computer vision,
- credit scoring,
- medical diagnosis,
- recommendation systems,
- speech recognition,
- network intrusion,
- natural language processing,
- and many others

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

Prediction problem

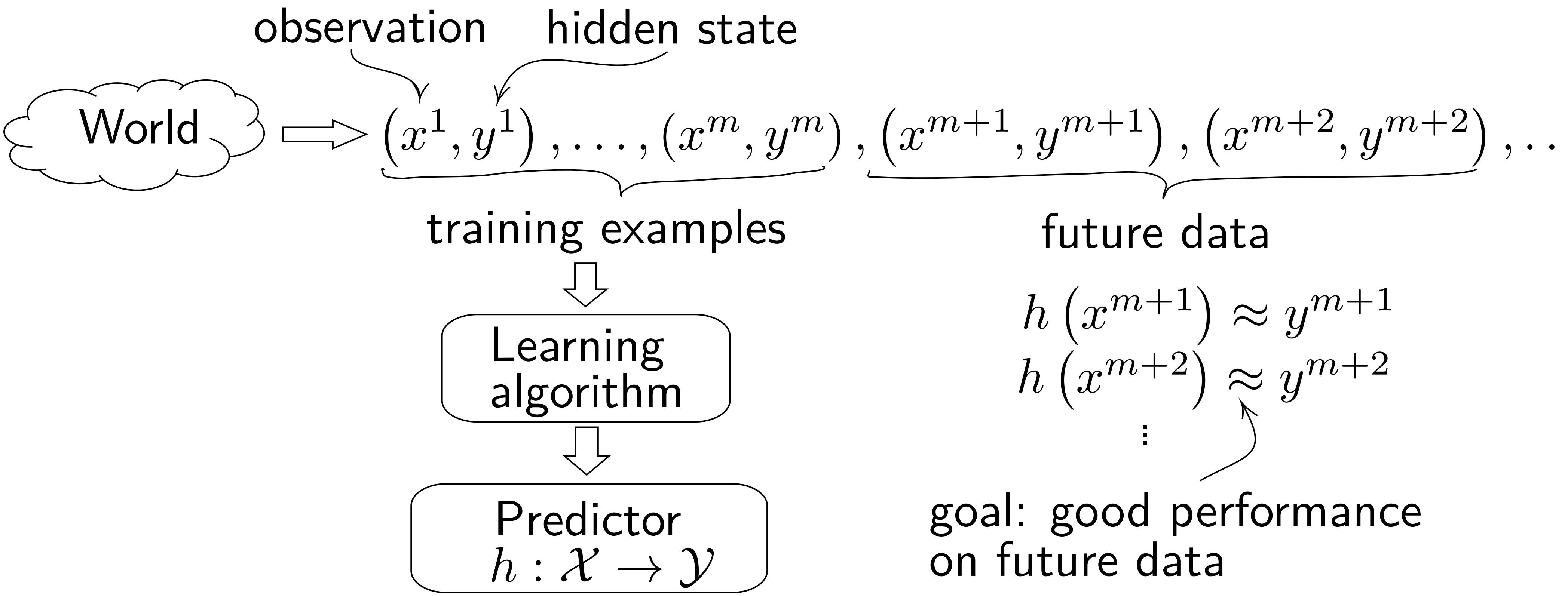
Example: *Given data representing weight of adult males and females:*

weight [kg]	99	65	83	76	77	$\cdots$
gender	male	female	male	male	female	$\cdots$

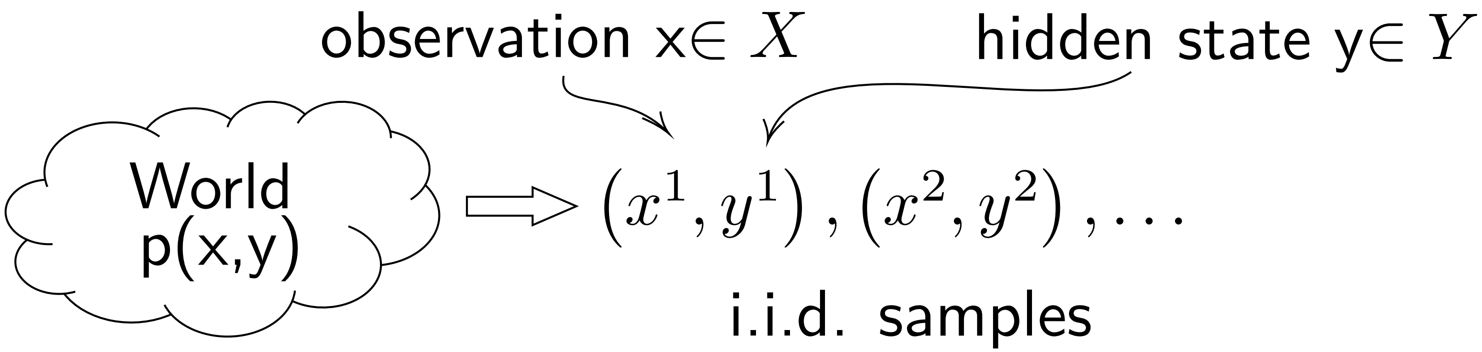
we want a predictor which outputs a person's gender given his/her weight.

- **Input observations (features)** $x \in \mathcal{X}$; x can be: a categorical variable, a scalar, a real valued vector, a tensor, a sequence of values, an image, a labelled graph, ldots
- **Hidden state (target variable, output)** $y \in \mathcal{Y}$; y can be: see above
- **Prediction strategy (predictor)** $h\colon \mathcal{X} \to \mathcal{Y}$; depending on the type of $\mathcal{Y}$:
  - y is a categorical variable $\Rightarrow$ classification
  - y is a real valued variable $\Rightarrow$ regression

Machine learning



Main assuption: i.i.d. data



- **Data:** $(x^1, y^1), (x^2, y^2), \ldots, (x^n, y^n)$ are samples drawn from independent and identically distributed (i.i.d.) pairs of random vars $(X^1, Y^1), (X^2, Y^2), \ldots$
- **Identically distributed:**

$$p(X^1 = x, Y^1 = y) = p(X^2 = x, Y^2 = y) = \cdots = p(X^n = x, Y^n = y)\,, \quad \forall x \in \mathcal{X}, y \in \mathcal{Y}$$

Remark: we will use $p(x, y)$ instead of $p(X = x, Y = y)$.

- **Independent:** the occurrence of one pair does not affect the occurrence of another:

$$p\big(x^1, y^1, x^2, y^2, \ldots, x^n, y^n\big) = p(x^1, y^1)\, p(x^2, y^2)\, \cdots p(x^n, y^n)$$

The optimal predictor

- **Loss function** $\ell\colon \mathcal{Y}\times\mathcal{Y}\to\mathbb{R}_+$ penalises wrong predictions, i.e. $\ell(y,\hat{y})$ is the loss for predicting $\hat{y}=h(x)$ when y is the true state.

  Example: 0/1-loss

$$\ell(y,\hat{y}) = \begin{cases} 0 & \text{if} \quad y=\hat{y} \\ 1 & \text{if} \quad y\neq\hat{y} \end{cases}$$

- **Expected risk (a.k.a. generalization error)** evaluates the performance of a predictor $h\colon \mathcal{X}\to\mathcal{Y}$ on unseen data:

$$R(h) = \int \sum_{y\in\mathcal{Y}} \ell(y,h(x))\, p(x,y)\,\mathrm{d}x = \mathbb{E}_{(x,y)\sim p}\Big[\ell(y,h(x))\Big]$$

  The expected risk $R(h)$ represents the average loss $\ell(y,h(x))$ when evaluated over large i.i.d. sample $(x^1,y^1),\ldots,(x^n,y^n)$:

$$\frac{1}{n}\Big(\ell(y^1,h(x^1)) + \ell(y^2,h(x^2)) + \cdots + \ell(y^n,h(x^n))\Big) \xrightarrow{p} R(h)$$

- **Bayes optimal predictor**:

$$h^* \in \operatorname*{arg\,min}_{h\in\mathcal{Y}^{\mathcal{X}}} R(h) \quad \Rightarrow \quad h^*(x) = \operatorname*{arg\,min}_{\hat{y}\in\mathcal{Y}} \sum_{y\in\mathcal{Y}} p(y\mid x)\,\ell(y,\hat{y})$$

Data split for learning and evaluation

- **Setup:** we have only samples i.i.d drawn from an unknown $p(x,y)$.

$$\text{World } p(x,y) \Longrightarrow \underbrace{(x^1,y^1),\dots,(x^m,y^m)}_{\text{training set}},\underbrace{(x^{m+1},y^{m+1}),\dots,(x^{m+l},y^{m+l})}_{\text{test set}},\dots$$

- **Learning**: find $h\colon \mathcal{X} \to \mathcal{Y}$ with small generalization error $R(h)$ using *training (sequence) set*

$$\mathcal{T}^m = ((x^i,y^i) \in (\mathcal{X} \times \mathcal{Y}) \mid i = 1,\dots,m) \quad \text{drawn i.i.d. from} \quad p(x,y)$$

- **Evaluation**: estimate the expected risk $R(h)$ of a given predictor $h\colon \mathcal{X} \to \mathcal{Y}$ using *test (sequence) set*

$$\mathcal{S}^l = ((x^i,y^i) \in (\mathcal{X} \times \mathcal{Y}) \mid i = 1,\dots,l) \quad \text{drawn i.i.d. from} \quad p(x,y)$$

Evaluation

- **Goal:** Given a predictor $h\colon \mathcal{X} \to \mathcal{Y}$ and a test set $\mathcal{S}^l = \big((x^i, y^i) \in \mathcal{X} \times \mathcal{Y} \mid i = 1, \ldots, l\big)$ drawn i.i.d. from an unknown distribution $p(x, y)$, estimate is the expected risk

$$R(h) = \mathbb{E}_{(x,y)\sim p}[\ell(y, h(x))]$$

- **Approach:**
  - Estimate the expected risk $R(h)$ by computing the empirical risk (test error)

$$R_{\mathcal{S}^l}(h) = \frac{1}{l}\big(\ell(y^1, h(x^1)) + \cdots + \ell(y^l, h(x^l))\big) = \frac{1}{l}\sum_{i=1}^{l} \ell(y^i, h(x^i))$$

- **Issues:**
  - How much can $R(h)$ deviate from $R_{\mathcal{S}^l}(h)$?
    Lecture: “Predictor Evaluation”

Empirical Risk Minimization (a.k.a discriminative) approach to learning

- **Goal:** given a training set $\mathcal{T}^m = ((x^i, y^i) \in \mathcal{X} \times \mathcal{Y} \mid i = 1, \ldots, m)$ i.i.d. draw from an unknown $p(x, y)$, find a good approximation of the Bayes predictor $h^*(x) = \arg\min_{y' \in \mathcal{Y}} \sum_{y \in \mathcal{Y}} p(y \mid x) \ell(y, y')$
- **Approach:**
  - Use prior knowledge to choose a hypothesis space $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}} = \{h \colon \mathcal{X} \to \mathcal{Y}\}$
  - Approximate the expected risk $R(h)$ by the empirical risk (training error)

$$R_{\mathcal{T}^m}(h) = \frac{1}{m}\big(\ell(y^1, h(x^1)) + \cdots + \ell(y^m, h(x^m))\big) = \frac{1}{m}\sum_{i=1}^{m} \ell(y^i, h(x^i))$$

  - Learn the predictor by minimizing the emprical risk:

$$h_m \in \arg\min_{h \in \mathcal{H}} R_{\mathcal{T}^m}(h)$$

- **Issues:**
  - How much can $R(h)$ deviate from $R_{\mathcal{T}^m}(h)$? How does it depend on $\mathcal{H}$?
    Lecture: "Empirical Risk Minimization"
  - How much can $R(h_m)$ deviate from $R(h^*)$?
    Lecture: "PAC learning"

Instances of ERM approach

- Learning algorithms implementing the Empirical risk minimization approach, i.e.

$$h_m \in \operatorname*{arg\,min}_{h \in \mathcal{H}} R_{\mathcal{T}^m}(h)$$

differ in the choice of the hypothesis space $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ and the loss $\ell\colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$.

- **Support Vector Machines:** $\mathcal{H} = \{h(x) = \operatorname{sign}(\langle \boldsymbol{\phi}(x), \boldsymbol{w} \rangle + b)\}$ is a space of linear classifiers; ℓ is the hinge loss.
  - Lecture: "Support Vector Machines"
- **Neural networks:** $\mathcal{H} = \{h(x) = h_L(\cdots(h_2(h_1(x)))\cdots)\}$ is a space of neural networks; ℓ is e.g. cross-entropy loss or L_2-loss.
  - Two lectures: "Supervised learning of deep neural networks" and "SGD"
- **Ensemble predictors:** $\mathcal{H} = \{h(x) = \psi(h_1(x)\alpha_1 + h_2(x)\alpha_2 + \cdots + h_L(x)\alpha_L)\}$ is a space of predictors that combine predictions of multiple models; ℓ is e.g. logistic loss or L_2 loss.
  - Two lectures: "Ensembling I" and "Ensembling II"

Generative Learning

- **Goal:** given a training set $\mathcal{T}^m = ((x^i, y^i) \in \mathcal{X} \times \mathcal{Y} \mid i = 1, \ldots, m)$ i.i.d. draw from an unknown $p(x, y)$, find a good approximation of the Bayes predictor $h^*(x) = \arg\min_{y' \in \mathcal{Y}} \sum_{y \in \mathcal{Y}} p(y \mid x)\ell(y, y')$

- **Approach:**
  - Use the training set $\mathcal{T}^m$ to find $\hat{p}(x, y)$ that approximates the true p.d. $p(x, y)$.
  - Use the estimated p.d. $\hat{p}(x, y)$ to construct the plugin Bayes predictor

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

- **Maximum Likelihood estimation:**
  - Assume the true p.d. $p(x, y)$ is in some parametrized family of distributions $\mathcal{P} = \{p_\theta(x, y) \mid \theta \in \Theta\}$.
  - Find ML estimate of the parameters $\theta_m = \arg\max_{\theta \in \Theta} \frac{1}{m} \sum_{i=1}^m \log p_\theta(x^i, y^i)$.
  - Insert $\hat{p}(x, y) = p_{\theta_m}(x, y)$ to the plugin Bayes predictor.
  - Two lectures: "Maximum Likelihood Estimator" and "Expectation-Maximization Algorithm"

Generative Learning

Bayesian Learning:

- Assume the true p.d. $p(x, y)$ is in some parametrized family of distributions $\mathcal{P} = \{p(x, y \mid \theta) \mid \theta \in \Theta\}$.
- Interpret the unknown parameter $\theta \in \Theta$ as a random variable.
- Assume we know the prior distribution $p(\theta)$ on Θ.
- Approach 1: MAP estimate

$$\theta_m = \arg\max_{\theta \in \Theta} p(\theta \mid \mathcal{T}^m)$$

  and set $\hat{p}(x, y) = p(x, y \mid \theta_m)$.

- Approach 2: Bayesian inference

$$h(x, \mathcal{T}^m) = \arg\max_{y \in \mathcal{Y}} \int_{\Theta} p(x, y \mid \theta)\, p(\theta \mid \mathcal{T}^m) d\theta$$

- Lecture: "Bayesian Learning"

Generative vs. Discriminative Learning

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

Comparison of discriminative and generative learning

	discriminative approach	generative generative
supervised data	yes	yes
semi-supervised data	(yes)	yes
unsupervised data	no	yes
prediction uncertainty	no	yes
theoretical guarantees	yes	(no)