Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Universität Zürich[UZH]

Machine Learning, Fall Semester 2014

Summary of the lectures in 2014

Diana Ponce-Morado
Joachim Ott
Imanol Studer
Benjamin Ellenberger

Monday 22nd December, 2014

Professor:
Prof. Dr. Joachim Maximilian Buhmann
Institute of Neuroinformatics, UZH

Contents

1 Representations

This is the chapter on Representations.

2 Measurements and Data

Patterns

Data Types, Transformations, Scale

3 Regression

This is the chapter on Regression.

Linear Regression

Ridge Regression

LASSO

Nonlinear Regression by basis expansion

Wavelet regression

Bias variance Tradeoff

Gaussian Processes

4 Numerical Estimation Techniques

This is the chapter on Numerical Estimation Techniques.

Cross-Validation

Bootstrap

Jackknife

Hypothesis Testing

5 Classification

This is the chapter on Classification.

Problem Setting for Bayesian Inference

Bayes Rule

Parametric Models, Bayesian Learning

6 Parametric Models

This is the chapter on Parametric Models.

Maximum Likelihood Method

Efficient Estimators

Bayesian Learning (batch/online)

7 Design of Linear Discriminant Functions

This is the chapter on Linear Discriminant Functions.

Perceptrons

Fisher's linear discriminant analysis

8 Support Vector Machines

This is the chapter on Support Vector Machines.

Lagrangian optimization theory

Hard margin SVMs

Soft margin SVMs

9 Nonlinear Support Vector Machines

This is the chapter on Nonlinear Support Vector Machines.

10 Ensemble Methods for Classifier Design

This is the chapter on Regression.

PAC Learning

Bagging

Boosting

Arcing

Exponential Loss

11 Unsupervised Learning

This is the chapter on Unsupervised Learning.

Nonparametric Density Estimation

Histograms

Parzen Estimators

k-Nearest Neighbor Estimator

12 Neural Networks

This is the chapter on Neural Networks.

Motivation by Computational Neuroscience

Multilayer Perceptrons and Backpropagation

NETtalk and ALVINN

Boltzmann machines

Deep Neural Networks

13 Mixture Models

This is the chapter on Mixture Models.

k-Means Algorithm

Mixture Models

Expectation Maximization Algorithm

Convergence Proof of EM Algorithm

14 Cheat sheet

Probability

Probability Rules

Sum Rule	$P(X = x_i) = \sum_{j=1}^{J} p(X = x_i, Y = y_i)$
Product rule	$P(X,Y) = P(Y\mid X)P(X)$
Independence	$P(X,Y) = P(X)P(Y)$
Bayes' Rule	$P(Y\mid X) = \frac{P(X\mid Y)P(Y)}{P(X)}$
Conditional independence	$X \perp Y \mid Z$
	$P(X,Y\mid Z) = P(X\mid Z)P(Y\mid Z)$
	$P(X\mid Z,Y) = P(X\mid Z)$

Expectation

$$\begin{aligned}
E(X) &= \int_{\inf}^{\inf} xp(x)dx \\
\sigma^2(X) &= E(x^2) - E(x)^2 \\
\sigma^2(X) &= \int_x (x-\mu_x)^2 p(x)dx \\
Cov(X,Y) &= \int_x \int_y p(x,y)(x-\mu_x)(y-\mu_y)dxdy
\end{aligned}$$

Gaussian

$$p(X\mid\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

Kernels

Requirements: Symmetric ($k(x,y) = k(y,x)$)
positive semi-definite K.

$$\begin{aligned}
k(x,y) &= ak_1(x,y) + bk_2(x,y) \\
k(x,y) &= k_1(x,y)k_2(x,y) \\
k(x,y) &= f(x)f(y) \\
k(x,y) &= k_3(\varphi(x), \varphi(y))
\end{aligned}$$

Linear	$k(x,y) = x^\top y$
Polynomial	$k(x,y) = (x^\top y + 1)^d$
Gaussian RBF	$k(x,y) = \exp\left(\frac{-\lVert x-y\rVert_2^2}{h^2}\right)$
Sigmoid (Neural Net)	$k(x,y) = \tanh(kx^\top y - b)$

Regression

Linear Regression:

$$\min_w \sum_{i=1}^{n} (y_i - w^\top x_i)^2$$

Closed form solution: $w^* = (x^\top x)^{-1} x^\top y$
Ridge Regression:

$$\min_w \sum_{i=1}^{n} (y_i - w^\top x_i)^2 + \lambda \lVert w\rVert_2^2$$

Closed form solution: $w^* = (x^\top x + \lambda I)^{-1} x^\top y$
Lasso Regression (sparse):

$$\min_w \sum_{i=1}^{n} (y_i - w^\top x_i)^2 + \lambda \lVert w\rVert_1$$

Kernelized Linear Regression:

$$\min_\alpha \lVert K\alpha - y\rVert_2^2 + \lambda \alpha^\top K \alpha$$

Closed form solution: $\alpha = (K - \lambda I)^{-1} y$

Classification

0/1 Loss	$w^* = \operatorname{argmin}_w \sum_{i=1}^{n} [y_i \neq sign(w^\top x_i)]$
Perceptron	$w^* = \operatorname{argmin}_w \sum_{i=1}^{n} [\max(0, y_i w^\top x_i)]$

SVM

Primal, constrained:

$$\min_w w^\top w + C \sum_{i=1}^{n} \xi_i, \text{ s.t. } y_i w^\top x_i \geq 1 - \xi_i, \xi_i \geq 0$$

Primal, unconstrained:

$$\min_w w^\top w + C \sum_{i=1}^{n} \max(0, 1 - y_i w^\top x_i) \text{ (hinge loss)}$$

Dual:

$$\max_\alpha \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i^\top x_j, \text{ s.t. } 0 \geq \alpha_i \geq C$$

Dual to primal: $w^* = \sum_{i=1}^{n} \alpha_i^* y_i x_i$, $\alpha_i > 0$: support vector.

Kernelized SVM

$$\max_\alpha \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j k(x_i, x_j), \text{ s.t. } 0 \geq \alpha_i \geq C$$

Classify: $y = sign(\sum_{i=1}^{n} \alpha_i y_i k(x_i, x))$

Misc

Lagrangian: $f(x,y) s.t. g(x,y) = c$

$$\mathcal{L}(x,y,\gamma) = f(x,y) - \gamma(g(x,y) - c)$$

Parametric learning: model is parameterized with a finite set of parameters, like linear regression, linear SVM, etc.

Nonparametric learning: models grow in complexity with quantity of data: kernel SVM, k-NN, etc.

Probabilistic Methods:

MLE

Least Squares, Gaussian Noise

$$L(w) = -\log(P(y_1...y_n \mid x_1...x_n, w)) = \frac{n}{2}\log(2\pi\sigma^2) + \sum_{i=1}^{n} \frac{(y_i - w^\top x_i)^2}{2\sigma^2}$$

$$\operatorname{argmax}_w P(y\mid x,w) = \operatorname{argmin}_w L(w) = \operatorname{argmin}_w \sum_{i=1}^{n} (y_i - w^\top x_i)^2$$

MAP

Ridge regression, Gaussian prior on weights

$$\underset{w}{\operatorname{argmax}} P(w)\prod_i^n P(y_i|x_i,w) = \underset{w}{\operatorname{argmin}} \frac{1}{2\sigma^2}\sum_{i=1}^n (y_i - w^\top x_i) + \frac{1}{2\beta^2}\sum_{i=1}^n w_i^2$$

$P(w)$ or $P(\theta)$ - conjugate prior (beta, Gaussian) (posterior same class as prior)
$P(y_i|\theta)$ - likelihood function (binomial, multinomial, Gaussian)
Beta distribution: $P(\theta) = Beta(\theta;\alpha_1,\alpha_2) \propto \theta^{\alpha_1-1}(1-\theta)^{\alpha_2-1}$

Logistic Regression

MLE with Bernoulli noise

$$\begin{aligned}\text{MLE: } \underset{w}{\operatorname{argmin}} L(w) &= \sum_{i=1}^n \log(1+\exp(-y_i w^\top x_i)) \\ \text{MAP: } &+ \left\{\lambda\|w\|_2^2, \lambda\|w\|_1\right\}\end{aligned}$$

Classification: $P(y|x,\hat{w} = \frac{1}{1+\exp(-y\hat{w}^\top x)}$

Bayesian Decision Theory

$$a^* = \underset{a\in A}{\operatorname{argmin}} E_y[C(y,a)|x]$$

Bayesian Model Averaging (BMA)

Ridge regression, but with probabilities

$$\begin{aligned}P(y|x,D) &= \int P(y|x,w)P(w|D)dw \\ P(w) &= \mathcal{N}(w;0,\sigma_w^2) \\ P(y|w,x) &= \mathcal{N}(y;wx,\sigma_y^2) \\ P(w|x,y) &= \mathcal{N}(w;\mu_{w|y},\sigma_{w|y}^2)\end{aligned}$$

$$\mu_{w|y} = \frac{xy\sigma_w^2}{x^2\sigma_w^2+\sigma_y^2}$$
$$\sigma_{w|y}^2 = \frac{\sigma_w^2\sigma_y^2}{x^2\sigma_w^2+\sigma_y^2}$$

$$\begin{aligned}\text{MAP}P(y'|x',\hat{w}) &= \mathcal{N}(y';x'\mu_{w|y},\sigma_y^2) \\ \text{BMA}P(y'|x',x,y) &= \mathcal{N}(y';x'\mu_{w|y},\sigma_y^2 + x'^2\sigma_{w|y}^2)\end{aligned}$$

Bayesian Linear Regression

$$\begin{aligned}\mathcal{N}(x_i,\mu_V,\Sigma_{VV}) &= \frac{1}{\sqrt{(2\pi)^d\Sigma_{VV}}}\exp(-\frac{1}{2}(x-\mu_V)^\top\Sigma_{VV}^{-1}(x-\mu_V)) \\ P(y|x,y_A) &= \mathcal{N}(y;\mu_{y|A},\sigma_{y|A}^2) \\ \mu_{y|A} &= \sum_{x,A}\Sigma_{AA}^{-1}y_A \\ \Sigma_{VV} &= \beta^2 XX^\top + \sigma^2 I \\ \sigma_{y|A}^2 &= \Sigma_{xx} - \Sigma_{xA}\Sigma_{AA}^{-1}\Sigma_{Ax}\end{aligned}$$

Gaussian Process (Kernelized BLR)

Replace $\Sigma_{VV} = K + \sigma^2 I_n$.

Active Learning

D-optimality: $x_t = \operatorname{argmax}_{x\in X}\sigma_{t-1}^2(x)$ (pick the most uncertain sample)
A-optimality: $x_t = \operatorname{argmax}_{x\in X}\int[\sigma_t^2(x) - \sigma_{t-1}^2(x)]dx$ (pick the sample that'll reduce the variance the most)

Ensemble Methods

Use combination of simple hypotheses (weak learners) to create one strong learner.

$$f(x) = \sum_{i=1}^n \beta_i h_i(x)$$

Bagging: train weak learners on random subsamples with equal weights.
Boosting: train on all data, but reweigh misclassified samples higher.

Decision Trees

Stumps: partition linearly along 1 axis

$$h(x) = sign(ax_i - t)$$

Decision Tree: recursive tree of stumps, leaves have labels. To train, either label if leaf's data is pure enough, or split data based on score.

Ada Boost

Effectively minimize exponential loss.

$$f^*(x) = \underset{f\in F}{\operatorname{argmin}}\sum_{i=1}^n \exp(-y_i f(x_i))$$

Train m weak learners, greedily selecting each one

$$(\beta_i,h_i) = \underset{\beta,h}{\operatorname{argmin}}\sum_{i=1}^n \exp(-y_i(f_{i-1}(x_j)+\beta h(x_j)))$$

Generative Methods

Discriminative - estimate $P(y|x)$ - conditional.
Generative - estimate $P(y,x)$ - joint, model data generation.

Naive Bayes

All features independent.

$$P(y|x) = \frac{1}{Z}P(y)P(x|y), Z = \sum_y P(y)P(x|y)$$

$$y = \underset{y'}{\operatorname{argmax}} P(y'|x) = \underset{y'}{\operatorname{argmax}} \hat{P}(y')\prod_{i=1}^d \hat{P}(x_i|y')$$

Discriminant Function

$$f(x) = \log(\frac{P(y=1|x)}{P(y==1|x)}), y = sign(f(x))$$

Fischer's Linear Discriminant Analysis (LDA)

$$c = 2, p = 0.5, \hat{\Sigma}_- = \hat{\Sigma}_+ = \hat{\Sigma}$$

$$\begin{aligned}y &= sign(w^\top x + w_0) \\ w &= \hat{\Sigma}^{-1}(\hat{\mu}_+ - \hat{\mu}_-) \\ w_0 &= \frac{1}{2}(\hat{\mu}_-^\top\Sigma^{-1}\hat{\mu}_- - \hat{\mu}_+^\top\Sigma^{-1}\hat{\mu}_+)\end{aligned}$$

Unsupervised Learning

K-means

(clustering = classification)

$$L(\mu) = \sum_{i=1}^n \min_{j\in\{1\ldots k\}}\|x_i - \mu_y\|_2^2$$

Lloyd's Heuristic: (1) assign each x_i to closest cluster
(2) recalculate means of clusters.

Gaussian Mixture Modeling

Same as Bayes, but class label z unobserved.

$$(\mu^*, \Sigma^*, w^*) = \text{argmin} - \sum_i log \sum_{j=1}^{k} w_j \mathcal{N}(x_i | \mu_i, \Sigma_j)$$

EM Algorithm

E-step: expectation: pick clusters for points. Calculate $\gamma_j^{(t)}(x_i)$ for each i and j
M-Step: maximum likelihood: adjust clusters to best fit points.

$$\begin{aligned} \omega_j^{(t)} &\leftarrow \frac{1}{n} \sum_{i=1}^{n} \gamma_j^{(t)}(x_i) \\ \mu_j^{(t)} &\leftarrow \frac{\sum_{i=1}^{n} \gamma_j^{(t)}(x_i)(x_i)}{\sum_{i=1}^{n} \gamma^{(t)}(x_i)} \\ \Sigma_j^{(t)} &\leftarrow \frac{\sum_{i=1}^{n} \gamma_j^{(t)}(x_i)(x_i - \mu_j^{(t)})(x_i - \mu_j^{(t)})^\top}{\sum_{i=1}^{n} \gamma_j^{(t)}(x_i)} \end{aligned}$$

PCA

(dimensional reduction = regression)

$$\Sigma = \frac{1}{n} \sum_{i=1}^{n} x_i x_j^\top, \quad \mu = \frac{1}{n} \sum_{i=1}^{n} x_i = 0$$

$$(W, z_1, ..., z_n) = \text{argmin} \sum_{i=1}^{n} \|W z_i - x_i\|_2^2$$

W is orthogonal, $W = (v_1 | ... | v_k)$ and $z_i = w^\top x_i$

$$\Sigma = \sum_{i=1}^{d} \lambda_i v_i v_i^\top \quad \lambda_1 \geq ... \geq \lambda_d \geq 0$$

Kernel PCA

$$\alpha_i^* = \arg \max_{\alpha^\top K \alpha = 1} = \alpha^\top K^\top K \alpha$$

$$\alpha^{(i)} = \frac{1}{\sqrt{\lambda_i}} \frac{v_i}{\|v_i\|_2}, \quad K = \sum_{i=1}^{n} \lambda_i v_i v_i^\top$$

15 TODO

This is the chapter on on what still has to be done.