IML S23 Nils Benz

Linear Regression

$f(x) = w_0 + w_1 x = w_0 + \sum_{i=1}^n w_i x_{[i]} = w_0 + w^T x$

$L(f) = \frac{1}{n} \sum_{i=1}^n \ell(f(x_i), y_i)$ (→ = MSE w/ ℓ_{square})

Loss functions: • $\ell_{square} = (f(x)-y)^2$

• $\ell_{huber} = \begin{cases} \frac{1}{2}(y-f(x))^2 & \text{for } |y-f(x)| \le \delta \\ \delta(|y-f(x)| - \frac{1}{2}\delta) & \text{else} \end{cases}$

Closed Form Solution: $\nabla_w L(\hat{w}) \stackrel{!}{=} 0$

Normal Equations: $(X^TX)^{-1}X^Ty$ where X is $n \times (d+1)$, $X = \begin{bmatrix} 1 & — x_1 — \\ 1 & — x_2 — \\ \vdots & \\ 1 & — x_n — \end{bmatrix}$

Gradient Descent: $w^{t+1} = w^t - \eta \nabla_w L(w^t)$

• e.g. stop when $|L(w^{t+1}) - L(w^t)| \le \varepsilon$

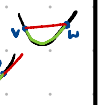
• momentum: $w^{t+1} - w^t = \alpha(w^t - w^{t-1}) - \eta \nabla_w L(w^t)$

Convexity, order conditions:

0 $L(\lambda w + (1-\lambda)v) \le \lambda L(w) + (1-\lambda) L(v)$

1 $L(v) \ge L(w) + \nabla L(w)^T (v-w)$

2 Hessian $\nabla^2 L(w) \succcurlyeq 0$



Convexity preserving operations

• $\alpha f + \beta g$ for $\alpha, \beta \ge 0$ and f, g convex

• $f \circ g = f(g(x))$ if a) f convex, g affine b) f non-decreasing, g convex

• $\max\{f(x), g(x)\}$ if f, g convex

Bias: dist. of avg. model to ground truth

$\mathbb{E}_x[(f^*(x) - \frac{1}{J}\sum_{j=1}^J \hat{f}_j(x))^2]$

Variance: avg. dist. of models to ground truth

$\mathbb{E}_x[\frac{1}{J}\sum_{j=1}^J (\hat{f}_j(x) - \bar{f}(x))^2]$

Regularization: (↙ focus on low degrees)

• Lasso: $\arg\min_w \|y - \Phi w\|_2^2 + \lambda \|w\|_1$

• Ridge: $\arg\min_w \|y - \Phi w\|_2^2 + \lambda \|w\|_2$ (↗ activate all degrees)

Matrix is psd: • All EVs ≥ 0

• All principal minors are non-neg.

↳ remove rows/cols w/ same index

• For pd only leading minors > 0

$\log(u \cdot v) = \log(u) + \log(v)$ / $a^b = e^{b \log(a)}$

$\log(u^v) = v \cdot \log(u)$ / $\log_b(u) = \frac{\log u}{\log b}$

Classification

Linear Classifier: $f(x) = w^Tx$ for some $w \in \mathbb{R}^d$

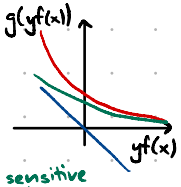
Non-linear: $f(x) = w^T\phi(x)$, $\phi \in \mathbb{R}^d \to \mathbb{R}^p$

Loss functions:

$g_{exp}(y\hat{f}(x)) = e^{-y\hat{f}(x)}$

$g_{log}(y\hat{f}(x)) = \log(1 + e^{-y\hat{f}(x)})$

$g_{lin}(y\hat{f}(x)) = -y\hat{f}(x)$



↑ less sensitive to outliers.

Cross-Entropy Loss ↙

$\ell_{ce}(f(x), y) = -\log\left(\frac{e^{f_y(x)}}{\sum_{k=1}^K e^{f_k(x)}}\right)$ ← $f_y = f_i$ where $y = i$

↑ $= (f_1(x), \dots, f_K(x))$

Hard-Margin SVM $w_{HM} = \arg\max_{\|w\|=1} \underbrace{\min_i y_i \langle w, x_i \rangle}_{=\text{margin}(w)}$

$w_{HM} \| \arg\min_{\hat{w}} \|\hat{w}\|_2$ s.t. $y_i \hat{w}^T x_i \ge 1 \quad \forall i = 1, \dots, n$

Null Hypothesis: the critical class, want no error

Type 1 Error: False Pos / Type 2 Error: FN

$FNR = \frac{\#FN}{\#P}$ / $FPR = \frac{\#FP}{\#N}$ / $TPR = \frac{\#TP}{\#P} = 1 - FNR$

Asymm. lin. classifier: $c_{FN} \cdot FNR + c_{FP} \cdot FPR$

ROC curve:

Precision: $\frac{\#TP}{\#[\hat{y} = +1]} \sim \mathbb{P}_n[y = 1 \mid \hat{y} = 1]$ } large

Recall: $\frac{\#TP}{\#[y = +1]} \sim \mathbb{P}_n[\hat{y} = 1 \mid y = 1]$

False Discovery Rate: $\frac{\#FP}{\#[\hat{y} = +1]} \sim \mathbb{P}_n[y = -1 \mid \hat{y} = 1]$ } small

FPR: $\frac{\#FP}{\#[y = -1]} \sim \mathbb{P}_n[\hat{y} = 1 \mid y = -1]$

F1 score $= \frac{2}{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}}$ → forces both to be large

Asymmetric loss: (False Positives ↙, False Neg. ↘)

$\ell(\hat{y}(x), y) = c_{FP} \mathbb{I}_{\hat{y}(x)=1, y=-1} + c_{FN} \mathbb{I}_{\hat{y}(x)=-1, y=1}$

$(AB\dots)^{-1} = \dots B^{-1}A^{-1}$

$(AB\dots)^T = \dots B^T A^T$

$\text{Tr}(A) = \sum_i \lambda_i$

$\text{Tr}(AB) = \text{Tr}(BA)$

$\text{Tr}(A+B) = \text{Tr}(A) + \text{Tr}(B)$

$a^Ta = \text{Tr}(aa^T)$

$\det(A) = \prod_i \lambda_i$

$\det(cA) = c^n \cdot \det(A)$

$\det(AB) = \det A \cdot \det B$

$\det(A^{-1}) = \frac{1}{\det(A)}$

$\det(I + uv^T) = 1 + u^Tv$

Let $A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$

$v_1 = \begin{bmatrix} A_{11} \\ \lambda_1 - A_{11} \end{bmatrix}$

$v_2 = \begin{bmatrix} A_{12} \\ \lambda_2 - A_{11} \end{bmatrix}$

$A^{-1} = \frac{1}{\det A} \begin{bmatrix} A_{22} & -A_{12} \\ -A_{21} & A_{11} \end{bmatrix}$

Let $f: \mathbb{R}^n \to \mathbb{R}^m$, $g: \mathbb{R}^m \to \mathbb{R}^p$

$\frac{d}{dx} g(f(x)) = J_g(f(x)) \cdot J_f(x) = \frac{\partial g}{\partial f} \cdot \frac{\partial f}{\partial x}$

Kernels

Solve $\hat{w} = \arg\min_{w \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^n \ell(y_i, w^T\phi(x_i))$

Output $\hat{f}(x) = \sum_j \hat{w}_j \phi_j(x) = \alpha \Phi \phi(x)$

1) Solve $\hat{\alpha} = \arg\min_{\alpha \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^n \ell(y_i, \alpha^T \Phi \phi(x_i))$

2) $\hat{f}(x)$ only depends on x via inner products

3) Replace $\langle \cdot, \cdot \rangle$ w/ $k(x,z)$ → $\hat{f}(x) = \sum_{i=1}^n \hat{\alpha}_i k(x, x_i)$

$\underset{(n \times p)}{\Phi} = \begin{pmatrix} -\phi(x_1)- \\ \vdots \\ -\phi(x_n)- \end{pmatrix}$ / $\Phi \cdot \phi(x_i) = \begin{pmatrix} \langle \phi(x_1), \phi(x_i) \rangle \\ \vdots \\ \langle \phi(x_n), \phi(x_i) \rangle \end{pmatrix}$

Kernel function $k: X \times X \to \mathbb{R}$

Kernel matrix $K = \begin{pmatrix} k(x_1,x_1) & \cdots & k(x_1,x_n) \\ \vdots & & \vdots \\ k(x_n,x_1) & \cdots & k(x_n,x_n) \end{pmatrix}$

Kernelized regression: $\hat{\alpha} = \arg\min_{\alpha \in \mathbb{R}^n} \|y - K\alpha\|^2$

w/ $\hat{f}(x) = \hat{\alpha}^T k_x \Rightarrow \hat{\alpha} = K^{-1}y$

Valid kernels: • symmetric $k(x,z) = k(z,x)$

• k is pos. semidefinite (all $\lambda \ge 0$)

⇒ e.g. $h(\langle x, z \rangle)$, $h(\|x - z\|)$

Kernel composition: if k_1, k_2 valid kernels:

• $k\left(\begin{pmatrix} x \\ y \end{pmatrix}, \begin{pmatrix} u \\ v \end{pmatrix}\right) = k_1(x,u)\,[+,\cdot]\,k_2(y,v)$

• $k(x,y) = k_1(x,y)\,[+,\cdot]\,k_2(x,y)$

Dimension Reduction

(compressed)

PCA: $(w^*, z^*) = \arg\min_{\|w\|_2=1, z} \sum_{i=1}^n \|z_i w - x_i\|_2^2$

$\Rightarrow w^* = \arg\min_{\|w\|_2=1} \sum_{i=1}^n \|ww^Tx_i - x_i\|_2^2$ ($z_i^* = w^Tx_i$)

$\overset{\mu=0}{=} \arg\max_{\|w\|_2=1} w^T \Sigma w$, $\Sigma = \frac{1}{n}\sum_{i=1}^n x_i x_i^T$

⇒ solution is principal EV of Σ

If $\Sigma = \sum_{i=1}^d \lambda_i v_i v_i^T$, $W = (v_1 | \cdots | v_k)$, $z_i = W^Tx_i$

↳ if $X \in \mathbb{R}^{n \times d} = USV^T$, first k cols of V are sol

Kernel PCA:

$\hat{\alpha} = \arg\max_{\alpha^TK\alpha = 1} \alpha^TK^TK\alpha \equiv \arg\max_\alpha \frac{\alpha^TK^TK\alpha}{\alpha^TK\alpha}$

Let $k = \sum_{i=1}^n \lambda_i v_i v_i^T$. Then $\alpha^{(i)} = \frac{1}{\sqrt{\lambda_i}} v_i$

and $z_i = \sum_{j=1}^n \alpha_j^{(i)} k(x_j, x)$

Autoencoders $\min_w \sum_{i=1}^n \|x_i - f(x_i; W)\|_2^2$

	0	$\frac{\pi}{6}$	$\frac{\pi}{4}$	$\frac{\pi}{3}$	$\frac{\pi}{2}$	$\frac{2\pi}{3}$	$\frac{3\pi}{4}$	$\frac{5\pi}{6}$	π
sin	0	$\frac{1}{2}$	$\frac{\sqrt{2}}{2}$	$\frac{\sqrt{3}}{2}$	1	$\frac{\sqrt{3}}{2}$	$\frac{\sqrt{2}}{2}$	$\frac{1}{2}$	0
cos	1	$\frac{\sqrt{3}}{2}$	$\frac{\sqrt{2}}{2}$	$\frac{1}{2}$	0	$-\frac{1}{2}$	$-\frac{\sqrt{2}}{2}$	$-\frac{\sqrt{3}}{2}$	-1
tan	0	$\frac{\sqrt{3}}{3}$	1	$\sqrt{3}$	-	$-\sqrt{3}$	-1	$-\frac{\sqrt{3}}{3}$	0

Neural Networks

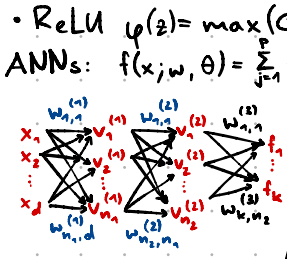
Activation Functions • id $\varphi(z) = z$

• Sigmoid $\varphi(z) = \frac{1}{1+\exp(-z)}$, $\varphi'(z) =$

• Tanh $\varphi(z) = \frac{\exp(z) - \exp(-z)}{\exp(z) + \exp(-z)}$, $\varphi'(z) =$

• ReLU $\varphi(z) = \max(0, z)$

ANNs: $f(x; w, \theta) = \sum_{j=1}^p w_j \varphi(\theta_j^T x)$ ($\theta_j^T x = z_j$, $\varphi(z_j) = v_j$)



$W^{(i)} = \begin{pmatrix} w_{1,1}^{(i)} & \cdots & w_{1,m_{i-1}}^{(i)} \\ \vdots & \ddots & \\ w_{m_i,1}^{(i)} & \cdots & w_{m_i,m_{i-1}}^{(i)} \end{pmatrix}$

Forward propagation • in layer: $v^{(0)} = [x; 1]$

• hidden layer $z^{(\ell)} = W^{(\ell)} \cdot v^{(\ell-1)}$

$v^{(\ell)} = [\varphi(z^{(\ell)}); 1]$

• output layer $f = W^{(L)} v^{(L-1)}$

• predict: $y = f$ regression, $y = \arg\max_j f_j$ class

$W^* = \arg\min_W \sum_{i=1}^n \ell(W; x_i, y_i)$ → Cross Entropy

SGD for ANNs: $W \leftarrow W - \eta_t \nabla_W \ell(W; x, y)$

Back propagation: 1) $\delta^{(L)} = \nabla_f \ell$

2) $\delta^{(\ell)} = \varphi'(z^{(\ell)}) \odot (W^{(\ell+1)T} \cdot \delta^{(\ell+1)})$ (⊙ = c-wise mult.)

$\nabla_{W^{(\ell)}} \ell = \delta^{(\ell)} v^{(\ell-1)T}$

Batch Normalization (compute $\bar{v}$)

1) MB-mean: $\mu_S = \frac{1}{|S|} \sum_{i \in S} v_i$

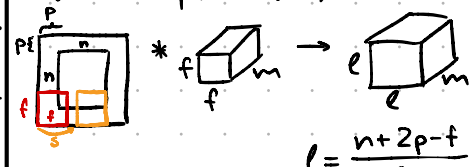
2) MB-variance: $\sigma_S^2 = \frac{1}{|S|} \sum_{i \in S} (v_i - \mu_S)^2$

3) Normalize points: $\hat{v}_i = \frac{v_i - \mu_S}{\sqrt{\sigma_S^2 + \varepsilon}}$

4) Scale/shift: $\bar{v}_i = \gamma \hat{v}_i + \beta$ → learn γ, β

Conv. Neural Nets

$v^{(\ell+1)} = \varphi(W * v^{(\ell)})$



$\ell = \frac{n + 2p - f}{s} + 1$

Pooling: ⊞ → □, e.g. avg or max

Clustering

k-means algorithm $(O(n \cdot k \cdot d))$

1) init $\mu^{(0)} = [\mu_1^{(0)}, \dots, \mu_k^{(0)}]$

2) until converge: $z_i \leftarrow \arg\min_j \|x_i - \mu_j^{(t-1)}\|_2$

$\mu_j^{(t)} \leftarrow \frac{1}{n_j} \sum_{i: z_i = j} x_i$ (→ mean)

k-means++: $\mu_1^{(0)}$ = random x_i

add other centers proportional to squared dist. to closest selected

↳ cost $O(\log k)$ times optimal

Gaussian Mixture Models

Hard-EM: 1) Init parameters $\theta^{(0)}$

2) E-step: $z_i^{(t)} = \arg\max_z P(z \mid x_i, \theta^{(t-1)})$

$= \arg\max_z P(z \mid \theta^{(t-1)}) \cdot P(x_i \mid z, \theta^{(t-1)})$

3) $D^{(t)} = \{(x_1, z_1^{(t)}), \dots, (x_n, z_n^{(t)})\}$

4) M-step: Compute MLE as for GBC

$\theta^{(t)} = \arg\max_\theta P(D^{(t)} \mid \theta)$

Soft-EM: 1) Init $\mu^{(0)}, \Sigma^{(0)}, w^{(0)}$

2) E-step: calc clusters probabilities

$\gamma_j^{(t)}(x_i)$ given estimates of $(t-1)$

3) M-step: Fit clusters to data points

$w_j^{(t)} \leftarrow \frac{1}{n} \sum_{i=1}^n \gamma_j^{(t)}(x_i)$

$\mu_j^{(t)} \leftarrow \frac{\sum_{i=1}^n \gamma_j^{(t)}(x_i) \cdot x_i}{\sum_{i=1}^n \gamma_j^{(t)}(x_i)}$

$\Sigma_j^{(t)} \leftarrow \frac{\sum_{i=1}^n \gamma_j^{(t)}(x_i)(x_i - \mu_j^{(t)})(x_i - \mu_j^{(t)})^T}{\sum_{i=1}^n \gamma_j^{(t)}(x_i)}$

Constrained GMMs:

		# params
Spherical	$\Sigma_i = \sigma_i^2 \cdot I$	k
Diagonal	$\Sigma_i = \text{diag}(\sigma_{i,1}^2, \dots, \sigma_{i,d}^2)$	$k \cdot d$
Tied	$\Sigma_1 = \dots = \Sigma_k$ → All shapes the same	$\frac{d(d+1)}{2}$
Full	w, μ, Σ arbitrary	$k \cdot \frac{d(d+1)}{2}$

Initialization: • weights: uniform dist.

• means: random or k-means++

• variances: spherical

Semi-supervised: $\gamma_j^{(t)}(x_i) = [j = y_i]$ for points w/ label. Rest Soft-EM

Probabilistic Modelling

Gaussian MLE, $p(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ ← Samples

log-likelihood $LLH(\mu, \sigma) = \log p(D; \mu, \sigma)$

$\nabla_\mu LLH \stackrel{!}{=} 0 \Rightarrow \hat{\mu}_{MLE} = \frac{1}{n} \sum_{i=1}^n x_i$

$\nabla_\sigma LLH \stackrel{!}{=} 0 \Rightarrow \hat{\sigma}_{MLE} = \frac{1}{n} \sum_{i=1}^n (x_i - \mu_{MLE})^2$

$\Rightarrow \mathbb{P} = \mathcal{N}(\hat{\mu}_{MLE}, \hat{\sigma}_{MLE})$

Discriminative: estimate $p(y \mid x)$

Decision boundaries btw. classes

Generative: estimate $p(y, x) = p(x \mid y) \cdot p(y)$

Model prob. distribution of data

Gaussian Naive Bayes: Decision boundary linear, if Σ_y diagonal and same for all y

Finding MLE for $\mathbb{P}_{X|Y}$

$\hat{\mu}_y = \frac{1}{\#Y=y} \sum_{i: y_i = y} x_i$, $\hat{\sigma}_{y,k} = \frac{1}{\#Y=y} \sum_{i: y_i = y} (x_{i,k} - \mu_{y,k})^2$

Bayesian Modeling: We know something about the distribution of $\theta \rightarrow p(D) = \int p(D \mid \theta) \cdot p(\theta) \, d\theta$

$\Rightarrow$ output $\arg\max_\theta p(D, \theta) = p(D \mid \theta) \cdot p(\theta)$

Generative Modeling for classification

1) estimate prior on labels $p(y)$

2) estimate conditional dist. $p(x \mid y)$

3) using Bayes rule: $p(y \mid x) = \frac{1}{Z} p(y) \cdot p(x \mid y)$ with $Z = \sum_y p(y) \cdot p(x \mid y)$

MLE for Gaussian Bayes Classifier

• Given dataset $D = \{(x_1, y_1), \dots, (x_n, y_n)\}$

• $P(Y = y) = \hat{p}_y = \frac{\text{count}(Y=y)}{n}$

• $P(x; \hat{\mu}_y, \hat{\Sigma}_y) \Rightarrow \hat{\mu}_y = \frac{1}{\text{count}(Y=y)} \sum_{i: y_i = y} x_i$

$\Rightarrow \hat{\Sigma}_y = \frac{1}{\text{count}(Y=y)} \sum_{i: y_i = y} (x_i - \hat{\mu}_y)(x_i - \hat{\mu}_y)^T$

$P(X, Y) = P(Y) \cdot P(X \mid Y)$

Transformers

Word i	v_i (meaning)	q_i	k_i (predicted)
music	[.9, 0, .1]	[0, 0]	[1, 1]
rock	[.2, 0, .6]	[1, 0]	[0, .1]
like	[0, 0, 0]	[0, 1]	[0, 0]

$\text{score}_{i,j} \propto q_i \cdot k_j^T$ (attending / attended to)

↳ want $\text{score}_{i,j} \approx 0$ if $j > i$