MODEL ERROR

Empirical Risk:

$$\hat{R}_{\mathcal{D}}(w) = \sum_{i=1}^{n} \left(w^\top x_i - y_i\right)^2$$

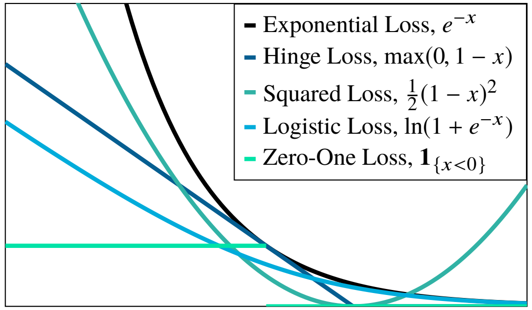
Generalisation Error (Pop. Risk):

$$L\left(f; \mathbb{P}_{X,Y}\right) = \mathbb{E}_{X,Y}\ell(f(X), Y)$$

Bias-Variance Tradeoff:

$$\mathbb{E}_{\mathcal{D}}[L(\hat{f};\cdot)] = \mathbb{E}_{X,\mathcal{D}}[(\hat{f}(X) - \mathbb{E}_{\mathcal{D}}[\hat{f}(X)])^2]$$
$$+ \mathbb{E}_X[(\mathbb{E}_{\mathcal{D}}[\hat{f}(X)] - f^*(X))^2] + \sigma$$
$$= \mathrm{Var}_{\mathcal{D}}(\hat{f}) + \mathrm{Bias}^2_{\mathcal{D}}(\hat{f}) + \mathrm{Noise}$$

Least Squares: $\mathbf{X}^\top\mathbf{X}\mathbf{w} = \mathbf{X}^\top\mathbf{y}$



REGULARIZATION

Lasso Regression (sparse):

$$\min_{w\in\mathbb{R}^d} \left(\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2 + \lambda\|\boldsymbol{w}\|_1\right)$$

Ridge Regression (more precise):

$$\min_{w\in\mathbb{R}^d} \left(\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2 + \lambda\|\boldsymbol{w}\|_2^2\right)$$
$$\nabla_w L(w) = 2\mathbf{X}^\top(\mathbf{X}w - \mathbf{y}) + 2\lambda\mathbf{w}$$

Solution: $\hat{\mathbf{w}} = (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y}$

large $\lambda \Longrightarrow$ larger bias, smaller variance

K-Fold Cross-Validation

Split Dataset into K sets (# methods), for each method, go through all sets and train it excluding that set and validating that set. Sum up all the validation errors of that method and choose smallest sum.

GRADIENT DESCENT

Converges only for convex case.

$$\mathbf{w}^{t+1} = \mathbf{w}^t - \eta_t \cdot \nabla\ell(\mathbf{w}^t)$$

For linear regression:

$$\|\mathbf{w}^t - \mathbf{w}^*\|_2 \leq \|\mathbf{I} - \eta\mathbf{X}^\top\mathbf{X}\|_{op}^t\|\mathbf{w}^0 - \mathbf{w}^*\|_2$$

$\exists\eta$ with conv. speed $\rho = \|\mathbf{I} - \eta\mathbf{X}^\top\mathbf{X}\|_{op}^t$.

$\eta_{\text{opt}} = \frac{2}{\lambda_{\min}+\lambda_{\max}}$ and max. $\eta \leq \frac{2}{\lambda_{\max}}$.

Momentum:

$$\mathbf{w}^{t+1} = \mathbf{w}^t + \gamma\Delta\mathbf{w}^{t-1} - \eta_t\nabla\ell(\mathbf{w}^t)$$

MAXIMUM-MARGIN SOLUTION

If linearly separable, we can get:

$$w_{\text{MM}} := \underset{\|w\|_2=1, w_0}{\arg\max} \min_{1\leq i\leq n} y_i(\boldsymbol{w}^\top\boldsymbol{x}_i + w_0)$$

Hard SVM

$\hat{\mathbf{w}} = \min_{\mathbf{w}} \|\mathbf{w}\|_2$ s.t. $\forall i, y_i\mathbf{w}^\top\mathbf{x}_i \geq 1$

Soft SVM (allows "slack" in constraints)

$$\hat{\mathbf{w}} = \min_{\mathbf{w},\xi} \tfrac{1}{2}\|\mathbf{w}\|_2^2 + \lambda\sum_{i=1}^{n} \max\left(0, 1 - y_i\mathbf{w}^\top\mathbf{x}_i\right)$$

Metrics

	True Class y		
	y=+1	y=-1	
f(x)=+1	TP	FP	Precision : $\frac{\text{TP}}{\text{TP+FP}}$
f(x)=+1	FN	TN	TPR/Recall: $\frac{\text{TP}}{\text{TP+FN}}$

err$_1$/FPR: $\frac{\text{FP}}{\text{TN+FP}}$, err$_2$/FNR: $\frac{\text{FN}}{\text{TP+FN}}$

ROC: Plot TPR=1-FNR vs. FPR and compare different ROC's with area under the curve.

F1-Score: $\frac{2\text{TP}}{2\text{TP+FP+FN}}$, Accuracy : $\frac{\text{TP+TN}}{\text{P+N}}$

Goal: large recall and small FPR.

KERNELS

$\exists\widehat{\alpha}.\ \widehat{\mathbf{w}} = \mathbf{\Phi}^\top\widehat{\alpha}$, $\mathbf{K} = \mathbf{\Phi}\mathbf{\Phi}^\top$, **Validity:**

1. $\mathbf{K}$ symmetric, $\forall x, z.\ k(x,z) = k(z,x)$
2. $\mathbf{K}$ positive semidef. (psd.), $\forall\mathbf{z}.\ \mathbf{z}^\top\mathbf{K}\mathbf{z} > 0$

lin.: $k(x,z) = x^\top z$, **poly.**: $k(x,z) = \left(x^\top z + 1\right)^m$

$$\exp\left(-\frac{\|\mathbf{x}-\mathbf{z}\|_p^2}{\tau}\right) = \begin{cases}\textbf{Laplacian Ker.} & p = 1\\ \textbf{Gauss./RBF K.} & p = 2\end{cases}$$

Composition Rules:

$k = k_1 + k_2,\quad k = k_1 \cdot k_2,\quad c > 0 \Rightarrow k = c\cdot k_1$

f convex; f polynomial or exp $\Rightarrow k = f(k_1)$

$\forall f.k(x,y) = f(x)k_1(x,y)f(y)$

Mercer's Theorem: Valid kernels can be decomposed into a lin. comb. of inner products.

Kern. Ridge Reg.: $\frac{1}{n}\|y - \mathbf{K}\alpha\|_2^2 + \lambda\alpha^\top\mathbf{K}\alpha$

KNN CLASSIFICATION

1. Pick k and distance metric d
2. For given x, find among $x_1, \ldots, x_n \in D$ the k closest to $x \rightarrow x_{i_1}, \ldots, x_{i_k}$
3. Output the majority vote of labels.

NEURAL NETWORKS

$\mathbf{w}$ are the weights and $\varphi : \mathbb{R} \mapsto \mathbb{R}$ is a *nonlinear activation function*: $\phi(\mathbf{x}, \mathbf{w}) = \varphi\left(\mathbf{w}^\top\mathbf{x}\right)$

ReLU: $\max(0, z)$, **Tanh**: $\frac{\exp(z)-\exp(-z)}{\exp(z)+\exp(-z)}$

Sigmoid: $\frac{1}{1+\exp(-z)}$

Rand. init. weights by distr. assumption for φ. ReLu: $2/n_{\text{in}}$; Tanh: $1/n_{\text{in}}$ or $1/(n_{\text{in}} + n_{\text{out}})$

Universal Approximation Theorem:

We can approximate any arbitrary smooth target function, with 1+ layer with sufficient width.

Forward Propagation

Input: $\mathbf{v}^{(0)} = [\mathbf{x}; 1]$ Output: $f = \mathbf{W}^{(L)}\mathbf{v}^{(L-1)}$

Hidden: $\mathbf{z}^{(l)} = \mathbf{W}^{(l)}\mathbf{v}^{(l-1)}, \mathbf{v}^{(l)} = \left[\varphi\left(\mathbf{z}^{(l)}\right); 1\right]$

Backpropagation

Non-convex; Reuse, Compute, Forward Pass

$$\left(\nabla_{\mathbf{W}^{(L)}}\ell\right)^\top = \frac{\partial\ell}{\partial\mathbf{W}^{(L)}} = \frac{\partial\ell}{\partial f}\frac{\partial f}{\partial\mathbf{W}^{(L)}}$$
$$\left(\nabla_{\mathbf{W}^{(L-1)}}\ell\right)^\top = \frac{\partial\ell}{\partial\mathbf{W}^{(L-1)}} = \frac{\partial\ell}{\partial f}\frac{\partial f}{\partial\mathbf{z}^{(L-1)}}\frac{\partial\mathbf{z}^{(L-1)}}{\partial\mathbf{W}^{(L-1)}}$$
$$\left(\nabla_{\mathbf{W}^{(L-2)}}\ell\right)^\top = \frac{\partial\ell}{\partial\mathbf{W}^{(L-2)}} = \frac{\partial\ell}{\partial f}\frac{\partial f}{\partial\mathbf{z}^{(L-1)}}\frac{\partial\mathbf{z}^{(L-1)}}{\partial\mathbf{z}^{(L-2)}}\frac{\partial\mathbf{z}^{(L-2)}}{\partial\mathbf{W}^{(L-2)}}$$

Overfitting Prevention

Regularization: See lasso/ridge regression.
Early Stopping: Stops training upon converg.
Dropout: Deactiv. neurons rand. during train.
Batch Norm.: Norm. layer inputs $\mu = 0, \sigma = 1$.

CNNs $\varphi\left(\mathbf{W} * v^{(l)}\right)$

The output dimension when applying m different $f \times f$ filters to an $n \times n$ image with padding p and stride s is: $l = \frac{n+2p-f}{s} + 1$ For each channel there is a separate filter.

UNSUPERVISED LEARNING

k-Means Clustering

Optimization Goal (non-convex):

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

Lloyd's heuristics: Init. cluster centers $\mu^{(0)}$

- Assign points to closest center
- Update μ_i as mean of assigned points

Converges in exp. time

Init. with k-Means++:

- Rand. data point $\mu_1 = x_i$
- Add $\mu_2, \ldots, \mu_k$ rand., with probability: Given $\mu_{1:j}$, pick $\mu_{j+1} = x_i$ where $p(i) = \frac{1}{z}\min_{l\in\{1,\ldots,j\}} \|x_i - \mu_l\|_2^2$. E.g. further away from any centroid, higher chance.

Converges in expectation $O(\log k) \times$ sol.$_{\text{opt}}$
Find k by negligible loss decrease or reg.

PRINCIPAL COMPONENT ANALYSIS

$$\underset{\substack{\mathbf{W}\in\mathbb{R}^{d\times k}:\mathbf{W}^\top\mathbf{W}=I_k\\ \mathbf{z}_1,\ldots,\mathbf{z}_n\in\mathbb{R}^k}}{\arg\min} \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{W}\mathbf{z}_i\|_2^2,$$

where (v_i are eigenvectors)

$$\mathbf{W}^* = (\mathbf{v}_1|\cdots|\mathbf{v}_k),\quad \mathbf{z}_i^* = \mathbf{W}^{*\top}\mathbf{x}_i.$$

Principal eigenvector ($\mathbf{v}_1$) shows into the direction of greatest variance in the data. The error is the deviation from it. Alternatively:

$$\mathbf{W}^* = \arg\max_{\mathbf{W}^\top\mathbf{W}=\mathbf{I}_k} \mathrm{tr}(\mathbf{W}^\top\Sigma\mathbf{W}),$$

Where $\Sigma = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i\mathbf{x}_i^\top$ is the empirical covariance. Closed form solution given by $\mathbf{w} = \mathbf{v}_1$ for $\lambda_1 \geq \ldots \geq \lambda_d \geq 0 : \Sigma = \sum_{i=1}^{d}\lambda_i\mathbf{v}_i\mathbf{v}_i^\top$.
For $k > 1$, we only take the first k principal eigenvectors, such that $\mathbf{W}^* = [\mathbf{v}_1, \ldots, \mathbf{v}_k]$.
In **SVD** the solution is given by the first k columns of $\mathbf{V}$, with $\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^\top$.

Kernel PCA

Ansatz: $\boldsymbol{w} = \sum_{j=1}^{n}\alpha_j\phi\left(\boldsymbol{x}_j\right) \Rightarrow \|\boldsymbol{w}\|_2^2 = \boldsymbol{\alpha}^\top\boldsymbol{K}\boldsymbol{\alpha}$

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

Closed form, with $\lambda_1 \geq \cdots \geq \lambda_n$:

$$\boldsymbol{\alpha}^* = \frac{1}{\sqrt{\lambda_1}}\boldsymbol{v}_1 \text{ for } K = \sum_{i=1}^{n}\lambda_i\boldsymbol{v}_i\boldsymbol{v}_i^T$$

Autoencoders

Use a NN with smaller hidden layer than input size = output size to find a optimal subspace. $\min_{\hat{\mathbf{x}}}\frac{1}{n}\sum_{i=1}^{n}\|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2$, lin. activ. f. $\Rightarrow$ PCA

MLE & MAP

Maximum Likelihood Estimation (MLE)

(*) if discriminative, $p(a;b)$ since frequentist

$$\widehat{\theta}_{\text{MLE}} := \underset{\theta\in\Theta}{\arg\max}\, p(\mathcal{D};\theta)$$
$$\overset{\text{iid}}{=} \underset{\theta\in\Theta}{\arg\max} \prod_{i=1}^{n} p(\boldsymbol{x}_i, y_i;\theta)$$
$$\overset{*}{=} \underset{\theta\in\Theta}{\arg\min} \sum_{i=1}^{n} -\log p(y_i \mid \boldsymbol{x}_i;\theta),$$

Maximum A Posteriori Estimator (MAP)

(*) if discriminative, $p(a|b)$ since bayesian

$$\widehat{\theta}_{\text{MAP}} := \underset{\theta\in\Theta}{\arg\max}\, p(\theta|\mathcal{D}) = \underset{\theta\in\Theta}{\arg\max}\, p(\mathcal{D}|\theta)p(\theta)$$
$$\overset{\text{iid}}{=} \underset{\theta\in\Theta}{\arg\max}\left(\prod_{i=1}^{n} p(\boldsymbol{x}_i, y_i \mid \theta)\right)\cdot p(\theta)$$
$$\overset{*}{=} \underset{\theta\in\Theta}{\arg\min} \sum_{i=1}^{n} -\log p(y_i \mid \boldsymbol{x}_i, \theta) - \log p(\theta)$$

Regression with MLE/MAP

$$\widehat{\theta}_{\text{MLE}} = \underset{\theta\in\Theta}{\arg\min} \sum_{i=1}^{n}\left(y_i - f\left(\boldsymbol{x}_i;\theta\right)\right)^2$$
$$\widehat{\theta}_{\text{MAP}} = \underset{\theta\in\Theta}{\arg\min} \sum_{i=1}^{n}\left(y_i - f\left(x_i;\theta\right)\right)^2 + \frac{\sigma_\varepsilon^2}{\sigma_\theta^2}\|\theta\|_2^2$$

Regularization can be understood as MAP inference, with different priors (= regularizers) and likelihoods (= loss functions).

Statistical Models for Classification

f minimizing the population risk:

$$f^*(\mathbf{x}) = \operatorname{argmax}_{\hat{y}}\, p(\hat{y} \mid \mathbf{x})$$

This is called the Bayes' optimal predictor for the 0-1 loss. Assuming iid. Bernoulli noise, the conditional probability is:

$$p(\mathbf{y} \mid \mathbf{x}, \mathbf{w}) \sim \text{Ber}\left(y; \sigma\left(\mathbf{w}^\top \mathbf{x}\right)\right)$$

Where $\sigma(z) = \frac{1}{1+\exp(-z)}$ is the sigmoid function. Using MLE we get:

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\text{argmin}} \sum_{i=1}^n \log\left(1 + \exp\left(-y_i \mathbf{w}^\top \mathbf{x}_i\right)\right)$$

Which is the logistic loss. Instead of MLE we can estimate MAP, e.g. with a Gaussian prior:

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\text{argmin}} \lambda \|\mathbf{w}\|_2^2 + \sum_{i=1}^n \log\left(1 + e^{-y_i \mathbf{w}^\top \mathbf{x}_i}\right)$$

BAYESIAN DECISION THEORY

Given $p(\mathbf{y} \mid \mathbf{x})$, a set of actions A and a cost $C : Y \times A \mapsto \mathbb{R}$, pick the action with the maximum expected utility.

$$a^* = \text{argmin}_{a \in A} \mathbb{E}_{\mathbf{y}}[C(\mathbf{y}, a) \mid \mathbf{x}]$$

Useful for asymmetric costs or abstention.

GENERATIVE MODELING (GM)

Aim to estimate $p(\mathbf{x}, \mathbf{y})$ for complex situations using Bayes' rule: $p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x} \mid \mathbf{y}) \cdot p(\mathbf{y})$

Naive Bayes Model

GM for classification tasks. Assuming for a class label, each feature is independent. This helps estimating $p(\mathbf{x} \mid \mathbf{y}) = \prod_{i=1}^d p(x_i \mid y_i)$.

Gaussian Naive Bayes Classifier

Naive Bayes Model with Gaussians features. Estimate the parameters via MLE:

MLE for class prior: $p(\mathbf{y}) = \hat{p}_{\mathbf{y}} = \frac{\text{Count}(\mathbf{Y}=\mathbf{y})}{n}$

MLE for feature distribution:

Where: $p(x_i \mid \mathbf{y}) = \mathcal{N}(x_i; \hat{\mu}_{\mathbf{y},i}, \sigma_{\mathbf{y},i}^2)$

$$\mu_{\mathbf{y},i} = \frac{1}{\text{Count}(\mathbf{Y}=\mathbf{y})} \sum_{j \mid \mathbf{y}_j = \mathbf{y}} x_{j,i}$$

$$\sigma_{\mathbf{y},i}^2 = \frac{1}{\text{Count}(\mathbf{Y}=\mathbf{y})} \sum_{j \mid \mathbf{y}_j = \mathbf{y}} \left(x_{j,i} - \hat{\mu}_{\mathbf{y},i}\right)^2$$

Predictions are made by:

$$\underset{\hat{\mathbf{y}}}{\text{argmax}} p(\hat{\mathbf{y}} \mid \mathbf{x}) = \underset{\hat{\mathbf{y}}}{\text{argmax}} p(\hat{\mathbf{y}}) \cdot \prod_{i=1}^d p(x_i \mid \hat{\mathbf{y}})$$

Equivalent to decision rule for bin. class.:

$$\mathbf{y} = \text{sgn}\left(\log \frac{p(\mathbf{Y} = +1 \mid \mathbf{x})}{p(\mathbf{Y} = -1 \mid \mathbf{x})}\right)$$

Where $f(\mathbf{x})$ is called the discriminant function. If the conditional independence assumption is violated, the classifier can be overconfident.

Gaussian Bayes Classifier

No independence assumption, model the features with a multivar. Gaussian $\mathcal{N}(\mathbf{x}; \mu_{\mathbf{y}}, \Sigma_{\mathbf{y}})$:

$$\mu_{\mathbf{y}} = \frac{1}{\text{Count}(\mathbf{Y}=\mathbf{y})} \sum_{j \mid \mathbf{y}_j = \mathbf{y}} \mathbf{x}_j$$

$$\Sigma_{\mathbf{y}} = \frac{1}{\text{Count}(\mathbf{Y}=\mathbf{y})} \sum_{j \mid \mathbf{y}_j = \mathbf{y}} \left(\mathbf{x}_j - \hat{\mu}_{\mathbf{y}}\right)\left(\mathbf{x}_j - \hat{\mu}_{\mathbf{y}}\right)^\top$$

This is also called the *quadratic discriminant analysis (QDA)*. LDA: $\Sigma_+ = \Sigma_-$, Fisher LDA: $p(\mathbf{y}) = \frac{1}{2}$, Outlier detection: $p(\mathbf{x}) \leq \tau$.

Avoiding Overfitting

MLE is prone to overfitting. Avoid this by restricting model class (fewer parameters, e.g. GNB) or using priors (restrict param. values).

Discriminative models: $p(\mathbf{y} \mid \mathbf{x})$, fewer assumptions about data distribution

Generative models: $p(\mathbf{x}, \mathbf{y})$, can detect outliers, gen. missing data, less robust to outliers.

GAUSSIAN MIXTURE MODEL

Data is generated from a mixture of Gaussians:

$$p(\mathbf{x} \mid \theta) = \sum_{j=1}^k w_j \mathcal{N}(\mathbf{x}; \mu_j, \Sigma_j)$$

Estimate parameters by minimizing:

$$\text{argmin}_\theta - \sum_{i=1}^n \log \sum_{j=1}^k w_j \mathcal{N}(\mathbf{x}_i \mid \mu_j, \Sigma_j)$$

Non-convex objective, iteratively update parameters by predicting labels and imputing missing data.

Hard-EM Algorithm

E-Step: predict the most likely class for each data point:

$$\begin{aligned} z_i^{(t)} &= \text{argmax}_z\, p(z \mid \mathbf{x}_i, \boldsymbol{\theta}^{(t-1)}) \\ &= \text{argmax}_z\, p(z \mid \boldsymbol{\theta}^{(t-1)}) \cdot p(\mathbf{x}_i \mid z, \boldsymbol{\theta}^{(t-1)}) \end{aligned}$$

M-Step: compute MLE of $\theta^{(t)}$ as for GBC.

Problems: Labels even if uncertain, tries to extract too much inf. Works poorly if clusters are overlapping. *Equivalent to k-Means with Lloyd's heuristics:* When having uniform weights and spherical covariances.

Soft-EM Algorithm

E-Step: Calculate the cluster membership weights for each point ($w_j = \pi_j = p(Z = j)$):

$$\gamma_j^{(t)}(\mathbf{x}_i) = p(Z = j \mid \mathbf{D}) = \frac{w_j \cdot p\left(\mathbf{x}_i; \theta_j^{(t-1)}\right)}{\sum_k w_k \cdot p\left(\mathbf{x}_i; \theta_k^{(t-1)}\right)}$$

M-Step: compute MLE with closed form:

$$w_j^{(t)} = \frac{1}{n} \sum_{i=1}^n \gamma_j^{(t)}(\mathbf{x}_i) \qquad \boldsymbol{\mu}_j^{(t)} = \frac{\sum_{i=1}^n \mathbf{x}_i \cdot \gamma_j^{(t)}(\mathbf{x}_i)}{\sum_{i=1}^n \gamma_j^{(t)}(\mathbf{x}_i)}$$

$$\boldsymbol{\Sigma}_j^{(t)} = \frac{\sum_{i=1}^n \gamma_j^{(t)}(\mathbf{x}_i)\left(\mathbf{x}_i - \boldsymbol{\mu}_j^{(t)}\right)\left(\mathbf{x}_i - \boldsymbol{\mu}_j^{(t)}\right)^\top}{\sum_{i=1}^n \gamma_j^{(t)}(\mathbf{x}_i)}$$

Init. the weights as uniformly distributed, rand. or with k-Means++ and for variances use spherical init. or empirical covariance of the data. Select k using cross-validation.

Special Cases of Gaussian Mixtures

- **Spherical**: Same variance in all directions. $\Sigma_k = \sigma_k^2 I$; Parameters: K
- **Diagonal**: Different variance for each dimension but no covariance. $\Sigma_k = \text{Diag}(\sigma_{k_1}^2, \sigma_{k_2}^2, \ldots, \sigma_{k_d}^2)$; Par.: $K \cdot d$
- **Tied**: Same cov. matrix for all components. $\Sigma_1 = \Sigma_2 = \cdots = \Sigma_k$; Parameters: $\frac{d(d+1)}{2}$
- **Full**: Free covariance in all dimensions. ω, μ, Σ; Parameters: $\frac{d(d+1)}{2} K$

Degeneracy of GMMs

GMMs can overfit with limited data. To prevent this, add $v^2 I$ to the covariance matrices, preventing collapse (equivalent to using a Wishart prior). Choose v via cross-validation.

Gaussian-Mixture Bayes Classifiers

Assume that $p(x \mid y)$ for each class can be modelled by a GMM.

$$p(\mathbf{x} \mid y) = \sum_{j=1}^{k_y} w_j^{(y)} \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_j^{(y)}, \boldsymbol{\Sigma}_j^{(y)})$$

Giving highly complex decision boundaries:

$$p(y \mid \mathbf{x}) = \frac{1}{z} p(y) \sum_{j=1}^{k_y} w_j^{(y)} \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_j^{(y)}, \boldsymbol{\Sigma}_j^{(y)}).$$

GMMs for Density Estimation

Can be used for anomaly detection or data imputation. Detect outliers, by comparing the estimated density against τ. Allows to control the FP rate. Use ROC curve as evaluation criterion and optimize using CV to find τ.

General EM Algorithm

E-Step: Take the expected value over latent variables z to generate likelihood function Q:

$$\begin{aligned} Q(\theta; \theta^{(t-1)}) &= \mathbb{E}_Z[\log p(\mathbf{X}, \mathbf{Z} \mid \theta) \mid \mathbf{X}, \theta^{(t-1)}] \\ &= \sum_{i=1}^n \sum_{z_i=1}^k \gamma_{z_i}(\mathbf{x}_i) \log p(\mathbf{x}_i, z_i \mid \theta) \end{aligned}$$

with $\gamma_z(\mathbf{x}) = p\left(z \mid \mathbf{x}, \theta^{(t-1)}\right)$.

M-Step: Compute MLE / Maximize:

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

We have monotonic convergence, each EM-iteration increases the data likelihood.

GAN

New loss: $\underset{w_G}{\min} \underset{w_D}{\max} \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}} [\log D(\mathbf{x}, w_D)]$
$+ \mathbb{E}_{\mathbf{z} \sim p_z} [\log(1 - D(G(\mathbf{z}, w_G), w_D))]$

- *Saddle Point:* Training seeks a saddle point.
- *Capacity:* Conv. if **G** and **D** have enough capacity.
- *Optimal* **D** *for fixed* **G**:
$$D_G(\mathbf{x}) = \frac{p_{\text{data}}(\mathbf{x})}{p_{\text{data}}(\mathbf{x}) + p_G(\mathbf{x})}$$
- *Fake Probability:* $1 - D_G$.
- *Issues:* Oscill., divergence, mode collapse.

Performance Metric:

$$DG = \max_{w_D'} M(w_G, w_D') - \min_{w_G'} M(w_G', w_D)$$

where $M(w_G, w_D)$ is the training objective. "'

VARIOUS

Derivatives:

$$\nabla_\mathbf{x} \mathbf{x}^\top \mathbf{A} = \mathbf{A} \quad \nabla_\mathbf{x} \mathbf{a}^\top \mathbf{x} = \nabla_\mathbf{x} \mathbf{x}^\top \mathbf{a} = \mathbf{a}$$

$\nabla_\mathbf{x} \mathbf{b}^\top \mathbf{A} \mathbf{x} = \mathbf{A}^\top \mathbf{b}$, $\nabla_\mathbf{x} \mathbf{x}^\top \mathbf{x} = 2\mathbf{x}$, $\nabla_\mathbf{x} \mathbf{x}^\top \mathbf{A} \mathbf{x} = 2\mathbf{A}\mathbf{x}$

Square Loss: $\nabla_w \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 = 2\mathbf{X}^\top \mathbf{X} \mathbf{w} - 2\mathbf{X}^\top \mathbf{y}$

Bayes Theorem:

$$p(y \mid x) = \frac{1}{p(x)} p(y) \cdot p(x \mid y)$$

Normal Distribution:

$$\mathcal{N}(x; \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^d \det(\Sigma)}} \exp\left(-\frac{(\mathbf{x}-\mu)^\top \Sigma^{-1} (\mathbf{x}-\mu)}{2}\right)$$

Exponential Distribution: $\text{Exp}(\lambda) = \lambda e^{-\lambda x}$

Other Facts

Memoryless: $p(X > a + b \mid X \geq a) = p(X > b)$

Tower Property: $\mathbb{E}[X] = \mathbb{E}[\mathbb{E}[X \mid Y]]$

$\Rightarrow \mathbb{E}[X] = \sum_{y \in \mathcal{Y}} \mathbb{E}[X \mid Y = y] p_Y(y)$

$\text{Tr}(AB) = \text{Tr}(BA)$, $\text{Var}(X) = \mathbb{E}\left[X^2\right] - \mathbb{E}[X]^2$,

$\mathbf{X} \in \mathbb{R}^{n \times d} : \mathbf{X}^{-1} \to \mathcal{O}\left(d^3\right); \mathbf{X}^\top \mathbf{X} \to \mathcal{O}\left(nd^2\right)$,

$\binom{n}{k} = \frac{n!}{(n-k)!k!}$, $\left\|\mathbf{w}^\top \mathbf{w}\right\|_2 = \sqrt{\mathbf{w}^\top \mathbf{w}}$

$$\text{Cov}[\mathbf{X}] = \mathbb{E}\left[(\mathbf{X} - \mathbb{E}[\mathbf{X}])(\mathbf{X} - \mathbb{E}[\mathbf{X}])^\top\right]$$

$$p(\mathbf{z} \mid \mathbf{x}, \theta) = \frac{p(\mathbf{x}, \mathbf{z} \mid \theta)}{p(\mathbf{x} \mid \theta)}$$

Convexity

- $\alpha f + \beta g$, $\alpha, \beta \geq 0$, convex if f, g convex
- $f \circ g$, convex if [f convex and g affine (e.g. $ax + b$)] or [f non-decresing and g convex]
- $\max(f, g)$, convex if f, g convex
- $L(\lambda \mathbf{w} + (1 - \lambda)\mathbf{v}) \leq \lambda L(\mathbf{w}) + (1 - \lambda) L(\mathbf{v})$

1. Order: $L(\mathbf{w}) + \nabla L(\mathbf{w})^\top (\mathbf{v} - \mathbf{w}) \leq L(\mathbf{v})$
2. Order: Hessian $\nabla^2 L(\mathbf{w}) \geq 0$ (psd)