Disclaimer

This document is an exam summary and follows the given material of the lecture *Introduction to Machine Learning*. Its contribution is a short summary that contains the most important concepts, formulas and algorithms. Due to curriculum content updates, some content may not be relevant to future versions of the course.

I do not guarantee the accuracy or completeness, nor is this document endorsed by the instructors. Any errors that are pointed out to me are welcome. The complete LaTeX source code can be found at `https://github.com/tstreule/eth-cheat-sheets`.

Introduction to Machine Learning

Timo Streule, tstreule@ethz.ch - *18.01.2022*

1 Basics

Fundamental Assumption

Data is iid for unknown P: $(x_i, y_i) \sim P(X,Y)$
Empirical risk: $\hat{R}_D(w) = \frac{1}{|D|}\sum_{(x,y)\in D}(y - w^\top x)^2$
True risk: $R(w) = \int p(x,y)\, r_i^2\, \partial x \partial y = \mathbb{E}_{x,y}[r_i^2]$

Standardization: $(\text{for } x_k \in X, k = 1, \dots, d)$

Centered data with unit variance:
$\tilde{x}_{i,k} = \frac{x_{i,k} - \hat{\mu}_k}{\hat{\sigma}_k}$
$\hat{\mu}_k = \frac{1}{n}\sum_{i=1}^n x_{i,k}, \quad \hat{\sigma}_k^2 = \frac{1}{n}\sum_{i=1}^n (x_{i,k} - \hat{\mu}_k)^2$

Parametric vs. Nonparametric:

Parametric: have finite set of parameters. e.g. linear regression, linear perceptron
Nonparametric: grow in complexity with the size of the data, more expressive. e.g. k-NN

Gradient Descent:

1. Pick arbitrary $w_0 \in \mathbb{R}^d$
2. $w_{t+1} = w_t - \eta_t \nabla_w \hat{R}(w_t)$

$\nabla_w = \left[\frac{\partial}{\partial w_1} \cdots \frac{\partial}{\partial w_d}\right]$

Stochastic Gradient Descent (SGD):

1. Pick arbitrary $w_0 \in \mathbb{R}^d$
2. $w_{t+1} = w_t - \eta_t \nabla_w \ell(w_t; x', y')$, with u.a.r. (random) data point $(x', y') \in D$

works if $\sum_t \eta_t = \infty$ and $\sum_t \eta_t^2 < \infty$, e.g. $\eta_t = \frac{1}{t}$

2 Regression

Solve $w^* = \arg\min_w \hat{R}(w) + \lambda C(w)$, $\quad y = w^\top x$

residual: $r_i = y_i - w^\top x_i$, **cost**: $\hat{R}(w)$

Linear Regression

$\hat{R}(w) = \sum_{i=1}^n r_i^2 = \|Xw - y\|_2^2$
$\nabla_w \hat{R}(w) = -2\sum_{i=1}^n r_i \cdot x_i$
closed form: $w^* = (X^\top X)^{-1} X^\top y$

Ridge regression

$\hat{R}(w) = \sum_{i=1}^n r_i^2 + \lambda\|w\|_2^2$
$\nabla_w \hat{R}(w) = -2\sum_{i=1}^n r_i \cdot x_i + 2\lambda w$
closed form: $w^* = (X^\top X + \lambda I)^{-1} X^\top y$

L1-regularized regression (Lasso)

$\hat{R}(w) = \sum_{i=1}^n r_i^2 + \lambda\|w\|_1$

3 Classification

Solve $w^* = \arg\min_w \hat{R}(w)$, $\quad y = \text{sign}(w^\top x)$
$\hat{R}(w) = \frac{1}{n}\sum_{i=1}^n \ell(w; x_i, y_i), \quad \nabla_w \hat{R} = \frac{1}{n}\sum_n \nabla_w \ell$

0/1 loss: $\to$ intractable

$\ell_{0/1}(w; x_i, y_i) = [y_i \neq \text{sign}(w^T x_i)] \in \{0,1\}$

Perceptron algorithm: $\to$ use ℓ_P and SGD

$\ell_P(w; x_i, y_i) = \max(0, -y_i w^\top x_i)$

$$\nabla_w \ell_P(w; x_i, y_i) = \begin{cases} 0 & \text{if } y_i w^\top x_i \geq 0 \\ -y_i x_i & \text{otw. (incorrect)} \end{cases}$$

Data lin. separable $\Rightarrow$ obtains a lin. separator

Support Vector Machine (SVM): $\to$ Hinge

$\ell_H(w; x_i, y_i) = \max(0, 1 - y_i w^T x_i)$
$\hat{R}(w) = \frac{1}{n}\sum_n \ell_H + \lambda\|w\|_2^2, \quad \nabla_w \hat{R} = \ldots + 2\lambda w$

$$\nabla_w \ell_H(w; x_i, y_i) = \begin{cases} 0 & \text{if } y_i w^\top x_i \geq 1 \\ -y_i x_i & \text{otw.} \end{cases}$$

$w_{t+1} \leftarrow w_t(1 - 2\eta_t\lambda) + y_i x_i \eta_t [y_i w^\top x_i < 1]$
For **L1-SVM** (feature selection) use $\|w\|_1$

4 Kernels $\hat{=}$ scalar prod. in feature space ϕ

K. trick: $\boldsymbol{x}_i^\top \boldsymbol{x}_j \overset{\text{Mercer}}{\rightsquigarrow}$ $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)^\top \phi(\boldsymbol{x}_j)$

Properties of kernel

$k: \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is an inn. prod. (symm., pos.-def.).
Need: $K \succeq 0 \; \forall \boldsymbol{x}_i$, where $K_{i,j} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$
Hence: • check pos. eigenvalues *or better*
• $vKv^\top = \sum_i \sum_j \alpha_i \alpha_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 0$
$\to$ check for $i = j = 1$ for *counter example*

Important kernels

Constant:	$k(x,y) = c$ with $c \geq 0$
Linear:	$k(x,y) = x^\top y$
Polynomial:	$k(x,y) = (x^\top y + 1)^d$
Gaussian:	$k(x,y) = \exp(-\|x-y\|_2^2 / h^2)$
Laplacian:	$k(x,y) = \exp(-\|x-y\|_1 / h)$

Composition rules

$\circ\; k = k_1 + k_2 \quad \circ\; k = c \cdot k,\; c > 0 \quad \circ\; k = k_1 \cdot k_2$
$\circ\; k = f(k_1)$, f: exp. *or* polyn. with all pos. coeff.

Kernelized Perceptron / SVM

Ansatz: $w^* \in \text{span}(X) \Rightarrow w = \sum_{j=1}^n \alpha_j y_j x_j$
$\alpha^* = \arg\min_\alpha \frac{1}{n}\sum_n \max\{0, 1 - y_i \alpha^\top k_i\} + \lambda\alpha^\top D_y K D_y \alpha$
with $k_i = [.., y_j k(x_i, x_j), ..]$ and $D_y = \text{diag}(y_i)$
Predict: $\hat{y} = \text{sign}(\sum_{i=1}^n \alpha_i y_i k(x_i, x))$

Kernelized linear regression (KLR)

Ansatz: $w = \sum_{j=1}^n \alpha_j x_j = \sum_{j=1}^n \alpha_j \phi(x_j)$
$\alpha^* = \arg\min_\alpha \frac{1}{n}\|\alpha^\top K - y\|_2^2 + \lambda\alpha^\top K\alpha$
closed form: $\alpha^* = (K + \lambda I)^{-1} y$
Predict: $\hat{y} = \sum_{i=1}^n \alpha_i k(x_i, x)$

Kernelized LogReg

$\hat{\alpha} = \arg\min_\alpha \sum_{i=1}^n \log(1 + \exp(-y_i \alpha^\top K_i)) + \lambda\alpha^\top K\alpha$
$P(y|x, \hat{\alpha}) = \left(1 + \exp(-y\sum_{j=1}^n \alpha_j k(x_j, x))\right)^{-1}$

Semi-parametric kernel

additive combination of linear and non-linear kernel fct's, e.g. $x \leftrightarrow \sin(x \cdot \gamma)$ "periodic" kernel

5 Imbalanced Data

Cost Sensitive Classification

Replace loss by: $\ell_{CS}(w; x, y) = c_y \ell(w; x, y)$
e.g. $\ell_\pm = c_\pm \ell(w; x, y) \to c_- \cdot \hat{R}(w; \frac{c_+}{c_-}, c_-)$

Metrics

acc $= \frac{\text{TP}+\text{TN}}{n}$, **prec** $= \frac{\text{TP}}{p_+}$
FPR $= \frac{\text{FP}}{n_-}$, **Recall/TPR** $= \frac{\text{TP}}{n_+}$

TP	FP	p_+
FN	TN	p_-
n_+	n_-	n

Fβ score: $F_\beta = \frac{(1+\beta)^2}{\frac{1}{\text{prec}} + \frac{\beta^2}{\text{rec}}}$, **ROC**: FPR vs. TPR

6 Multi-class

1-vs-all: c models, confidence $f^{(i)}(x) = w^{(i)\top}x$
1-vs-1: $c\frac{c-1}{2}$ models, voting scheme
Multi-class Hinge: $\ell(w^{(1:c)}; x, y) = \max\{0, 1 + \underbrace{\max_{i:j\neq y} w^{(j)\top}x}_{(**)} - \underbrace{w^{(y)\top}x}_{(***)}\}$
Confidence: $w^{(y)\top}x \geq \max_{j\neq y} w^{(i)\top}x + 1 \; (*)$

$$\nabla_{w^{(j)}}\ell = \begin{cases} 0 & (*) \text{ satisfied } or\; j \notin \{y, \hat{y}\} \\ -x & \neg(*) \text{ and } j = y \rightsquigarrow (**) \\ +x & \neg(*) \text{ and } j = \hat{y} \rightsquigarrow (***) \end{cases}$$

7 Neural networks $\phi_j(x_i) \leftrightarrow \phi(x_i, \theta)$

Parameterize feature map: $\phi(x, \theta)$ instead of $\phi(x)$, usually: $\phi(x,\theta) = \varphi(\theta^\top x)$ $= \varphi(z)$
$\Rightarrow w^* = \arg\min_{w,\theta} \sum_{i=1}^n \ell(y_i; \underbrace{\sum_{j=1}^m w_j \phi(x_i, \theta_j)}_{f(x; w, \theta)})$

Activation functions $\varphi(z)$

Sigmoid: $\frac{1}{1+\exp(-z)} \in [0,1]$, $\varphi'(z) = (1 - \varphi(z)) \cdot \varphi(z)$
Tanh: $\varphi(z) = \tanh(z) = \frac{\exp(z) - \exp(-z)}{\exp(z) + \exp(-z)} \in [-1, 1]$
ReLu: $\varphi(z) = \max(z, 0)$ $\to$ not smooth

Predict: forward propagation

1. $v^{(0)} = x$
2. $v^{(l)} = \varphi(z^{(l)})$, $z^{(l)} = W^{(l)} v^{(l-1)}$, for $l = 1 : L-1$
3. $f = z^{(L)} = W^{(l)} v^{(L-1)}$
4. **Pred.**: $\hat{y} = f$ (Regr.) *or* $\begin{cases} \hat{y} = \text{sign}(f) \; or \\ \hat{y} = \arg\max_i (f_i) \end{cases}$

Compute gradient: backpropagation

Output: $[\cdots \delta_k^{(L)} \cdots] = \delta^{(L)} = \ell'(f) = [\cdots \ell_k'(f_k) \cdots]$
Hidden layer: for $l = L-1 : 1$,
$\delta_k^{(l)} = \varphi'(z_k) \cdot \sum_{j \in \text{layer}_{(l+1)}} w_{jk}\delta_j$ **Grad.**: $\frac{\partial \ell}{\partial w_{j,k}} = \delta_j^{(l)} v_k^{(l-1)}$

$W = [w_{jk}^{(L)} \cdots w_{jk}^{(1)}]_{jk}, \quad L(W) \equiv L = \sum_j \ell_j(y_j, f_j)$

Learning with momentum

1. $a \leftarrow ma + \eta_t \nabla_W \ell(W; y, x)$ **2.** $W \leftarrow W - a$

Convolutional NNs

$\to \overbrace{\text{conv.} \to \text{pooling}}^{\text{repeat } n \text{ times}} \to \overbrace{\text{fully connected}}^{\text{Perceptron } (m \text{ times})} \to \text{out}$
Convolution: for edge regions $\to$ 0-padding
pooling (subsampling): e.g. 'max' pooling
output dim's: $\alpha = \frac{n + 2p - f}{s} + 1$
where m: # $f \times f$ filters, n: img. dim., p: padding, # of added zeros, s: strides (amount by which filter shifts)

8 Clustering — Unsupervised

k-Means

$\hat{R}(\mu) = \sum_{i=1}^n \min_{j \in \{1,\dots,k\}} \|x_i - \mu_j\|_2^2 = \sum_i d(x_i, \mu_j)$
$\hat{\mu} = \arg\min_\mu \hat{R}(\mu)$ $\leftarrow$ non-convex, NP-hard

Lloyd's heuristic: *Initialize* centers $\mu_{1:k}^{(0)}$, *assign* points to closest center $\to \arg\min$, *update centers* to mean of each cluster, repeat

k-Means++:

$\to$ for initialization $2 \dots j$
$\text{P(pick } x_l) = \frac{1}{z} d(x_i - \mu_{1:j-1})$, $\mu_j^{(0)} \leftarrow x_i$

Optimal *k*-value:

$\to$ "Elbow" trick *or* Regularization $L(\mu) = \min_k \min_{\mu_k} \hat{R}(\mu_{1:k}) + \lambda k$

9 Dimension reduction

PCA

$\to f: \mathbb{R}^d \to \mathbb{R}^k, k < d \qquad (\lambda_1 \geq \ldots \geq \lambda_d \geq 0)$
centered: $\mu = \mathbb{E}(X) = \frac{1}{n}\sum_{i=1}^n x_i = 0$
empir. cov.: $\Sigma = \frac{1}{n}\sum_{i=1}^n x_i x_i^\top = \sum_{i=1}^d \lambda_i v_i v_i^\top$
$(\hat{W}, \hat{z}_1, .., \hat{z}_n) = \arg\min \sum_{i=1}^n \|W z_i - x_i\|_2^2$
Sol.: $\hat{z}_i = \hat{W}^\top x_i$, $\hat{W} = (v_1|..|v_k) \in \mathbb{R}^{d\times k}$, orth.

Kernel PCA

$\to$ non-linear, feature discov.
Ansatz: see KLR, **Constraint**: $\|w\|_2 = \alpha^\top K\alpha = 1$
Kernel PC: $\alpha^{(1)}, .., \alpha^{(k)} \in \mathbb{R}^n$, $\alpha^{(i)} = \frac{1}{\sqrt{\lambda_i}} v_i$,
$K = \sum_{i=1}^n \lambda_i v_i v_i^\top, \lambda_1 \geq .. \geq \lambda_d \geq 0$
New point: $\hat{z}_i = \sum_{j=1}^n \alpha_j^{(i)} k(\hat{x}_i, x_j)$

Autoencoders: Find identity fct.: $x \approx f(x;\theta)$
$f(x;\theta) = f_{\text{decode}}(f_{\text{encode}}(x;\theta_{\text{enc.}});\theta_{\text{dec.}})$

10 Probability modeling

Find $h(x)$ that min. pred. error:
$R(h) = \mathbb{E}_{x,y}[\ell(y;h(x))] = \int P(x,y)\,\ell(y;h)\,dx\,dy$

Bayes optimal predictor $\to \hat{y} = h^*(x)$
$\frac{d\ell(\hat{y})}{d\hat{y}} = 0 \to \hat{y} = \mathbb{E}[Y|X=x] = \int y \cdot \hat{P}(y|X=x)\,dy$

MLE $-\log P(y|x,w)$
$\theta^* = \underset{\theta}{\text{argmax}}\,\hat{P}(y_{1:n}|x_{1:n},\theta) \overset{\text{iid}}{=}$
$\underset{\theta}{\text{argmin}} - \sum_{i=1}^n \ln P(y_i|x_i,\theta)$

e.g. lin. Gauss: $y_i = w^\top x_i + \varepsilon_i,\quad \varepsilon_i \sim \mathcal{N}(0,\sigma^2)$
i.e. $y_i \sim \mathcal{N}(w^\top x_i, \sigma^2) \xrightarrow{\text{MLE and log}}$ LS regression

Bias/Variance/Noise
Prediction error = Bias^2 + Variance + Noise
- Noise: risk incurred by the optimal model, $P(y|x)$ loss/likelihood fct.
- Variance: est. model from limited data
- Bias: incurred by regularizer

higher bias implies much lower variance

MAP $-\log P(w)$
Prior: bias on param's, e.g. $w_i \sim \mathcal{N}(0,\beta^2)$
$\xrightarrow{\text{Bay.}} P(w|x_{1:n},y_{1:n}) = \frac{P(w|x)P(y|x,w)}{P(y|x)} = \frac{P(w)P(y|x,w)}{P(y|x)}$

Logistic regression (Classification)
Link fct.: $\sigma(w^\top x) = \frac{1}{1+\exp(-w^\top x)}$ **(Sigmoid)**
$P(y|x,w) = Ber(y;\sigma(w^\top x)) = \sigma(yw^\top x)$
MLE: $\hat{w} = \underset{w}{\text{argmin}} \sum_i \log(1+\exp(-y_i w^\top x_i))$
with $\hat{R}(w) = \sum_{i=1}^n \ell_{\text{logistic}}(w;x_i,y_i)$
Grad.: $\nabla_w \ell(w) = P(Y \neq y|x)\,(-yx) \quad \to (Y = -y)$
MAP: Gauss. prior $\to \|w\|_2^2$, Lap. $\to \|w\|_1^1$

11 Bayesian decision theory

giv.: $P(y|x)$, set of actions $\mathcal{A}$, cost $C: \mathcal{Y} \times \mathcal{A} \to \mathbb{R}$
opt. action: $a^* = \underset{a \in \mathcal{A}}{\text{argmin}}\,\mathbb{E}_y[C(y,a)|x]$
$= \underset{a}{\text{argmin}} \sum_n P(y|x)C(y,a)$

Classification: $C(y,a) = [y \neq a]$; asymmetric:
$$C(y,a) = \begin{cases} c_{FP} & \text{if } y=-1, a=+1 \\ c_{FN} & \text{if } y=+1, a=-1 \\ 0 & \text{otw.} \end{cases}$$
E.g. $y \in \{-1,+1\}$, predict '+' if $c_+ < c_-$,
$c_+ = \mathbb{E}_y(C(y,+1)|x) = (1-p)c_{FP}$
$= P(y=1|x)\cdot 0 + P(y=-1|x)\cdot c_{FP}$
Regression: $C(y,a) = (y-a)^2$; asymmetric:
$C(y,a) = c_1 \max(y-a,0) + c_2 \max(a-y,0)$

12 Discriminative / generative modeling

Discr.: estimate $P(y|x)$, **Generative:** $P(y,x)$

Chain rule: $P(x,y) = P(y)P(x|y)$

Deriving Decision Rules

1. Estimate prior on labels $P(y)$
2. Est. cond. distr. for *each class* y: $P(x|y)$
   $\rightsquigarrow Z = P(x) = \sum_{y'} P(x,y')$
3. Predict using Bayes: $P(y|x) = \frac{1}{Z}P(x,y)$,
4. Minimize misclassification error:
   $\hat{y} = \text{argmax}\,P(y|x) = \underset{y}{\text{argmax}}\,P(y)P(x|y)$
   $= \ldots \prod_{i=1}^d P(x_i,y)$

Binary: $y = \text{sign}(f(x))$, $f(x) = \log\frac{P(Y=+1|x)}{P(Y=-1|x)}$
$y_{\text{pred}} = [P(X,1) \geq P(X,0)] = [p_1 P(X|)1 \geq p_0 P(X|0)]$

Examples

MLE for Class.: $P(y) = p_y = \frac{\text{Count}(Y=y)}{n} = \frac{n_y}{n}$

MLE for P(x|y): $P(x_i|y) = \mathcal{N}(x_i;\mu_{y,i},\sigma^2_{y,i})$
$\hat{\mu}_{y,i} = \frac{1}{n_y}\sum_{j:y_j=y} x_{j,i}$ $\quad \hat{\sigma}^2_{y,i} = \frac{1}{n_y}\sum_{j:y_j=y}(x_{j,i}-\hat{\mu}_{y,i})^2$
$x_{j,i}$: value of feature i for instance j (x_j,y_j)

MLE for Poi.: $\lambda = \text{avg}(x_i)$
$\mathbb{R}^d$: $P(X=x|Y=y) = \prod_{i=1}^d Pois(\lambda_y^{(i)}, x^{(i)})$

Gaussian Bayes Classifier

$\hat{P}(x|y) = \mathcal{N}(x;\hat{\mu}_y,\hat{\Sigma}_y)$

MLE: $\hat{\mu}_y = \frac{1}{n_y}\sum_{j:y_j=y} x_j \in \mathbb{R}^d$
$\hat{\Sigma}_y = \frac{1}{n_y}\sum_{j:y_j=y}(x_j-\hat{\mu}_y)(x_j-\hat{\mu}_y)^T \in \mathbb{R}^{d\times d}$

c = 2: $f(x) = \log\frac{p}{1-p} + \frac{1}{2}\Big[\log\frac{|\hat{\Sigma}_-|}{|\hat{\Sigma}_+|} + \ldots$
$((x-\hat{\mu}_-)^\top\hat{\Sigma}_-^{-1}(x-\hat{\mu}_-)) - ((x-\hat{\mu}_+)^\top\hat{\Sigma}_+^{-1}(x-\hat{\mu}_+))\Big]$

c = 2 – Fisher's LDA:
Assume: $p = 0.5$; $\hat{\Sigma}_- = \hat{\Sigma}_+ \equiv \hat{\Sigma}$
$\implies f(x) = w^\top x + w_0$
where $w = \hat{\Sigma}^{-1}(\hat{\mu}_+ - \hat{\mu}_-)$ and
$w_0 = \frac{1}{2}(\hat{\mu}_-^\top\hat{\Sigma}^{-1}\hat{\mu}_- - \hat{\mu}_+^\top\hat{\Sigma}^{-1}\hat{\mu}_+)$

Outlier Detection
$P(x) \leq \tau$
Categorical (Naive) Bayes Classifier
MLE for feature distr.: $\hat{P}(X_i = c|Y=y) = \theta_{c|y}^{(i)}$
$\theta_{c|y}^{(i)} = \frac{\text{Count}(X_i=c,Y=y)}{\text{Count}(Y=y)}$, $\hat{p}_y = \frac{\text{Count}(Y=y)}{n}$

13 Missing data, w/o labels →

Mixture modeling
Model each class as prob. distribution $P(x|\theta_j)$
$P(D|\theta) = \prod_{i=1}^n \sum_{j=1}^c w_j P(x_i|\theta_j)$
$L(w,\theta) = -\sum_{i=1}^n \log\sum_{j=1}^c w_j P(x_i|\theta_j)$
$\implies \theta^* = \text{argmin}\,L(\theta) \to$ **non convex**
Gaussian-Mixture Bayes classifiers
Estimate prior $P(y)$; Est. cond. distr. for each class: $P(x|y) = \sum_{j=1}^{k_y} w_j^{(y)}\mathcal{N}(x;\mu_j^{(y)},\Sigma_j^{(y)})$
Hard-EM algorithm

Initialize $\theta^{(0)}$; Let $Q^{(t)}(z) = P(z|x,\theta^{(t)})$
For $t = 1,2,\ldots$ **do:**
- **E-step**: estimate log-likelihood
  Predict most likely class for each point x_i
  - $z_i^{(t)} = \underset{z}{\text{argmax}}\,P(z|x_i,\theta^{(t-1)})$
    $= \underset{z}{\text{argmax}}\,P(z|\theta^{(t-1)})\,P(x_i|z,\theta^{(t-1)})$
- **M-step**: Maximize (MLE)
  - $L^{(t)}(\theta) = \mathbb{E}_{Q^{(t)}}[\log P(x,y|\theta^{(t)})]$
  - $\theta^{(t)} = \underset{\theta}{\text{argmax}}\,L^{(t)} \to \frac{\partial}{\partial\theta}L = 0 \to \theta^{(t)} = \ldots$

$\theta^{(t)} = \underset{\theta}{\text{argmax}}\,P(D^{(t)}|\theta)$, i.e. $\mu_j^{(t)} = \frac{1}{n_j}\sum_{i:z_i=j} x_j$

Soft-EM algorithm
E-step: Calc p for each point and cls.: $\gamma_j^{(t)}(x_i)$
M-step: Fit clusters to weighted data points:
$w_j^{(t)} = \frac{1}{n}\sum_{i=1}^n \gamma_j^{(t)}(x_i)$; $\mu_j^{(t)} = \frac{\sum_{i=1}^n \gamma_j^{(t)}(x_i)x_i}{\sum_{i=1}^n \gamma_j^{(t)}(x_i)}$
$\sigma_j^{(t)} = \frac{\sum_{i=1}^n \gamma_j^{(t)}(x_i)(x_i-\mu_j^{(t)})^T(x_i-\mu_j^{(t)})}{\sum_{i=1}^n \gamma_j^{(t)}(x_i)}$
Soft-EM for semi-supervised learning
labeled y_i: $\gamma_j^{(t)}(x_i) = [j = y_i]$, unlabeled:
$\gamma_j^{(t)}(x_i) = P(Z=j|x_i,\mu^{(t-1)},\Sigma^{(t-1)},w^{(t-1)})$

14 Useful math

Probabilities
$$\mathbb{E}_x[X] = \begin{cases} \int x\cdot p(x)\,dx & \text{if continuous} \\ \sum_x x\cdot p(x) & \text{discrete} \end{cases}$$
$\text{Var}[X] = \mathbb{E}[(X-\mu_X)^2] = \mathbb{E}[X^2] - \mathbb{E}[X]^2$
Bayes Rule: (using **chain** rule)
$P(A|B) = \frac{P(B|A)\cdot P(A)}{P(B)}$, $P(Z|X,\theta) = \frac{P(X,Z|\theta)}{P(X|\theta)}$
p-Norm: $\|x\|_p = (\sum_{i=1}^n |x_i|^p)^{\frac{1}{p}}$, $1 \leq p < \infty$
Some gradients
$\nabla_x\|x\|_2^2 = 2x$
$f(x) = x^\top Ax$; $\nabla_x f(x) = (A + A^\top)x$
E.g. $\nabla_w \log(1+\exp(-yw^\top x)) = \ldots$
$= \frac{1}{1+\exp(-yw^\top x)}\cdot\exp(-yw^\top x)\cdot(-yx)$
$= \frac{1}{1+\exp(yw^\top x)}\cdot(-yx)$
Convex / Jensen's inequality
$f(x)$ convex $\Leftrightarrow f''(x) > 0 \Leftrightarrow x_i \in \mathbb{R}, \lambda \in [0,1]$:
$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda)f(x_2)$
Jensen's inequality: $g(\mathbb{E}[X]) \leq \mathbb{E}[g(X)]$
Gaussian / Normal distribution
$\mathcal{N}(\mu,\sigma^2) \sim \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$
Multivariate Gaussian:
$f(x) = \frac{1}{2\pi\sqrt{|\Sigma|}}\exp\left(\frac{1}{2}(x-\mu)^\top\Sigma^{-1}(x-\mu)\right)$
with $\Sigma = \begin{bmatrix}\sigma_{11}^2 & \sigma_{12} \\ \sigma_{21} & \sigma_{22}^2\end{bmatrix}$ and $\mu = \begin{bmatrix}\mu_1 \\ \mu_2\end{bmatrix}$
Multivariate Gaussian
Σ = covariance matrix, μ = mean
$f(x) = \frac{1}{2\pi\sqrt{|\Sigma|}}e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}$
Empirical: $\Sigma = \frac{1}{n}\sum_{i=1}^n x_i x_i^T$ (needs centered data points)
Positive semi-definite matrices
$M \in \mathbb{R}^{n\times n}$ is psd $\Leftrightarrow$
$\forall x \in \mathbb{R}^n : x^T Mx \geq 0 \Leftrightarrow$
all eigenvalues of M are positive: $\lambda_i \geq 0$