Advanced course in machine learning
582744
Lecture 1

Lecturer: Arto Klami
Assistant: Aditya Jitta

Today

Motivation (Section 1)

Practicalities

Background

- Probability and statistics (Section 2)
- Linear algrebra (Matrix cookbook)

Machine Learning

what society thinks I do

what my friends think I do

what my parents think I do

$$L_P = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{l} \alpha_i y_i (\mathbf{x}_i \cdot \mathbf{w} + b) + \sum_{i=1}^{l} \alpha_i$$

$$\alpha_i \geq 0, \forall i$$

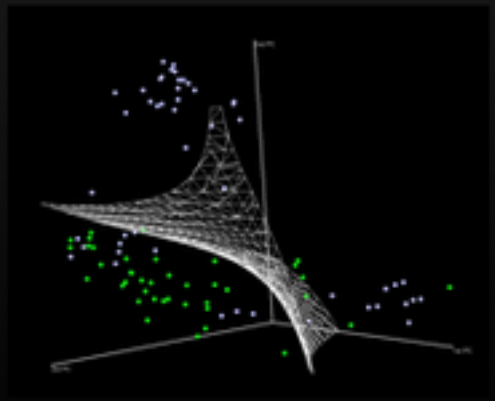
$$\mathbf{w} = \sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i, \sum_{i=1}^{l} \alpha_i y_i = 0$$

$$\nabla \hat{g}(\theta_t) = \frac{1}{n} \sum_{i=1}^{n} \nabla \ell(x_i, y_i; \theta_t) + \nabla r(\theta_t).$$

$$\theta_{t+1} = \theta_t - \eta_t \nabla \ell(x_{i(t)}, y_{i(t)}; \theta_t) - \eta_t \cdot \nabla r(\theta_t)$$

$$\mathbb{E}_{i(t)}[\ell(x_{i(t)}, y_{i(t)}; \theta_t)] = \frac{1}{n} \sum_i \ell(x_i, y_i; \theta_t).$$

what other programmers think I do



what I think I do

```
>>> from scipy import SVM
```

what I really do

Machine learning

- "Field of study that gives computers the ability to learn without being explicitly programmed" (A. Samuel, 1959)

Machine learning

- "Field of study that gives computers the ability to learn without being explicitly programmed" (A. Samuel, 1959)
- "...a set of methods that can automatically detect patterns in data, and then use the uncovered patterns to predict future data, or to perform other kinds of decision making under uncertainty" (K. Murphy)

Machine learning

- "Field of study that gives computers the ability to learn without being explicitly programmed" (A. Samuel, 1959)
- "...a set of methods that can automatically detect patterns in data, and then use the uncovered patterns to predict future data, or to perform other kinds of decision making under uncertainty" (K. Murphy)
- Subfield of artificial intelligence; covers the AI approaches that are based on learning
- Heavily relies on statistics, but emphasizes computation and general-purpose tools more

Machine learning

- "Field of study that gives computers the ability to learn without being explicitly programmed" (A. Samuel, 1959)
- "...a set of methods that can automatically detect patterns in data, and then use the uncovered patterns to predict future data, or to perform other kinds of decision making under uncertainty" (K. Murphy)
- Subfield of artificial intelligence; covers the AI approaches that are based on learning
- Heavily relies on statistics, but emphasizes computation and general-purpose tools more
- This course: Overview of the most important concepts with sufficient technical detail. Should be useful for a wide range of ML and data science careers

Machine learning - why?

- For many tasks, learning from data is easier than directly programming the solution

Machine learning - why?

- For many tasks, learning from data is easier than directly programming the solution
- How would you write an algorithm to recognize images of cats?

Machine learning - why?

- For many tasks, learning from data is easier than directly programming the solution
- How would you write an algorithm to recognize images of cats?
  - The programming approach: Manually describe the appearance of all possible cats
  - The learning approach: Show a lot of cat images to a learning algorithm

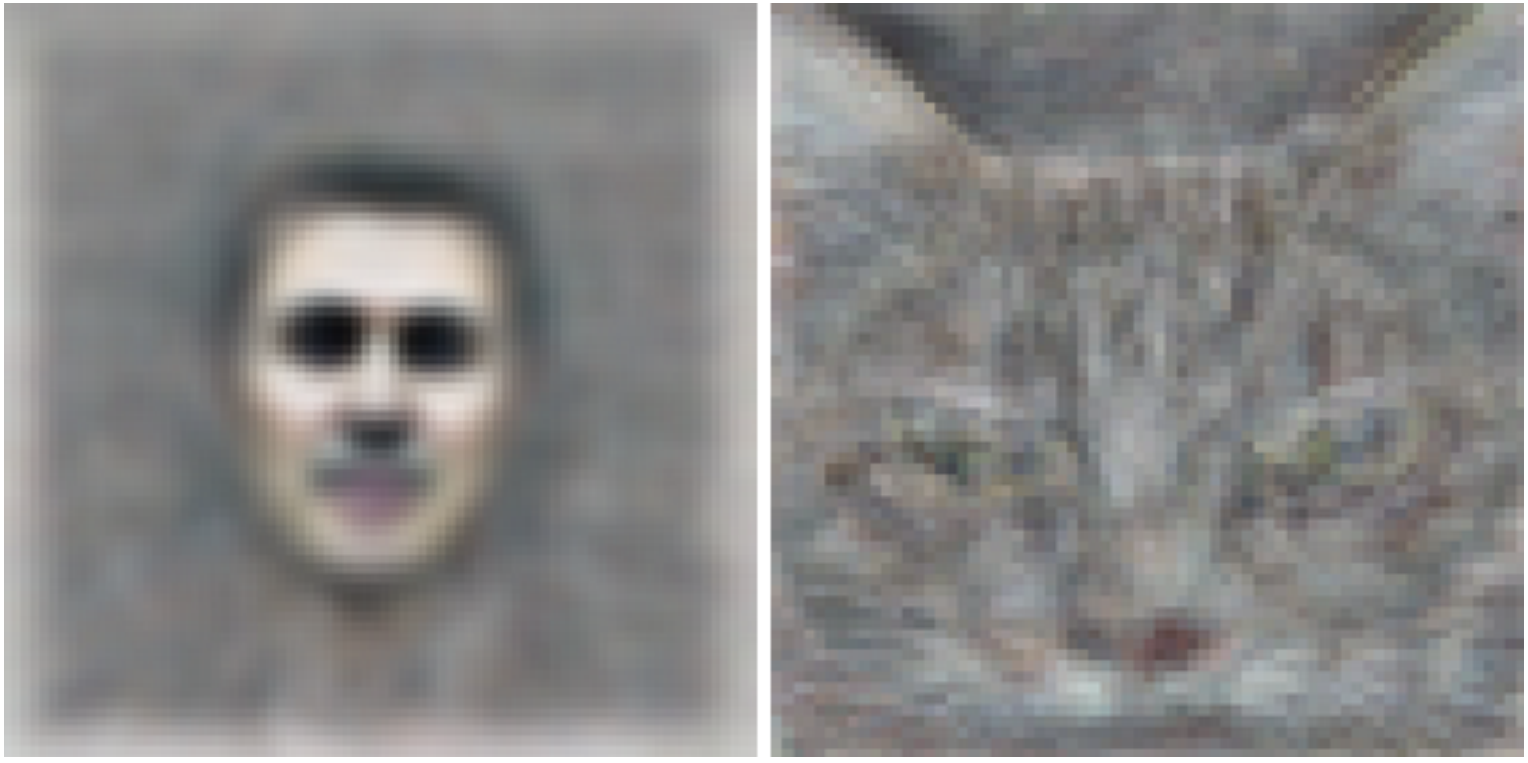
Machine learning - why?

- For many tasks, learning from data is easier than directly programming the solution
- How would you write an algorithm to recognize images of cats?
  - The programming approach: Manually describe the appearance of all possible cats
  - The learning approach: Show a lot of cat images to a learning algorithm
- Often understanding data is the actual task

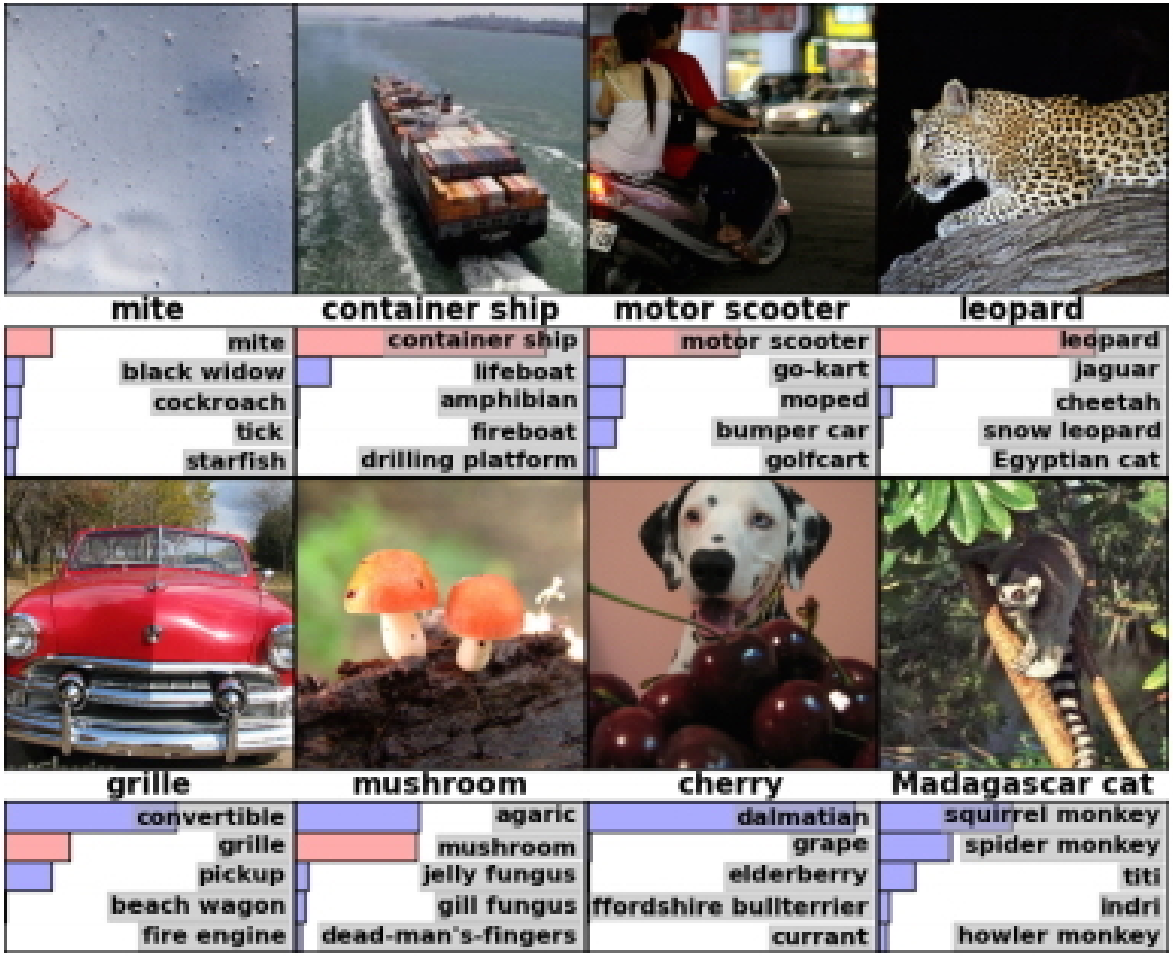
Machine learning - why?

- For many tasks, learning from data is easier than directly programming the solution
- How would you write an algorithm to recognize images of cats?
  - The programming approach: Manually describe the appearance of all possible cats
  - The learning approach: Show a lot of cat images to a learning algorithm
- Often understanding data is the actual task
  - Which genes are activated in a cancerous cell?
  - Is there a Higgs boson?
  - Where did all my money go to?

Google cat detector



Visual object classification



Krizhevsky et al.: ImageNet Classification with deep convolutional neural networks (2012)

Recommender engines

Leaderboard

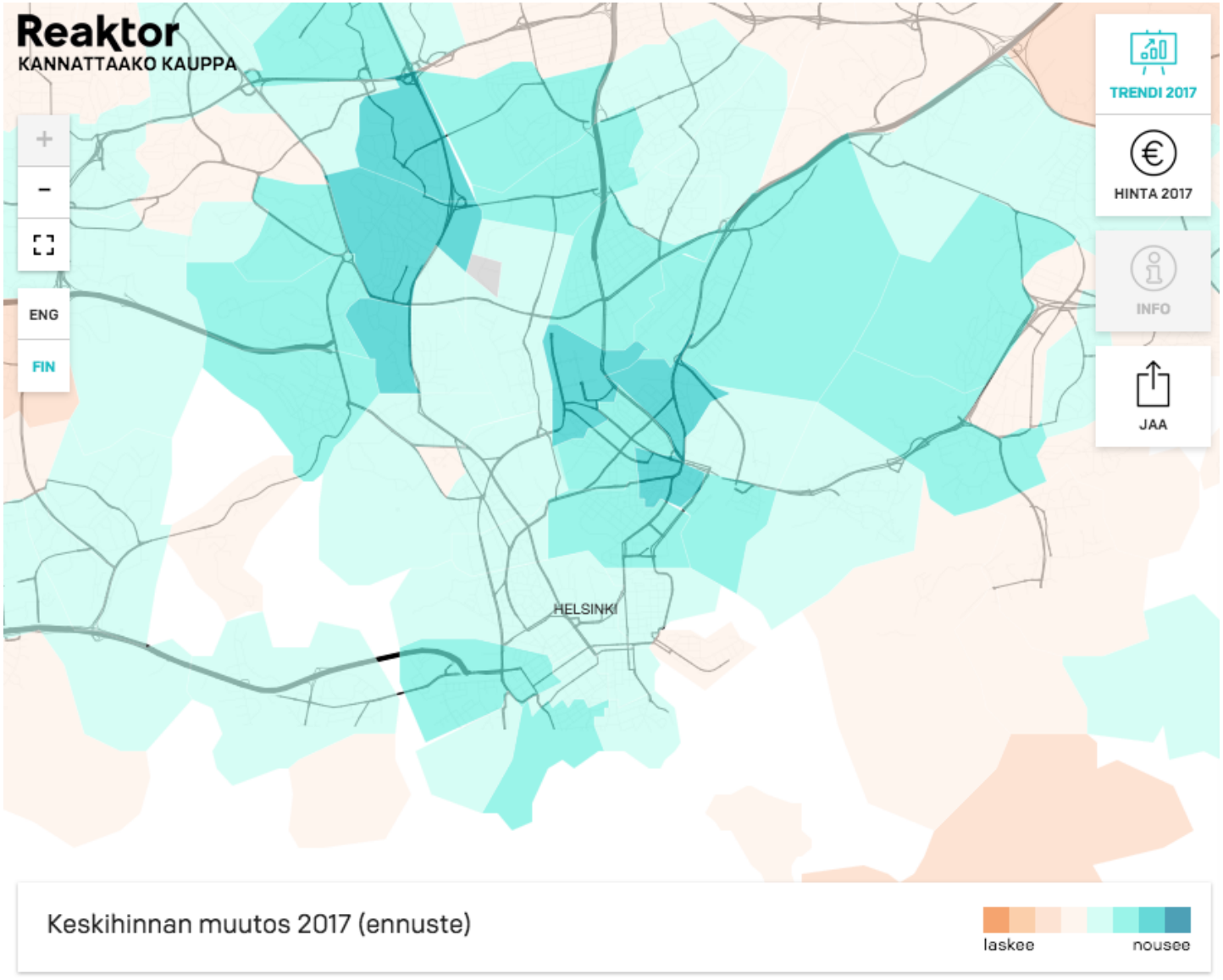
Showing Test Score. Click here to show quiz score

Display top 20 **leaders.**

Rank	Team Name	Best Test Score	% Improvement	Best Submit Time
Grand Prize - RMSE = 0.8567 - Winning Team: BellKor's Pragmatic Chaos				
1	BellKor's Pragmatic Chaos	0.8567	10.06	2009-07-26 18:18:28
2	The Ensemble	0.8567	10.06	2009-07-26 18:38:22
3	Grand Prize Team	0.8582	9.90	2009-07-10 21:24:40
4	Opera Solutions and Vandelay United	0.8588	9.84	2009-07-10 01:12:31
5	Vandelay Industries !	0.8591	9.81	2009-07-10 00:32:20
6	PragmaticTheory	0.8594	9.77	2009-06-24 12:06:56
7	BellKor in BigChaos	0.8601	9.70	2009-05-13 08:14:09
8	Dace	0.8612	9.59	2009-07-24 17:18:43

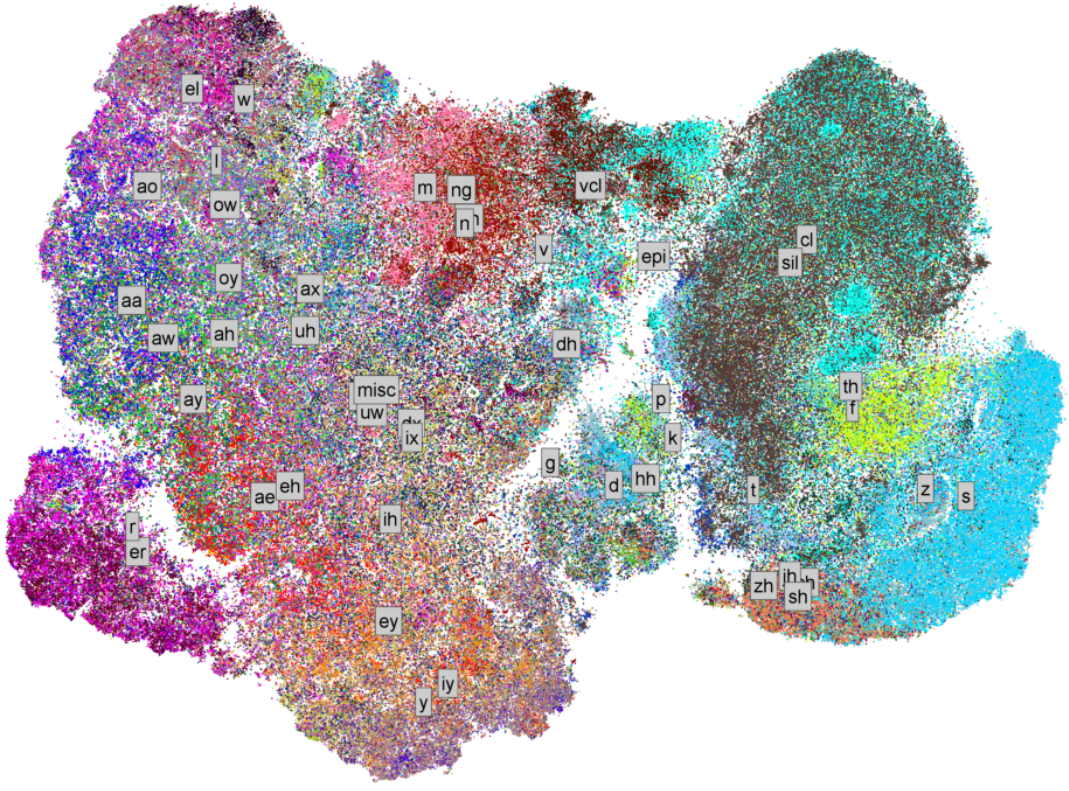
The Netflix Prize challenge

House price development



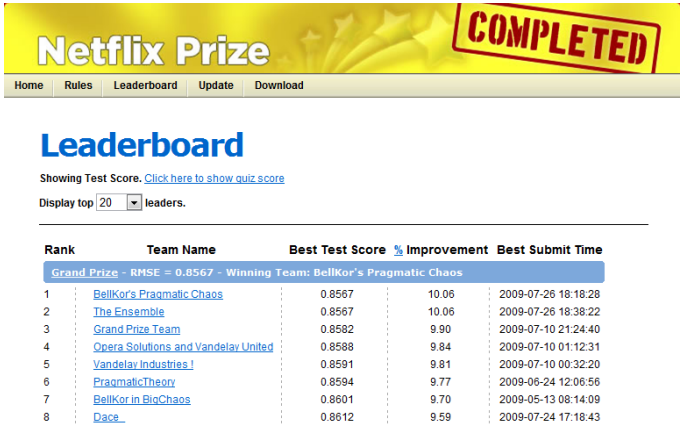
Reaktor: http://kannattaakokauppa.fi/

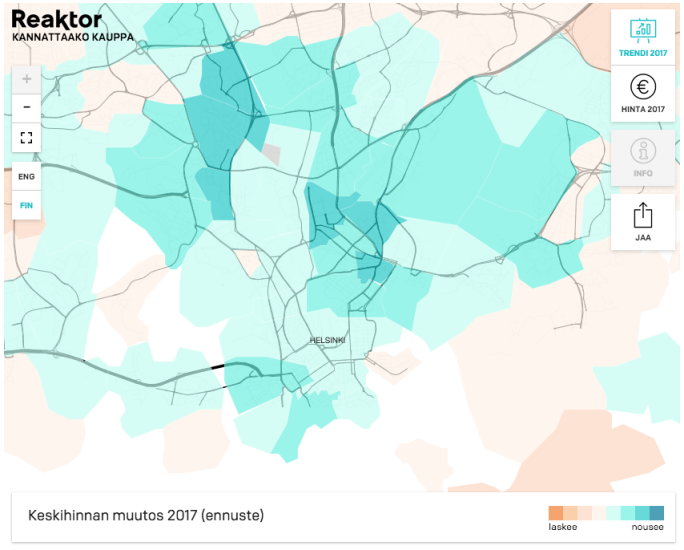
Visualization of phonemes

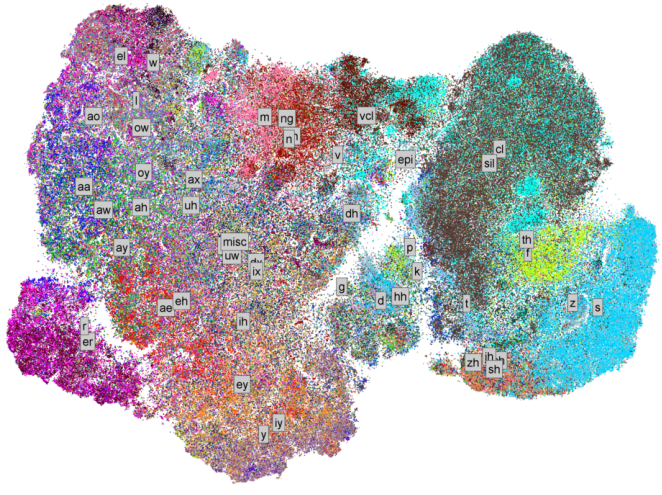


Yang et al. Scalable optimization of neighbor embedding for visualization (2013)

Which of these could you replicate after the course?







Schedule

- 13 lectures, every Tuesday (10-12) and Thursday (12-14) until May 3rd in this hall
- Exercise sessions on Wednesdays 10-12 in B221
- No lectures on March 24th and 29th
- Exam on May 11th (double-check before the exam)
- First separate exam June 16th

Prerequisites

- "Introduction to machine learning" or similar knowledge
- Having taken "Probabilistic models" also helps, as does background in statistical modeling
- All methods and algorithms presented from scratch, but some might be difficult to follow if the basics are not familiar
- Refresh knowledge on probability densities now if needed
- Linear algebra and optimization also useful

Prerequisites

- "Introduction to machine learning" or similar knowledge
- Having taken "Probabilistic models" also helps, as does background in statistical modeling
- All methods and algorithms presented from scratch, but some might be difficult to follow if the basics are not familiar
- Refresh knowledge on probability densities now if needed
- Linear algebra and optimization also useful

Note: (Kind of) replaces both "Unsupervised ML" and "Supervised ML" as a more general course, but cannot go as deep in either topic

Requirements and grading

- Exercises give 40% of the total points
- Exam gives 60% of the total points
- Minimum of 50% required for both parts
- Numerical grading based on the total points

Requirements and grading

- Exercises give 40% of the total points
- Exam gives 60% of the total points
- Minimum of 50% required for both parts
- Numerical grading based on the total points

Alternative: Separate exam combined with a small project. The project emulates the exercises, but might require a bit more effort.

Exercises

- Exercise sessions: Demonstrations, examples, guidance, group-work, model solutions, ...
- Solutions returned by email
- Two types of exercises:
  1. Pen-and-paper derivations or verbal questions
  2. Computer exercises; implement details of an algorithm or perform some simple experiments.
- This is not a programming course: The code quality does not influence grading (unless so bad the code cannot be evaluated)
- ...and you can use a language of your choice, within reason (at least Python, R, Matlab and Julia are okay)
- The mathematical derivations and verbal answers should be clear and well formulated

Exercise schedule

- Exercises released on Tuesday
- Due in one week, model solutions released and presented in the next exercise session
- ...or would you prefer two?
- In case you miss some, a small set of extra exercises will be given near the end of the course so you can compensate for that

Course material

- Course book: "Machine Learning: a Probabilistic Perspective" by Kevin P. Murphy
- The library has ordered five copies, but they are not available yet
- Covers almost everything on the course; when other material is needed it is freely available
- The book is a good reference, but not really a course book. However, getting familiar with this kind of presentation style is useful since it is similar to how scientific papers are written

Course material

- Course book: "Machine Learning: a Probabilistic Perspective" by Kevin P. Murphy
- The library has ordered five copies, but they are not available yet
- Covers almost everything on the course; when other material is needed it is freely available
- The book is a good reference, but not really a course book. However, getting familiar with this kind of presentation style is useful since it is similar to how scientific papers are written
- Alternative: A collection of freely available sources cover the same topics but several sources are required to cover the whole course. I will do my best to guarantee that the exercises and exam can be answered also based on these, but reading all this material will take more effort.

Learning goals

After the course you will be:

- Familiar with probabilistic formulation for machine learning
- Familiar with the main supervised and unsupervised learning problems and the typical solutions
- Able to solve typical supervised and unsupervised learning problems with competitive performance (though we will not cover the software as such)
- Able to implement basic machine learning methods from scratch
- Able to recognize and avoid over-fitting and other similar problems
- Able to recognize some more advanced learning setups

Topics

- Linear algebra, probabilities and optimization – the tools of ML
- Unsupervised learning: Clustering, dimensionality reduction, visualization
- Supervised learning: Classification and regression, kernel methods, decision trees, boosting
- Neural networks and deep learning
- Evaluation and development of ML models
- Some Bayesian inference, if time permits

Not covered on this course

- Bayesian networks (Section 10)
- Exact inference in graphical models (Section 20)
- Structure learning for graphical models (Section 26)
- Nonparametric Bayesian statistics (Section 15)
- State-space models, sequence models (Sections 17-19)
- Generalized linear models (Section 9)
- Reinforcement learning (not in the book either)
- Software libraries for ML
- Feature engineering, or other "data science" tasks

Necessary background

The two branches of math we need

- Probabilities and statistics
- Linear algrebra

We go through these very quickly today

Sources for refreshing your knowledge:

- Section 2 of MLaPP
- The matrix cookbook
- Undergrad math books
- Section 2 of UML lecture notes:
  `http://www.cs.helsinki.fi/u/ahyvarin/teaching/uml2015/uml2015_lecturenotes.pdf`

Probability

On this course we can forget the measure-theoretic probability theory (σ-algebras and all that). We just use probabilities as tools:

- Basic concept: Random variable X with uncertain value
- If X is discrete: *probability mass function* $p(X = x)$ tells the probability of an event x
- $p(X = x) \geq 0$, $\sum_x p(X = x) = 1$
- For disjoint a and b,
  $p(X = a \text{ or } X = b) = p(X = a) + p(X = b)$
- Joint probability $p(X_1 = a, X_2 = b) = p(X_1 = a \text{ and } X_2 = b)$

Discrete random variables

Joint probability $p(X_1 = a, X_2 = b)$

- Marginalization:
  $p(X_1 = a) = \sum_b p(X_1 = a|X_2 = b)p(X_2 = b)$
- Product/chain rule:
  $p(X_1 = a, X_2 = b, X_3 = c) = p(X_1 = a)p(X_2 = b|X_1 = a)p(X_3 = c|X_1 = a, X_2 = b)$
- ...and is valid for all orders
- Conditional probability: $p(X_1 = a|X_2 = b) = \frac{p(X_1=a,X_2=b)}{p(X_2=b)}$
- Bayes' rule: $p(X_1 = a|X_2 = b) = \frac{p(X_2=b|X_1=a)p(X_1=a)}{p(X_2=b)}$

You really need to understand these!

Discrete random variables

Joint probability $p(X_1, X_2)$

- Marginalization: $p(X_1) = \sum_{X_2} p(X_1|X_2)p(X_2)$
- Product/chain rule:
  $p(X_1, X_2, X_3) = p(X_1)p(X_2|X_1)p(X_3|X_1, X_2)$
- ...and is valid for all orders
- Conditional probability: $p(X_1|X_2) = \frac{p(X_1,X_2)}{p(X_2)}$
- Bayes' rule: $p(X_1|X_2) = \frac{p(X_2|X_1)p(X_1)}{p(X_2)}$

You really need to understand these!

Discrete random variables

Useful concepts

- Independence: $X \perp\!\!\!\perp Y \quad : \quad p(X, Y) = p(X)p(Y)$
- Expectations: $\mathbb{E}[f(X)] = \sum_x f(x)p(X = x)$
- Mean: $f(X) = X$
- Variance: $f(X) = (X - \mathbb{E}[X])^2$
- Covariance: $f(X, Y) = (X - \mathbb{E}[x])(Y - \mathbb{E}[Y])$
- Correlation: Covariance normalized to $[-1, 1]$

Expectation is a linear operator:
$\mathbb{E}[a \times f(X) + b \times g(Y)] = a \times \mathbb{E}[f(X)] + b \times \mathbb{E}[g(Y)]$

Information theory

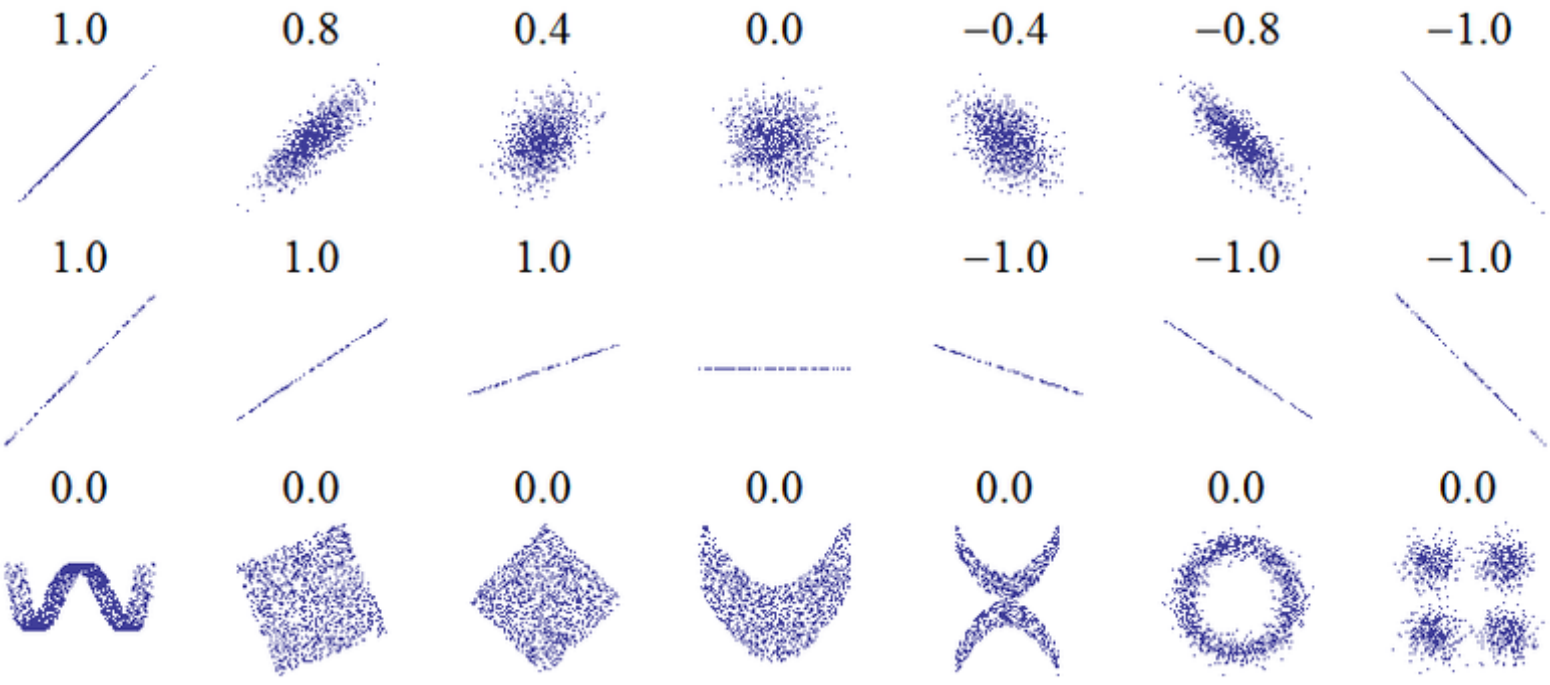
Measures characterizing the uncertainty and dependency of the whole distribution

- Entropy: $H(X) = -\sum_x p(X = x) \log p(X = x)$
- Mutual information: $I(X, Y) = H(X) + H(Y) - H(X, Y)$

High entropy: We are very uncertain about the outcome

High mutual information: $p(X|Y)$ has low entropy.
$I(X, Y) = 0$ if and only if $X \perp\!\!\!\perp Y$.

Correlation and independence



Discrete probability distributions

- Bernoulli distribution: $\text{Ber}(x|p) = p^{I(x=1)}(1-p)^{I(x=0)}$
- Binomial distribution: Sum of N independent draws from Bernoulli
- Categorical distribution: Generalize Bernoulli for K different outcomes
- Multinomial distribution: Sum of N independent draws from the categorical distribution
- Poisson distribution: $\text{Poi}(\lambda) = \exp^{-\lambda} \frac{\lambda^x}{\lambda!}$

You can always check the exact equations from a reference, but you need to understand what each distribution means and how they are related.

Continuous random variables

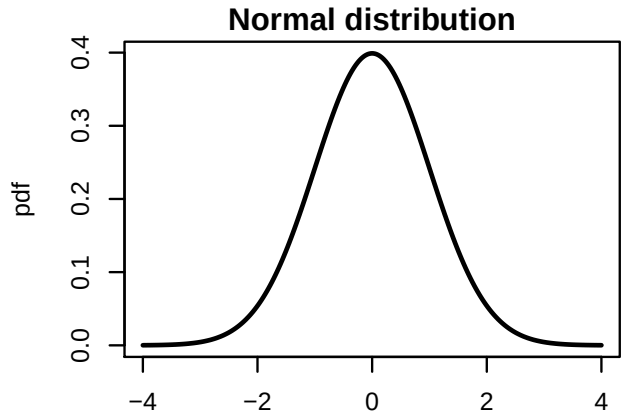
Everything presented above generalizes for random variables that take continuous output values

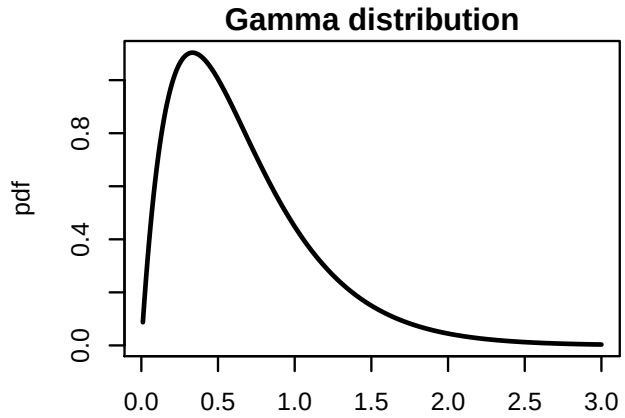
- Probability mass function replaced by *probability density function* (pdf) $p(x)$
- $p(x) \geq 0$, $\int p(x)dx = 1$
- Note: $p(x)$ may be above one
- All summations become integrals
- Expectation: $\mathbb{E}[f(x)] = \int_x f(x)p(x)dx$
- Cumulative density function: $P(X \leq a) = \int_{x=-\infty}^{a} p(x)dx$

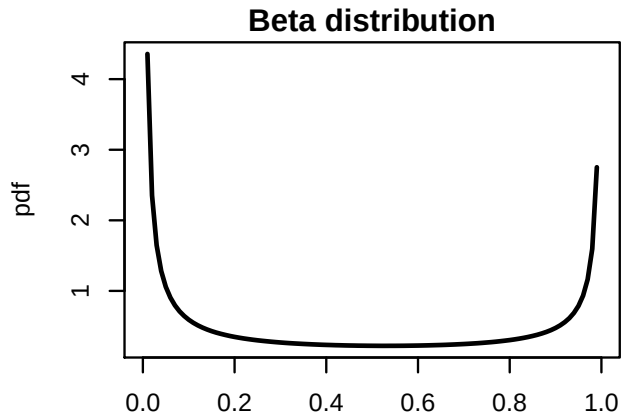
Probability distributions

Probability distributions, univariate:

- Normal: Real values centered around a given mean value, dispersion controlled by the variance
- Gamma: Positive real values centered around shape/rate, variance decreases for large rates
- Beta: Real values within $[0, 1]$, mean given by $\frac{\alpha}{\alpha+\beta}$







Probability distributions

Probability distributions, multivariate:

- Normal: Each marginal density is a univariate normal distribution, but the different variables can correlate. The correlations given by the covariance matrix $\mathbf{\Sigma}$
- Dirichlet: Multivariate generalization of the beta distribution. Takes values over the probability simplex; $x_d \geq 0$ and $\sum_d x_d = 1$

Multivariate normal distribution is important. Learn also the pdf: $p(\mathbf{x}|\boldsymbol{\mu}, \mathbf{\Sigma}) = \frac{1}{(2\pi)^{D/2}|\mathbf{\Sigma}|^{1/2}} \exp(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \mathbf{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}))$

With diagonal covariance it is equivalent to a product of D independent univariate normal distributions

Living with probabilities

The tricks of the trade:

- We often need $\prod_{n=1}^{N} p(x_n)$, which quickly tends to zero (Try it out: What is 0.01^{200}?)
- $\log \prod_n p(x_n) = \sum_n \log p(x_n)$ behaves much better
- Hence, almost all computation done in the log-space
- Normalization of probabilities: $p(X = x) = \frac{f(x)}{\sum_y f(y)}$
- Normalization in the log-space: $\log \sum_n p(x_n) = \log \sum_n \exp(\log p(x_n)) =$ $\log \sum_n \exp(\log p(x_n) - A) + A$ (see Section 3.5.3)

You will need these in many of the computer exercises

Linear algebra

Basic concepts:

- $\mathbf{x} \in \mathbb{R}^R$ is a R-dimensional column vector
- $\mathbf{X} \in \mathbb{R}^{R \times C}$ is a matrix with R rows and C columns
- Transpose: $\mathbf{A}^T_{i,j} = \mathbf{A}_{j,i}$
- Matrix product: $\mathbf{AB} = \mathbf{D}$, where $\mathbf{D}_{i,j} = \sum_{k=1}^{C_A} \mathbf{A}_{i,k}\mathbf{B}_{k,j}$
- ...and hence C_A needs to equal R_B
- Matrix-vector product $\mathbf{Ax}$ follows trivially
- Outer and inner products for vectors: $\mathbf{x}\mathbf{x}^T$ (a $R \times R$ matrix) and $\mathbf{x}^T\mathbf{x}$ (a scalar)

Linear algebra

Basic concepts:

- Determinant ($\approx$ absolute value): $\det(\mathbf{A}) = |\mathbf{A}| = \prod \lambda_i$, where λ_i are the eigenvalues
- $|\mathbf{A}^{-1}| = 1/|\mathbf{A}|$, $|\mathbf{A}| = |\mathbf{A}^T|$, $|\mathbf{AB}| = |\mathbf{A}||\mathbf{B}|$, ...
- Inverse: $\mathbf{A}^{-1}\mathbf{A} = \mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$, only for square matrix and exists only if $\det(\mathbf{A}) \neq 0$
- Trace: Sum of the diagonal elements of a square matrix

See the Matrix cookbook for an excellent resource

Linear algebra

Matrix decompositions:

- eigen-value decomposition: $\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{-1}$, $\mathbf{\Lambda}$ is a diagonal matrix of eigenvalues λ_i and $\mathbf{Q}$ is an orthonormal matrix of eigenvectors. Only for diagonal matrices.
- Singular value decomposition (SVD): Generalizes the above for arbitrary matrices. $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$
- Cholesky decomposition: $\mathbf{A} = \mathbf{L}\mathbf{L}^T$, where $\mathbf{L}$ is a lower triangular matrix. Useful for operating with covariance matrices, applicable for positive definite matrices.
- Positive definity: All eigenvalues are positive.

No need to know the algorithms for computing these, but you should know what they do and what they can be used for

What for?

- Linear algebra used for compact notation of linear operators
- ...and we can do a lot of learning with linear operators
- Matrix product tells how linear operators are combined: If $\mathbf{y} = \mathbf{A}\mathbf{x}$ and $\mathbf{z} = \mathbf{B}\mathbf{y}$ then $\mathbf{z} = \mathbf{B}\mathbf{A}\mathbf{x}$
- Determinant tells how the volume changes: If we transform a unit cube with $\mathbf{A}$ the volume of the transformed cube is $|\mathbf{A}|$
- The factorization techniques are needed for computing inverses, determinants etc.