REGRESSION

Linear

$L(w) = \frac{1}{n}\sum_i^n (y_i - w^T x_i)^2 = \frac{1}{n}\|y - Xw\|_2^2$

$\nabla_w L(w) = -\frac{2}{n}(w^T X - y_i)x_i = \frac{2}{n}(X^T X w - X^T y) \in \mathbb{R}^d$

$\nabla_w^2 L(w) = \frac{2}{n} X^T X$

$\hat{w} = (X^T X)^{-1} X^T y$

$o(d^3 + nd^2)$; $X^T X$ is invertible

Lasso (convex (↛ strongly if X is full rank))

$L(w) = \|y - \Phi w\|_2^2 + \lambda \|w\|_1$

Ridge (strictly convex, even if X not full rank)

$L(w) = \|y - \Phi w\|_2^2 + \lambda \|w\|_2^2$ → GD directly → conv. faster

$\hat{w} = (X^T X + \lambda I_d)^{-1} X^T y$ → better conditioned as changing λmax and λmin of Hessian

$\uparrow \lambda$

OPTIMIZATION

Gradient Descent (conv. $^{(*)}$ to stationary pt.)

$w^{t+1} = w^t - \eta_t \nabla_w L(w^t)$

Learning rate

$\|w^t + \hat{w}\|_2 \le \rho^t \|w^0 - \hat{w}\|_2$

$\rho = \|I - \eta X^T X\|_{op}$

$\eta \overset{(*)}{<} \frac{2}{\lambda_{max}(X^T X)}$

$\eta_{opt} = \frac{2}{\lambda_{max} + \lambda_{min}} \Rightarrow \rho_{opt} = \frac{K-1}{K+1}$; $K = \frac{\lambda_{max}}{\lambda_{min}}$

Momentum: dampends oscillations

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

Adaptive: $w_{i,t}$ changed a lot → smaller stepsize

SGD:

$w^{t+1} = w^t - \eta \nabla_b L(w^t)$

Training loss might increase at some iter.

Destroys. in S sampled uniformly w replacement

SGD update is a descent direction in expectation

Large batch size → small variance

Escape saddlepoints

Fewer steps than GD, but may take more and also not converge to right sol, but swirl

Convexity (loc → glob)

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

$L(w) \ge L(w_0) + \nabla L(w_0)^T (w - w_0)$

$\nabla^2 L(w) \ge 0$

Strong convexity → uniqueness

Complexity

GD: Steps τ req. to satisf. $\|w^\tau - w^*\| < \varepsilon$

$\tau = \frac{\log \varepsilon - \log\|w^0 - w^*\|}{\log \rho}$

Tot.: $\sigma(nd^2 + \tau d^2)$; Single: $\sigma(nd)$

CF: $\sigma(d^3)$; lin. sys. of eqs: $\sigma(nd^2)$

MODEL SELECTION

Expected estimation error

$E_X[l(\hat{f}_D(x), f^*(x))]$

Generalization error

$L(\hat{f}_D; P_{X,Y}) = E_{X,Y}[l(\hat{f}_D(x), Y)]$

$= \sum_{\vec{x}}\sum_y l(\hat{f}_D(\vec{x}), y) \cdot P(\vec{x}, y)$

$= \int l(\hat{f}_D(x), y) p(x,y) dx dy$

$= E_X[L(\hat{f}_D(x), f^*(x))] + \sigma^2$

Training error

$\frac{1}{|D_t|}\sum_{(x,y)\in D_t} l(\hat{f}_D, x, y)$

Cross validation

For all folds $k = 1, \dots, K$:

Train model $\hat{f}_{m,k} = A_m(D_{train} \setminus D_k)$

Validation error $L_k^m = \frac{1}{|D_k|}\sum_{(x,y)\in D_k} l(\hat{f}_{m,k}(x), y)$

Cross validation error $CV_K(M) = \frac{1}{K}\sum_k^K L_k(M)$

Model selection: Pick M^* with lowest $CV_K(M)$

Model training: Train model $\hat{f} = \hat{f}_{M^*, D_{train}}$

Model evaluation: Estimate gen. error using D_{test}

LOOCV: $K = |D_{train}|$ (deterministic)

- $M(D_k) \approx M(D_{train})$
- Poor L_k^n, but $CV_K(M)$ may be good gen err. est.
- Intensive computationally (trains $|D_{train}|$ models)

BIAS-VARIANCE TRADEOFF

Bias

$Bias^2(\hat{f}_0) = E_X[(f^*(x) - \bar{f}(x))^2]$; $\bar{f} = \frac{1}{J}\sum_{i=1}^J \hat{f}_i = \|w - E_\varepsilon[\hat{w}]\|_2^2$

Variance

$Var_D(\hat{f}_0) = E_X[\frac{1}{J}\sum_{j=1}^J (\hat{f}_j(x) - \bar{f}(x))^2] = E_\varepsilon[(\|\hat{w} - E_\varepsilon[\hat{w}]\|)^2]$

Bias-Variance decomposition

$E_D[L(\hat{f}_0; P_{X,Y})] = Var_D(\hat{f}_0) + Bias_D^2(\hat{f}_0) + \sigma^2$ ($\sigma^2 = Var(\varepsilon)$)

Unregularized

$Bias(w^*) = 0$; $Var = \sum_{i=1}^m \frac{\sigma^2}{\sigma_i^2}$ (σ_i: singular val. of X)

Ridge

$Bias^2(w_r^*) = \|\lambda(X^T X + \lambda I_d)^{-1} w\|^2 = \|V \cdot diag(\frac{\lambda}{\sigma_i^2 + \lambda}) V^T w^*\|_2^2$

$Var(w_r^*) = \sigma^2 \sum_{i=1}^m \frac{\sigma_i^2}{(\sigma_i^2 + \lambda)^2}$

CLASSIFICATION

$l_{0-1}(y\hat{f}(x)) = \mathbb{I}_{\{y\hat{f}(x)\}}$ (dis. cont. non. diff. non-conv.)

$l_{hinge}(y\hat{f}(x)) = \max\{0, 1 - y\hat{f}(x)\}$

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

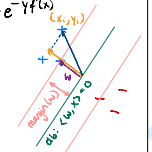
$l_{log}(y\hat{f}(x)) = \log(1 + e^{-y\hat{f}(x)})$

$= -\log(P(y|x))$; $P(y|x) = \frac{1}{1 + e^{-y\hat{f}(x)}}$

Max-margin (non-convex)

$w_{hm} = \underset{\|w\|_2=1}{\operatorname{argmax}} \min_i y_i \frac{\langle w, x_i\rangle}{\|w\|_2} = \underset{\|w\|_2=1}{\operatorname{argmax}} \min_i y_i \langle w, x_i\rangle = \underset{\|w\|_2=1}{\operatorname{argmax}} margin(w)$

unique



non-unique

↳ $w_{hm} \| \underset{w}{\operatorname{argmax}} \frac{margin(w)}{\|w\|_2} = \frac{margin(\alpha w)}{\|\alpha w\|_2}$; $\hat{w} = \alpha w$, $\alpha = \frac{1}{margin(w)}$

$margin(\hat{w}) = \min_i y_i \hat{w}^T x_i = \alpha \cdot \min_i y_i w^T x_i = \alpha \cdot margin(w) = 1$

min dist to db: $\frac{margin(\hat{w})}{\|\hat{w}\|_2} = \frac{1}{\|\hat{w}\|_2}$

SVM (convex): maximize min dist to db

$\hat{w} = \underset{w}{\operatorname{argmin}} \|\hat{w}\|_2$ s.t. $y_i \hat{w}^T x_i \ge 1$ (unique)

$margin(\hat{w}) = 1$

Multi-class classification

Prediction error: $l_{0-1}(\hat{y}, y) = \mathbb{I}_{\hat{y} \neq y}$

Training

One-vs-rest: have to run K opt. procedures

Cross-entropy loss

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

$= -\log(P(\hat{Y} = k | X))$

Metrics

Null hypothesis $H_0(x)$: $y = -1$ (the critical class)

	$y=-1$	$y=+1$
$\hat{y}=-1$	TN	FN
$\hat{y}=+1$	FP	TP

Error I/FPR $= \frac{FP}{TN+FP} = P(\hat{y}=1|y=-1)$

Error II/FNR $= \frac{FN}{TP+FN} = P(\hat{y}=-1|y=1)$ $= 1 - Recall$

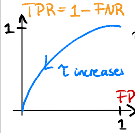
Precision $= \frac{TP}{TP+FP} = P(y=1|\hat{y}=1)$

Recall/TPR $= \frac{TP}{TP+FN} = P(\hat{y}=1|y=1)$

FDR $= \frac{FP}{FP+TP} = P(y=-1|\hat{y}=1)$ $= 1 - precision$

Accuracy $= \frac{TP+TN}{P+N}$ large ↑ ↳ many $\hat{y}=-1$ ↳ FN inc ↳ FP dec

TPR = 1 − FNR



$\hat{y}_\tau(x) = \begin{cases} +1, & \hat{f}(x) > \tau \\ -1, & \hat{f}(x) < \tau \end{cases}$

$F1 = \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}} = \frac{2TP}{2TP + FP + FN}$

KERNES

$p = \sigma\binom{d+m}{m} = (d+m)C(m) \approx \begin{cases} \sigma(d^m) & d \gg n \\ \sigma(m^d) & m \gg d \end{cases}$ $\phi(x) \in \mathbb{R}^p$

Classification: $w = \sum_{j=1}^n \alpha_j y_j x_j$

Kernel trick

1. Mem. red. ($\sigma(np) \sim \sigma(nd^m) \to \sigma(n^2)$)

$\hat{w} = \underset{w \in \mathbb{R}^d}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^n l(y_i, w^T \phi(x_i))$ ($f(x_i)$) $\quad w = \Phi^T \alpha$

$\hat{\alpha} = \underset{\alpha \in \mathbb{R}^n}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^n l(y_i, \alpha^T \Phi \phi(x_i))$

$= \underset{\alpha \in \mathbb{R}^n}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^n l(y_i, \sum_{j=1}^n l(y_i, \alpha_j \langle \phi(x_j), \phi(x_i)\rangle))$

$f(x) = w^T \phi(x) = \alpha^T \Phi \phi(x) = \sum_{i=1}^n \alpha_j \langle \phi(x_j), \phi(x_i)\rangle$

↳ minimizer only dep. on the feature vectors via their inner prod

2. Comp. red. ($\sigma(n^2 p) \sim \sigma(n^2 d^m) \to \sigma(n^2(d+m))$)

$\langle \phi(x), \phi(z)\rangle = K(x,z)$, $\mathcal{K} = \Phi\Phi^T \in \mathbb{R}^{n \times n}$

$\mathcal{K}_{ij} = K(x_i, x_j)$

Kernelized Ridge regression

$\hat{\alpha} = \underset{\alpha \in \mathbb{R}^n}{\operatorname{argmin}} \frac{1}{n}\|y - K\alpha\|_2^2 + \lambda \alpha^T K \alpha$

Properties of kernels

1. Symmetric: $K(x,z) = K(z,x)$
2. Kernel function K is psd $\Leftrightarrow$ $\mathcal{K}$ is psd for any inputs $x_1, \dots, x_n$ and any $n \in \mathbb{N}$

Valid kernels

Constant	$K(x,y) = C$	$C \ge 0$
Inner product/linear	$K(x,y) = x^T y$	
Polynomial	$K(x,y) = (x^T y + 1)^m$	G: $\alpha = 2$, L: $\alpha = 1$
RBF	$K(x,y) = \exp(-\frac{\|x-y\|_2^\alpha}{\tau})$	bandwidth

All valid Kernels induce feature maps

Feature map Gaussian kernel: $\phi_j(x) = \frac{1}{\sqrt{j!}} e^{-\frac{x^2}{2}} x^j$; $j = 1, \dots, \infty$

Composition rules

$K(x,y) = K_1(x,y) + K_2(x,y)$

$K(x,y) = K_1(x,y) \cdot K_2(x,y)$

$K(x,y) = c K_1(x,y)$; $C > 0$

$K(x,y) = f(K_2(x,y))$; f is exponential or polynomial with pos. coeff.

$K((x_1, y_1), (x_2, y_2)) = K_1(x_1, x_2) \cdot K_2(y_1, y_2)$

$K((x_1, y_1), (x_2, y_2)) = K_1(x_1, x_2) + K_2(y_1, y_2)$

$K(x,y) = f(x) K_1(x,y) f(y)$ $\forall f$

NON-LINEAR NON PARAM METHODS

KNN: For given x, find kNN and output majority label

- sensitive to k → choose using CV
- erratic in high. dim.
- need high n to preform well

Decision Trees

Split data along canonical coord. until low uncert. Each parent node splits data with chosen rule, each child node returns its majority label

- leaf nodes are small → prone to overfit noise → choose # leaves wisely

NEURAL NETWORK

Activation functions

Identity: $\varphi(z) = z$

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

Tanh: $\varphi(z) = \frac{\exp(z) - \exp(-z)}{\exp(z) + \exp(-z)}$

ReLu: $\varphi(z) = \max(0, z)$

Universal Approximation Theorem

nn with 1(+) layer can approx. any cont. func.

Forward propagation

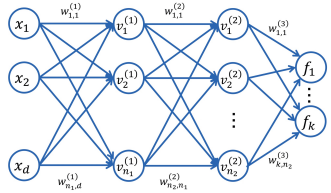
Input: $v^{(0)} = [x_i ; 1]$

Hidden: $z_j^{(L)} = \sum_{i=0}^{m_{L-1}} W_{ji}^{(L)} v_i^{(L-1)}$

$v^{(L)} = [\varphi(z^{(L)}) ; 1]$ ($L = 1, \dots, L-1$)

Output: $f = \sum_{i=0}^{m_{L-1}} W_i^{(L)} v_i^{(L-1)}$

$W^{(L)} = \begin{bmatrix} w_{11}^{(L)} & \cdots & w_{1,m_{L-1}}^{(L)} \\ \vdots & & \vdots \\ w_{m_L,1} & & w_{m_L, m_{L-1}}^{(L)} \end{bmatrix} \in \mathbb{R}^{d_L \times d_{L-1}}$



Optimization (non-convex → initialization matters)

$W^* = \underset{W}{\operatorname{argmin}} \sum_{i=1}^n l(W; x_i, y_i)$

Weight space symmetries → multiple local AND global opt

Multiple distinct weights compute the same predictions

- permuting units on hidden layer preserves f

Multiple local-minima can be equivalent in terms of input-output mapping

- φ is symmetric $\varphi(z) = -\varphi(z)$

Backpropagation

Output layer L:

$\delta_j^{(L)} = \frac{\partial l}{\partial f_j}$, $\frac{\partial l}{\partial w_{ji}^{(L-1)}} = \delta_j^{(L)} v_i^{(L-1)}$

Hidden layers $l = L-1 : 1$:

$\delta_j^{(l)} = \varphi'(z_j^{(l)}) \sum_{i=1}^{n_{l+1}} w_{ij}^{(l)} \delta_i^{(l+1)}$

$\frac{\partial l}{\partial w_{ji}^{(l-1)}} = \delta_j^{(l)} v_i^{(l-1)}$

SGD: $W_i^{(t+1)} \leftarrow W_i - \eta \nabla_{W_i} L(W^{(t)})$

Weight initialisation

Goal: keep variance of weights approximately constant across layers

How: Control the norm of v by randomly initialize the weights

Ex: ReLu, $E[x_i] = 0$, $Var[x_i] = 1$

$\Rightarrow w_i \overset{iid}{\sim} \mathcal{N}(0, \sigma^2) = \mathcal{N}(0, \frac{2}{d})$

Initialization schemes

Tanh: $w_i \sim \mathcal{N}(0, \frac{1}{n_{in}})$ or $w_i \sim \mathcal{N}(0, \frac{2}{n_{in} + n_{out}})$

ReLu: $w_i \sim \mathcal{N}(0, \frac{2}{n_{in}})$

Learning rate

- Decaying learning rate schedule
- Monitor ratio of weight change (gradient) to weight magnitude
- Learning with momentum

Regularization

- Regularization
- Early stopping: Stop when val. loss stops decreasing
- Drop out:
  - Training: randomly ignore hidden units with prob p
  - Test: Keep all units, but multiply weights by p

Batch normalization: (during training)

- Noralizes unit activations in each layer
- "Reduces internal covariate shift", enables larger learning rates, regularizing effect, alleviates vanishing grad. prob.

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

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

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

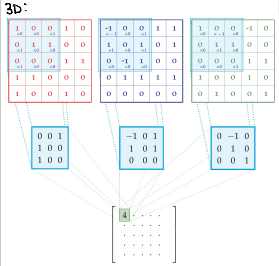
Scale and shift: $\overline{v}_i = \gamma \hat{v}_i + \beta$

Convolutional nn

1D: $w \in \mathbb{R}^k$, $x \in \mathbb{R}^d$, $z \in \mathbb{R}^{d+k+1}$

$z_i = \sum_{j=\max(1, i-d+1)}^{\min(i,k)} w_j \cdot x_{i-j+1}$

3D:



Output dim.: $l = \frac{n + 2p - f}{s}$

Parameters

Conv: $(\#C_{in} \cdot k_W \cdot k_H \cdot k_0 + 1) \cdot \#Filter$

Fully: $(\#C_{in} \cdot W \cdot H \cdot D + 1) \cdot \#C_{out}$

ResNets

Residual connections: $v^{(l+1)} = v^{(l)} + r(v^{(l)})$ (eg $r(v) = w^2 \varphi(Wv)$) +1000 layers

Introduce skip connections

↳ Efficiently train deep networks

↳ "Indentity shortcut" → avoid vanishing grad.

CLUSTERING

Optimization (non-convex, NP hard)

$\hat{R}(\mu) = \sum_{i=1}^n \min_{j \in \{1, \dots, K\}} \|x_i - \mu_j\|_2^2$

$= \sum_{i=1}^n \|x_i - \mu_{\hat{c}_i}\|_2^2$

Lloyd's heuristic

1. Init. clustering centers $\mu^{(0)} = [\mu_1^{(0)}, \dots, \mu_K^{(0)}]$
2. While not converged:

Assign x_i to closest center

$\hat{c}_i \leftarrow \underset{j \in \{1, \dots, K\}}{\operatorname{argmin}} \|x_i - \mu_j^{(t-1)}\|_2$

Update centers as mean of assigned pts

$\mu_j^{(t)} \leftarrow \frac{1}{n_j}\sum_{i : \hat{c}_i = j} x_i$

Properties

- Guaranteed to monotonically decrease $\hat{R}(\mu^{(t)}) \ge \hat{R}(\mu^{(t+1)})$
- Cost per iteration: $\sigma(n \cdot k \cdot d)$
- Converges to local opt. → dep. init!
- # iter required can be exponential
- Clusters of arbitrary shape → kernel
- Determining K
  - ↳ "Diminishing returns" in loss func.
  - ↳ Regularization: $K \to \lambda$ (more robust)

K-Means++

Add $\mu_2, \dots, \mu_K$ randomly, prop. to sqrt. dist to closest already selected center

Given $\mu_{1:j}^{(0)}$, pick $\mu_{j+1}^{(0)} := x_i$ where $P(x_i) \propto \min_{i \in \{1, \dots, j\}} \|x_i - \mu_i^{(0)}\|_2^2$

Expected cost

$\hat{R}(\mu_{K\text{-means++}}) \le \sigma(\log K) \cdot \min_\mu \hat{R}(\mu)$

DIMENSION REDUCTION

Data is assumed centered

PCA: K=1

$w^*, z_1^*, \dots, z_n^* = \underset{\|w\|_2=1, z}{\operatorname{argmin}} \sum_{i=1}^n \|x_i - z_i w\|_2^2$ ($\bar{x}_i$)

$w^* = \underset{\|w\|_2=1}{\operatorname{argmin}} \sum_{i=1}^n \|x_i - w w^T x_i\|_2^2$

$z_i = \frac{w^T x_i}{\|w\|_2} = \underset{\|w\|_2=1}{\operatorname{argmax}} w^T \Sigma w$

$= w^T x_i = v_1$

Σ is sym. and psd ↳ eigen-decomposition

$\Sigma = \frac{1}{n}\sum_{i=1}^n x_i x_i^T = \frac{1}{n} X^T X = \sum_{i=1}^d \lambda_i v_i v_i^T$

PCA: arbitrary K

$(W, z_1, \dots, z_n) = \underset{W^T W = I_K, z}{\operatorname{argmin}} \sum_{i=1}^n \|W z_i - x_i\|_2^2$ ($\bar{x}_i$)

$W = (v_1 | \cdots | v_K) \in \mathbb{R}^{d \times K}$, $z_i = W^T x_i$

Note: $W^T W = I_K$; WW^T: projection matrix

Note: Minimizer not unique $(+W, -W)$

SVD

First K columns of V where $X = USV^T$

$\Rightarrow$ j-th eigenvalue of $\frac{1}{n} X^T X$: $\lambda_j = \frac{\sigma_j^2}{n}$

Empirical variance

Empirical variance along the j-th principal component direction is equal to the j-th eigenvalue λ_j of Σ:

$\frac{1}{n}\sum_{i=1}^n \|v_j v_j^T x_i\|_2^2 = \frac{1}{n}\sum_{i=1}^n (v_j^T x_i)^2$

$= v_j^T \Sigma v_j = \lambda_j$

Reconstruction error = variance of data along the excluded components

$L^{(k)} = \sum_{j=k+1}^d \lambda_j$

$\Rightarrow L^{(k-1)} - L^{(k)} = \lambda_k$

$\Rightarrow L^{(0)} - L^{(k)} = \sum_{j=1}^k \lambda_j$

Kernel PCA

$\alpha^* = \underset{\alpha^T K \alpha = 1}{\operatorname{argmax}} \alpha^T K^T K \alpha = \underset{\alpha}{\operatorname{argmax}} \frac{\alpha^T K^T K \alpha}{\alpha^T K \alpha}$

Closed form sol. from eigendecomp of K

$\alpha^{(i)} = \frac{1}{\sqrt{\lambda_i}} v_i \in \mathbb{R}^n$; $K = \sum_{i=1}^n \lambda_i v_i v_i^T$

New point x is projected as $z \in \mathbb{R}^K$

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

Center the kernel

$K' = K - KE - EK + EKE \Rightarrow K'(x,x') = K(x,x') + K(x_i,x) + K(x,x_j) + K(x_i,x_j)$

Autoencoders

If the activation function is the identity and the loss is the squared reconstruction error, then fitting an autoencoder with bottleneck m is equiv. to preforming PCA with m components

MAXIMUM LIKELIHOOD ESTIM.

Likelihood of the data for given distr. $p(\cdot;\theta)$

$$p(D;\theta) = \prod_{i=1}^n p(x_i, y_i;\theta) \quad : \mathbb{P}_X$$
$$= \prod_{i=1}^n p(y_i|x_i;\theta) \quad : \mathbb{P}_{Y|X}$$

Maximum likelihood param. est.

$$\hat\theta_{MLE} = \underset{\theta}{\operatorname{argmax}}\ \log p(D;\theta)$$

Unsupervised learning, $\mathbb{P}_X$

Gaussian distribution, $\mathcal{N}(\mu,\sigma^2)$

$$\hat\mu_{MLE} = \frac{1}{n}\sum_{i=1}^n x_i \ , \quad \hat\sigma^2_{MLE} = \frac{1}{n}\sum_{i=1}^n (x_i - \mu_{MLE})^2$$

Supervised learning, $\mathbb{P}_{XY}$

(*) Gaussian noise model (regression)

Assume $y = w^T x + \epsilon \ ; \ \epsilon \sim \mathcal{N}(0,\sigma_\epsilon^2)$

$$\Rightarrow p(y|x;w) \sim \mathcal{N}(w^T x, \sigma_\epsilon^2)$$

squared loss, lin. reg $\Rightarrow \hat w_{MLE} = \underset{w}{\operatorname{argmin}}\ (y_i - w^T x_i)^2$

Logistic noise model (classification)

$$\Rightarrow p(y|x;\theta) = \sigma(y\theta^T x) = \frac{1}{1+e^{-y\theta^T x}}$$

$$\Rightarrow \hat y(x) = \underset{y\in\{-1,1\}}{\operatorname{argmax}}\ \hat p(y|x)$$

logistic loss, logistic reg $\Rightarrow \hat\theta_{MLE} = \underset{\theta\in\mathbb{R}^d}{\operatorname{argmax}} \sum_{i=1}^n \log(1 - e^{-y_i\theta^T x_i})$

MAXIMUM A POSTERIORI ESTIM.

Posterior distribution of θ given data

$$p(\theta|D) = \frac{p(D,\theta)}{p(D)} = \frac{p(D|\theta)\cdot p(\theta)}{p(D)}$$

$$\mathbb{P}_X = \frac{\prod_{i=1}^n p(x_i|\theta)\cdot p(\theta)}{\int \prod_{i=1}^n p(x_i|\theta)\cdot p(\theta)\,d\theta} \quad \left| \quad \mathbb{P}_{Y|X} = \frac{\prod_{i=1}^n p(y_i|x_i,\theta)\cdot p(\theta)}{\int \prod_{i=1}^n p(y_i|x_i,\theta)\cdot p(\theta)\,d\theta}\right.$$

Maximum a posteriori param est.

$$\hat\theta_{MAP} = \underset{\theta}{\operatorname{argmax}}\ \log p(\theta|D) = \underset{\theta}{\operatorname{argmax}}\ \log p(D,\theta)$$

Supervised distribution fam. $\mathcal{P}$ for $\mathbb{P}_{XY}$

(*) + Gaussian prior : $w \overset{iid.}{\sim} \mathcal{N}(0, \sigma_w^2 I_d)$

$$\Rightarrow \hat w_{MAP} = \underset{w}{\operatorname{argmin}}\ \frac{1}{2\sigma_\epsilon^2}\|y - Xw\|_2^2 + \frac{1}{2\sigma_w^2}\|w\|_2^2$$

$\lambda = \frac{\sigma_\epsilon^2}{\sigma_w^2}$ $= \underset{w}{\operatorname{argmin}}\ \|y - Xw\|_2^2 + \frac{\sigma_\epsilon^2}{\sigma_w^2}\|w\|_2^2$

(*) + Laplacian prior : $w \sim \text{Laplace}(0, \sigma_w)$

$$\Rightarrow \hat w_{MAP} = \underset{w}{\operatorname{argmin}}\ \frac{1}{2\sigma_\epsilon^2}\|y - Xw\|_2^2 + \frac{1}{\sigma_w}\|w\|_1$$

$\lambda = \frac{2\sigma_\epsilon^2}{\sigma_w}$ $= \underset{w}{\operatorname{argmin}}\ \|y - Xw\|_2^2 + \frac{2\sigma_\epsilon^2}{\sigma_w}\|w\|_1$

Bayesian modeling averaging

Bayesian modeling allow outputting a distr. of possible θ
↳ Instead of point est., posterior $p(\theta|D)$ automatically gives a distribution over distributions

Bayesian model averaging for $\hat{\mathbb{P}}_X$ and $\hat{\mathbb{P}}_{Y|X}$, respectively

$$\hat p(x|D) = \hat{\mathbb{E}}_{\theta|D}[p(x|D)] = \int_\theta \underbrace{p(x|\theta)}_{\text{by choice of model}} \underbrace{\hat p(\theta|D)}_{\text{computed from data}} d\theta$$

$$\hat p(y|x,D) = \hat{\mathbb{E}}_{\theta|D}[p(y|x,\theta)] = \int_\theta p(y|x,\theta)\,\hat p(\theta|D)\,d\theta$$

Conjugate prior: yield a posterior within the same distr. as given likelihood function

GENERATIVE MODELING

Model $p(y,x) = p(x|y)\cdot p(y)$

Estimate prior on labels: $p(y)$
Estimate cond. distr. for each class y: $p(x|y)$
Pred. distr using Bayes: $p(y|x) = \frac{p(y)\cdot p(x|y)}{p(x)}$ (*z)
Decision rule: $\hat y = \underset{y'}{\operatorname{argmax}} P(y'|x) = \underset{y'}{\operatorname{argmax}} P(y')\,P(x|y')$
$= \underset{y'}{\operatorname{argmax}} \log P(y') + \log P(x|y')$

Gaussian Naive Bayes (classification)

Naive: for each class label, each feature is indep.
↳ $\Sigma_y = \text{diag}(\sigma_{y,1}^2, \dots, \sigma_{y,d}^2)$
↳ fail to capture correlation → overly confident
Model class label as generated from cat. variable: $P(Y=y) = p_y$
Model features by (conditionally) independent Gaussians
$P(x_i|y) = \mathcal{N}(x_i|\mu_{y,i}, \sigma_{y,i}^2)$

MLE: $\hat p_{y\,MLE} = \frac{\#\{Y=y\}}{n}$

$$\hat\mu_{y\,MLE} = \frac{1}{\#Y=y}\sum_{i:y_i=y} x_i$$

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

Compare with log reg.: smaller uncertainty band, right shape

Gaussian Bayes classifier / QDA

Model class label as generated from cat. variable: $P(Y=y) = p_y$
Model features by multivariate Gaussians
$P(x_i|y) = \mathcal{N}(x_i|\mu_y, \Sigma_y)$

MLE: $\hat p_y = \frac{\text{Count}(Y=y)}{n}$

$$\hat\mu_y = \frac{1}{\text{Count}(Y=y)}\sum_{i:y_i=y} x_i$$

$$\hat\Sigma_y = \frac{1}{\text{Count}(Y=y)}\sum_{i:y_i=y}(x_i - \hat\mu_y)(x_i - \hat\mu_y)^T$$

Linear Discriminant analysis

$\sigma_y = \sigma$ $(\Sigma_1 = \dots = \Sigma_K)$ for all $y \Rightarrow$ linear dlo

Poisson Naive Bayes

$$\hat p_y = \frac{\text{Count}(Y=y)}{n} \ , \quad \lambda_{y,j} = \frac{\sum_{i:y_i=y} x_j^{(i)}}{\text{Count}(Y=y)}$$

DECISION THEORY

$$a^* = \underset{a\in\mathcal{A}}{\operatorname{argmin}}\ \hat{\mathbb{E}}[C(y,a)|X=x]$$
$$= \underset{a\in\mathcal{A}}{\operatorname{argmin}}\ \sum_y C(y,a)\cdot P(Y=y|X=x)$$
$$= \underset{a\in\mathcal{A}}{\operatorname{argmin}}\ \sum_y C(y_i,a)\cdot P(Y=y, X=x)$$

GAUSSIAN MIXTURE M.

Assume data is generated from a convex-combination of Gaussian distributions:

$$p(x|\theta) = P(x|\mu,\Sigma,w) = \sum_{j=1}^k w_j \mathcal{N}(x;\mu_j,\Sigma_j)$$

Objective (non-convex, not closed form) | x_i are iid

$$L(w_{1:k},\mu_{1:k},\Sigma_{1:k}) = -\sum_{i=1}^n \log \sum_{j=1}^k w_j \mathcal{N}(x_i|\mu_j,\Sigma_j)$$

Fitting a GMM = training a GBC without labels

Hard-EM

E-step: Predict most likely class for each data pts.

$$z_i = \underset{z}{\operatorname{argmax}}\ P(z|x_i,\theta^{(t-1)})$$
$$= \underset{z}{\operatorname{argmax}}\ P(z|\theta^{(t-1)})\cdot P(x_i|z,\theta^{(t-1)})$$

M-step: Compute MLE as for GBC (closed form)

$$\theta^{(t)} = \underset{\theta}{\operatorname{argmax}}\ P(D^{(t)}|\theta)$$

Problems: assigns fixed labels, even when model is uncertain
↳ works poorly if clusters are overlapping

Expectation maximization

E-step: Calculate cluster membership weights for each pt.

$$\gamma_j(x) = P(z=j|x,\Sigma,\mu,w) = \frac{P(X=x|z=j)\cdot P(z=j)}{P(X=x)}$$
$$= \frac{w_j P(x|\Sigma_j,\mu_j)}{\sum_{l=1}^k w_l P(x|\Sigma_l,\mu_l)}$$

$\gamma_j(x_i) = [j = y_i]$ (SSL)

M-step: Fit clusters to weighted data pts (closed form MLE)

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

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

Constrained GMMs

(spherical, diff. size) ← (ellipse is axis-aligned)
(same shape and size) ← (sym)
(linear dlo)

		#params ($\Sigma_{1:k}$)
Spherical	$\Sigma_i = \sigma_i^2 I$	K
Diagonal	$\Sigma_i = \text{diag}(\sigma_{i,1}^2,\dots,\sigma_{i,d}^2)$	kd
Tied	$\Sigma_i = \dots = \Sigma_K$	$\frac{d(d+1)}{2}$
Full	w, μ, Σ arbitrary	$K\cdot\frac{d(d+1)}{2}$

K-means

Special case of Hard-EM
- uniform weights $w_{1:K} = \frac{1}{K}$ and identical, spherical covariances $\Sigma_1 = \dots = \Sigma_K = \sigma^2 I$

Limiting case of soft-EM
+ variances tending to 0: $\sigma \to 0$

Initialization

Weights: uniform distribution
Means: randomly, K-means++
Variances: spherical, e.g. according to empirical variance in data
K: cross-validation (only works if data truly gen. from a GMM)

Degeneracy

Problem: loss function not lower bounded:
↳ loss converge to $-\infty$ as $\mu = x$ and $\sigma \to 0$
↳ "Optimal" GMM chooses $K = n \to$ Overfitting

Solution:
- lower bound the width of the Gaussians
↳ add $\nu^2 I$ to variance Σ_j (regularizer)

Gaussian-Mixture Bayes Classifier

Estimate class prior $P(y)$
Estimate cond. distr. for each class as GMM:

$$P(x|y) = \sum_{j=1}^{k_y} w_j^{(y)} \mathcal{N}(x;\mu_j^{(y)},\Sigma_j^{(y)})$$

Imputing missing features

Covariance matrix

$$\Sigma = \begin{bmatrix} \text{Var}[X_1] & \text{Cov}[X_1,X_2] \\ \text{Cov}[X_1,X_2] & \text{Var}[X_2]\end{bmatrix}$$

$$\mathbb{E}[X_1|X_2 = x_2] = \mu_1 + \frac{\text{Cov}(X1,X2)}{\text{Var}(X2)}(x_2 - \mu_2)$$

Convergence of EM

Monotonically increases the likelihood → loc. optim.
↳ initialization matter → rerun and use sol. with largest LH

General EM

E-step: Calculate expected complete data log-likelihood

$$Q(\theta;\theta^{(t-1)}) = \mathbb{E}_{z_{1:n}}[\log P(x_{1:n}, z_{1:n}|\theta)|x_{1:n},\theta^{(t-1)}]$$
$$= \sum_{i=1}^n \sum_{j=1}^k \underbrace{p(z_i=j|x_i,\theta^{(t-1)})}_{\gamma_j(x_i)} \cdot \log \underbrace{p(x_i,z_i|\theta)}_{w_j\mathcal{N}(x_i;\mu_j,\Sigma_j)}$$

M-step: Maximize

$$\theta^{(t)} = \underset{\theta}{\operatorname{argmax}}\ Q(\theta;\theta^{(t-1)})$$

LLMs

- Joint distr. over dependent categorical rand. var.

$$P(X_t = x | X_{1:t-1} = x_{1:t-1}) \approx P(X_t = x | X_{t-k:t-1} = x_{t-k:t-1}, \theta)$$
$$:= \text{Cat}(x|\text{softmax}(f(x_{t-k:t-1},\theta)))$$

Transformer model

Input: $Z_0 = XW_e + W_p$ (We: learnable word emb., Wp: fixed pos. emb.)
Trans. blok: $Z_l = \text{transformer\_block}(Z_{l-1}) \ \forall i \in \{1,\dots,n\}$
Output: $P = \text{softmax}(Z_n W_e^T)$

Transformer block: "masked multi-head self attention"
Attention mechanism learns to predict a weighted, directed graph
- let words borrow features of words it attends to
Self-attention: restricts the nodes on both sides to be the same
Masking: only allow to attend to previous words

$$z_i^{l+1} = \sum_{j=1:k} \text{score}_{ij}\, z_j^l \ ; \ \text{score}_{ij} \propto \exp\left(\frac{q_i k_j^T}{\sqrt{d_k}} + m_{ij}\right)$$

$$m_{ij} = \begin{cases} -\infty & \text{if } j > i \\ 0 & \text{otherwise}\end{cases}$$

MATH

Probabilities

$P(x)$: Evidence ; $P(y)$ = Prior
$P(x|y)$: Likelihood ; $P(y|x)$ = Posterior

KL-divergence

$$KL(p|q) = \sum_x p(x)\cdot\frac{p(x)}{q(x)} \geq 0, \quad KL(p|q) = 0 \text{ if } p = q$$

Distributions

Gaussian: $f(x;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}}\cdot\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$

Poisson: $f(k;\lambda) = \frac{\lambda^k e^{-\lambda}}{k!}$;

$$P(X \leq \tau) = F(\tau) = 1 - e^{-\lambda\tau}$$

Cont. prob: $P(z_i) = P(z_i | z_i \geq \tau)\cdot P(z_i \geq \tau)$

Expectation and variance

$$\text{Var}[X] = \mathbb{E}[X^2] - (\mathbb{E}[X])^2 = \frac{1}{N-1}\sum_{i=1}^N (x_i - \overline{X})^2$$
$$\text{Var}[aX+b] = a^2\,\text{Var}[X]$$
$$\text{Var}[X+Y] = \text{Var}[X] + \text{Var}[Y] + 2\cdot\text{Cov}(X,Y)$$
$$\mathbb{E}[X\cdot Y] = \mathbb{E}[X]\cdot\mathbb{E}[Y]$$
$$\text{Cov}(X,Y) = \frac{1}{N-1}\sum_{i=1}^N (X_i - \overline{X})(Y_i - \overline{Y})$$

Bayes Theorem

$$p(y|x) = \frac{1}{p(x)}\cdot\underbrace{p(y)\cdot p(x|y)}_{p(x,y)} = \frac{p(y)\cdot p(x|y)}{\sum_{y'} p(y')p(x|y')}$$

Linear alg:

X has full rank $\Leftrightarrow X^TX$ invertible $\Leftrightarrow X^TX$ is pd $\Leftrightarrow$ $L(w) = \frac{1}{n}\|y - Xw\|_2^2$ is strongly convex

XX^T is always at least psd!

Inverse

$$A = \begin{bmatrix} a & b \\ c & d\end{bmatrix} \ , \ A^{-1} = \det(A)\begin{bmatrix} d & -b \\ -c & a\end{bmatrix}$$

A diagonal matrix is invertible only if all its diagonal elements are non-zero

Positive (semi-) definite matrices

M is p.s.d $\Leftrightarrow \forall x \in \mathbb{R}^n : x^TMx \geq 0 \Leftrightarrow \lambda_i \geq 0$

pd if and only if its trace AND determinant are positive > 0

Eigenvectors and eigenvalues

$$Av = \lambda v \Leftrightarrow A^{-1}v = \frac{1}{\lambda}v$$

SVD:

$$X \in \mathbb{R}^{n\times d} = USV^T \leftarrow d\times d$$
(n×n, n×d)
$U^TU = UU^T = I_{n\times n}$, $V^TV = VV^T = I_{d\times d}$

$X^TX = VS^2V \Leftrightarrow$ V is the eigenvectors of X^TX with eigenvalues $\sigma_i^2 = S_{i,i}^2$
(A is s.m.)

Norm

$\|A\|_{op} = \max\{|\lambda_{max}(A)|, |\lambda_{min}(A)|\}$

$\|w\|_2 \leq \|w\|_1 \leq \sqrt{\|w\|_0}\cdot\|w\|_2$

Matrix:

$(AB)^{-1} = B^{-1}A^{-1}$, $(AB)^T = B^TA^T$
$(A^T)^{-1} = (A^{-1})^T$, $\text{Tr}(AB) = \text{Tr}(BA)$

Linear approximation of loss around point w^{now}

$$L(w^{next}) = L(w^{now} + \tilde\eta\tilde v) \approx L(w^{now}) + \tilde\eta\langle\nabla L(w^{now}), v\rangle$$

Convexity

$\alpha f + \beta g$; $\alpha,\beta \geq 0$, convex if f,g convex
$f\circ g$, convex if f convex and g affine or f non-dec. and g convex
$\max(f,g)$, convex if f,g convex