TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

12 Semester Recap & Outlook

Machine Learning 1: Foundations

Marius Kloft *(TUK)*

Lecture 1: What is machine learning?

Arthur Samuel, 1959

"Field of study that gives computers the ability to **learn** [from data] without being explicitly programmed"

Aim

- Aim is to write a computer program that uses
  - the data instances $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$ (called "inputs")
  - and their respective annotations

    $y_1, \ldots, y_n \in \mathcal{Y} \overset{\text{e.g.}}{=} \{-1, +1\}$ (called "labels")

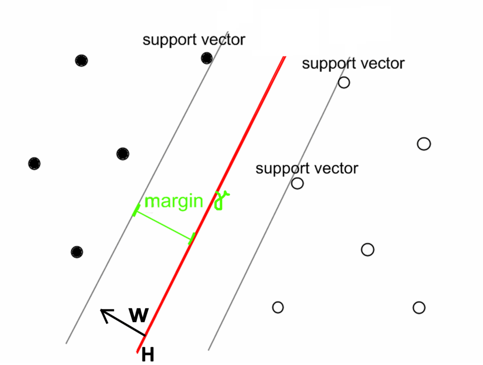
  to compute a function $f : \mathbb{R}^d \to \mathcal{Y}$
  - that predicts for new instances $\mathbf{x}$ the correct label y

Examples

Computer Security

Learn from source code or network traffic $\mathbf{x}_1, \ldots, \mathbf{x}_n$ to discriminate benign code ($y = -1$) from malicious code ($y = +1$)

Lecture 2: Linear Support Vector Machines

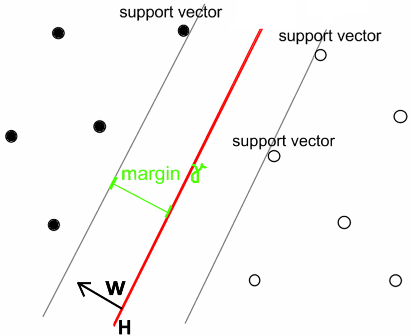


Hard-Margin SVM

$$\max_{\gamma, H_{\mathbf{w}}} \quad \gamma \quad \text{s.t.} \quad \gamma \leq y_i d(\mathbf{x}_i, H_{\mathbf{w}}) \quad \text{for all } i = 1, \ldots n$$

- Find hyperplane with maximum margin γ
- such that inputs lie outside of the margin

Lecture 2: Linear Support Vector Machines

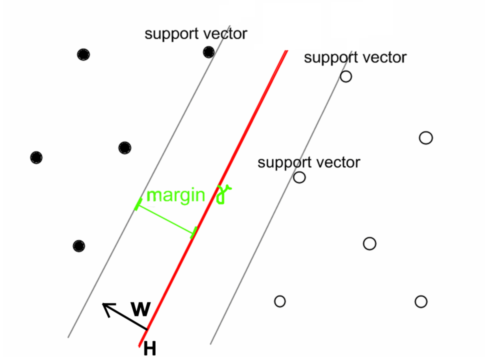


Hard-Margin SVM

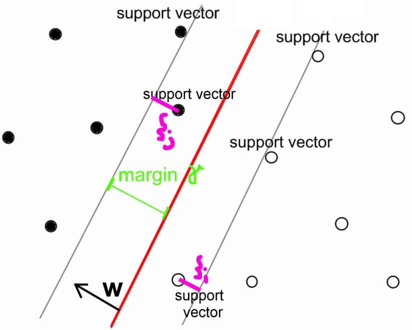
$$\max_{\mathbf{w}\in\mathbb{R}^d, b\in\mathbb{R}} \boxed{\frac{1}{2}\|\mathbf{w}\|^2} \quad \text{s.t.} \quad \boxed{1} \leq y_i\left(\mathbf{w}^\top \mathbf{x}_i + b\right) \quad \text{for all } i = 1, \ldots n$$

(equivalently)

Lecture 2: Linear Support Vector Machines



Lecture 2: Linear Support Vector Machines

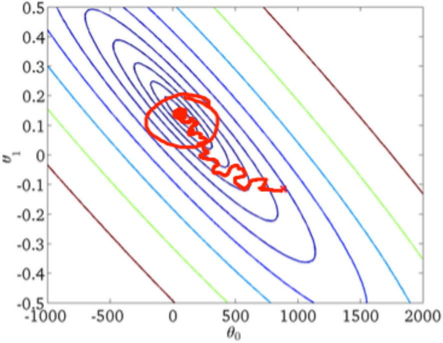


Soft-Margin SVM

$$\min_{b\in\mathbb{R},\mathbf{w}\in\mathbb{R}^d} \quad \frac{1}{2}\|\mathbf{w}\|^2 \; + \; C\sum_{i=1}^{n}\xi_i$$

Where we allow for some violations of the margin: $\xi_i := \max\left(0, 1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b)\right)$

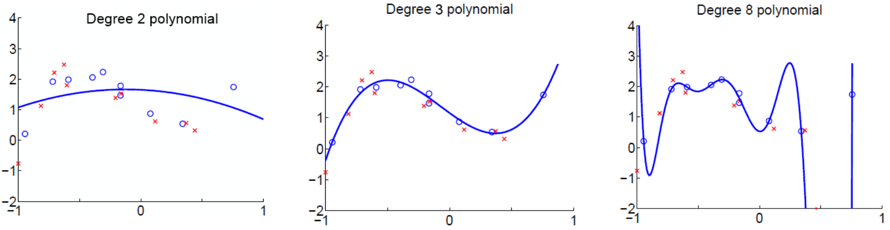
Lecture 3: Convex Optimization via SGD



Lecture 4: Kernels Methods

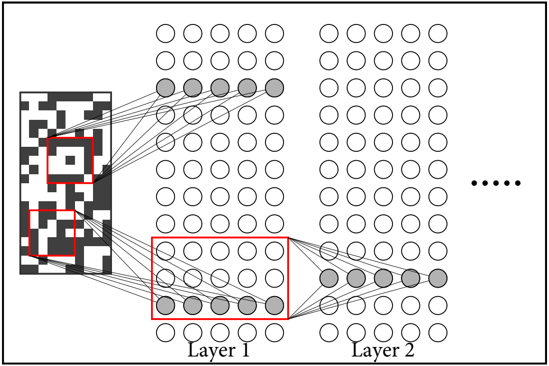
Kernel Trick

- Substitute all occurrences of scalar products $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ in SVM by kernel $k(\mathbf{x}_i, \mathbf{x}_j)$
- E.g., polynomial kernel $k(\mathbf{x}_i, \mathbf{x}_j) := (\langle \mathbf{x}_i, \mathbf{x}_j \rangle + b)^m$
- Makes linear learning algorithms non-linear

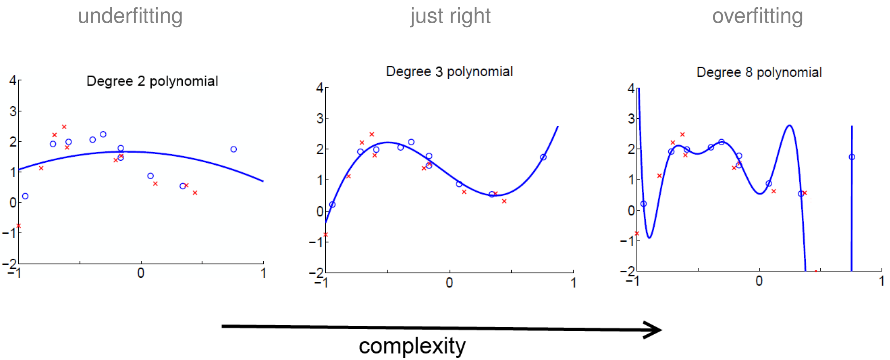


Lectures 5 and 6: Deep Learning

$$\min_{\mathbf{w},W} \quad \frac{1}{2}\|\mathbf{w}\|^2 + \frac{1}{2}\sum_{l=1}^{L}\|W_l\|_{\text{Fro}}^2 + C\sum_{i=1}^{n}\ln(1+\exp(-y_i\mathbf{w}^\top\phi_W(\mathbf{x}_i)))$$



Lecture 7: Overfitting & Regularization

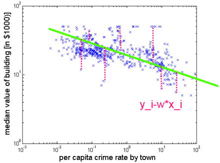


Avoiding overfitting by proper model selection and regularization.

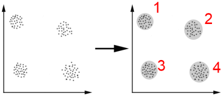
- Regularization smoothens the prediction function (making it less complex)

Lectures 8–10: Beyond binary classification:

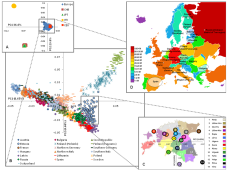
L8: Regression



L9: Clustering

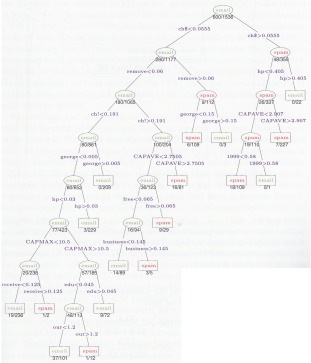


L10: Dimensionality Reduction



All those algorithms can be kernelized and “deepified”!

Lecture 11: Random Forests



Unifying View of Regr., Classif., and Dim. Reduction

$$\min_{[W,]\ b,\mathbf{w}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} l\big(f(\mathbf{x}_i)\ [, y_i]\big) \quad \left[+\frac{1}{2}\sum_{l=1}^{L}\|W_l\|_{\mathrm{Fro}}^2\right],$$

with model

- $f(\mathbf{x}) = \langle \mathbf{w}, \phi(\mathbf{x})\rangle + b$ for regression and classification
- $f(\mathbf{x}) = \left\|\phi(\mathbf{x}) - \mathbf{w}\mathbf{w}^\top\phi(\mathbf{x})\right\|^2$ for dimensionality reduction

and loss

- $l(t, y) := \max(0, 1 - yt)$ for SVM ("hinge loss")
- $l(t, y) := \ln(1 + \exp(-yt))$ for LR and ANN ("logistic loss")
- $l(t, y) := (t - y)^2$ for regression
- $l(t) := t$ for dimensionality reduction

and feature map

- $\phi := \mathrm{id}$ for linear SVM, linear LR, RR, and PCA.
- $\phi := \phi_k$ for kernel SVM, kernel LR, KRR, and KPCA.
- $\phi := \phi_W$ for ANN, DR, and AE.

Grey terms only for neural methods, blue terms not for dimensionality reduction

Limitation: k-means and random forests do not fit into unifying view

ML1 in One Table

ML1 methods as special cases of the equation:

loss \ map	linear	kernel	neural
hinge	linear SVM	SVM	deep SVM
logistic	log. regression (LR)	kernel LR	DNN
squared	RR	KRR	deep regression
reconstruction	PCA	kPCA	autoencoder

The **Statistical** Dimension

We have seen that ML1 can be summarized as a 2D table with the following dimensions:

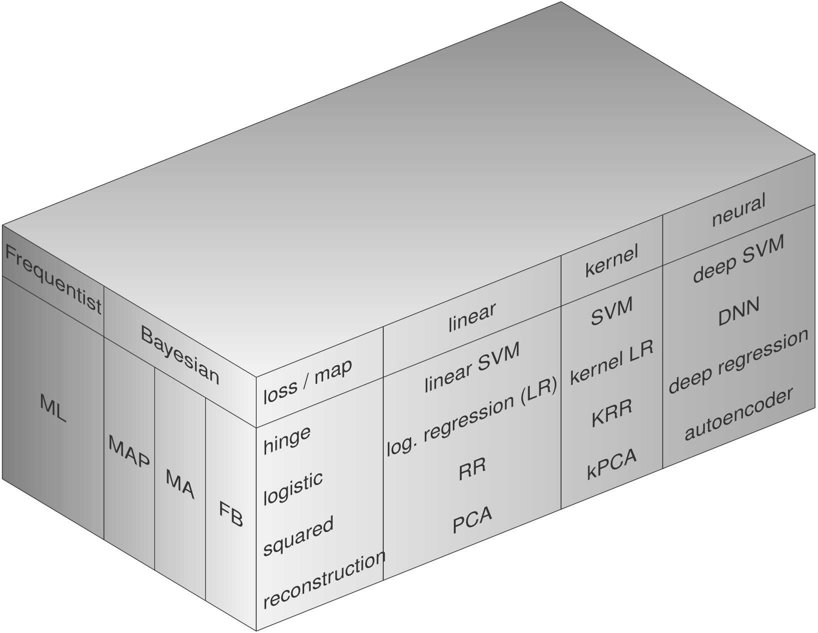
- Feature map (lin, kernel, neural)
- Loss function (hinge, logistic, ...)

In ML2, we learn that there is another dimension:

- Statistical approach

This dimension has the four possible choices, of which we used **only one** in ML1!

ML2: Extend 2D Table into 3D Table



Topics in ML2

- Statistical ML
- Frequentist vs Bayesian approach
- Bayes rule and Bayes classifier
- Gaussian processes
- Bayesian neural networks
- EM algorithm
- Probabilistic PCA
- Variational autoencoders
- ...and more!

Hope you enjoyed ML1 and will enjoy ML2 :)