General Classification Learning an indicator function

Classes: $1, \dots, k, \mathcal{D}, \mathcal{O}$ (doubts and outliers) Bayes: Both feature vector X and class Y are random vars Two different problems (notation: $p_y(x)$ = probability that $X = x$ looking at class y = "class conditional density" = "likelihood"):

Bayes	prior	P(model)	π_y
	likelihood	P(data|model)	$p_y(x) = \prod_i p(x_i\|y)$ if i.i.d.
	posterior	P(model|data)	$p(y\|x)$
	evidence	P(data)	$\sum_z \pi_z p_z(x)$

posterior = likelihood · prior/evidence

For known densities If 0-1-loss: Optimal classifier:

- Select class y as the $\arg\max\ p(y|x)$ (the class that has the maximum posterior) if the posterior is $> 1 - d$
- otherwise select doubt class $\mathcal{D}$

If any other loss function: Sum up the losses and compare with d.

Parametric models Assume that classes can be treated independently (θ_y is not informative for class $\mathcal{X}_z$)

Maximum Likelihood Estimation Maximize the likelihood given the data: For each class $y : \hat{\theta}_y = \arg\max_{\theta y} P(\mathcal{X}_y|\theta_y)$ How? find the extremum of the log likelihood function. Log turns product $P = \prod_i p(x_{iy}|\theta_y)$ into sum, then derive and set to zero.

- Consistency: For $n \to \infty$, MLE converges to the best model θ_0.
- Equivariance: if $\hat{\theta}$ MLE of θ then $f(\hat{\theta})$ MLE of $f(\theta)$
- Asymptotic normality: $\sqrt{n}(\hat{\theta} - \theta)$ converges to a normal distr. in probability
- Asymptotic efficiency: For well-behaved estimators, MLE has the smallest variance for large n

Rao Cramer inequality Lower bound on variance for unbiased estimators of θ: Inverse Fisher information.

Bayes estimation θ is a random variable, not a point. Can be applied recursively: For the likelihood $p(\mathcal{X}^n|\theta) = p(x_n|\theta)p(X^{n-1}|\theta) \to$ Calculate result without x_n, then multiply with next likelihood and scale to 1 by dividing by the integral to get the posterior, at the base case of this is the prior $p(\theta)$.

For reasonable priors, MLE/Bayes are equal in the limit.

Error estimation

Epsilon-Delta for classifier quality $P_{X,Y}\{\mathcal{R}(\hat{c}_n) \leq \mathcal{R}(c^{Bayes}) + \epsilon\} > 1 - \delta$. The trained classifier is ϵ-close to the optimal error (Bayes) more than $(1 - \delta) \cdot 100$ percent of the time (PAC, Probably Approximately Correct). **ToDo: Lecture 9, strong vs. weak learning**

Cross-Validation Estimate error as $\frac{1}{K}\sum_{v \in K} \hat{\mathcal{R}}_v$ (average error rate per bucket).Usually $K = min(\sqrt{n}10)$ is chosen.

Problem: Underfitting because we don't train with entire data.

Leave one out $K = n$, bucketsize 1. Predicts true error without bias but can have a large variance (training sets are very similar).

Bootstrap Estimate mean and variance of error as usual by using e.g. normal sample set as "ground truth".

Problem: Too optimistic. **TODO: Probabilities 0.632 etc**

Jackknife Tries to estimate the bias of an estimator $\hat{S_n}$ (so you can then subtract it). Recompute the statistic estimate n times using leave-one-out. Calculate estimated bias as $(n-1)(\tilde{S_n} - \hat{S_n})$ with $\tilde{S_n} = \frac{1}{n}\sum_{i=1}^n \hat{S}_{n-1}^{(-i)}$.

Also with bootstrapping. Estimate $\hat{bias} = \frac{1}{|B|}\sum_{b \in B}\hat{S}(b) - \hat{S}$.

Linear Discriminant Functions

Perceptron Separate two classes ($y_i \in \{-1, 1\}$) by a hyperplane (homogeneous coordinates). w is the normal, $w^T x$ is the distance. Idea: Take the loss function $Q = \sum_M w^T x_i y_i$ for all misclassified points and minimize it. Use $\frac{\partial Q}{\partial w} = \sum_M x_i y_i$ and descend towards the negative gradient.

This yields the update rule $w_{t+1} = w_t + \eta \sum_M x_i y_i$ for some not too large η. We can also sum individual points ($w_{t+1} = w_t + \eta x_t y_t$ for misclassifieds (x_t, y_t) while there still are misclassifieds).

Fisher's Linear Discriminant Analysis (LDA) Idea: Project k classes to $k-1$ dimensions s.t. they are optimally separated (maximize inter-class, minimize intra-class scatter).

- Define within-scatter of class α as $\Sigma_\alpha = \Sigma_{x \in \alpha}(x - m_\alpha)(x - m_\alpha)^T$, within-scatter of all classes $\Sigma_W = \Sigma_1 + \Sigma_2 + \dots$, projected scatter by w is $w^T \Sigma_W w$.
- Define between-class scatter $\frac{1}{k}\sum_\alpha (m_\alpha - m)(m_\alpha - m)^T$, m is global mean
- Minimize $J(W) = \frac{|W^T \Sigma_B W|}{|W^T \Sigma_W W|}$ by the generalized EV problem $\Sigma_b w_i = \lambda_i \Sigma - W w_i$.
- We now have as the vectors in W the $(k-1)$ discriminant functions for $y_i = w_i^T x$.

2 class case unscaled: $\hat{w} = (\Sigma_1 + \Sigma_2)^{-1}(m_1 - m_2)$.

Support Vector Machines (SVM)

Primal problem Maximize m under constraints $z_i(\mathbf{w}^T \mathbf{y}_i + w_0) \geq m$, $z_i \in \{-1, 1\}$. Rescale $\mathbf{w} \to \frac{\mathbf{w}}{m}$ to get constraints $z_i(\mathbf{w}^T y + w_0) \geq 1$. With $m = \frac{1}{||\mathbf{w}||}$, we can as well minimize $\frac{1}{2}\mathbf{w}^T\mathbf{w}$. Therefore: minimize $\frac{1}{2}\mathbf{w}^T\mathbf{w}$ s.t. $z_i(\mathbf{w}^T \mathbf{y}_i + w_0) \geq 1$.

Lagrange formulation $L(w, w_0, \alpha) = \frac{1}{2} - \sum_{i=1}^n \alpha_i(y_i(w^T x_i + w_0) - 1)$ Derive wrt. w and w_0, set to 0, solve.

Dual problem Obtained from the primal form by substituting $\frac{\partial L}{\partial w} = \frac{\partial L}{\partial w_0} = 0$. Maximize $W(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i=1}^n \sum_{j=1}^n z_i z_j \alpha_i \alpha_j \mathbf{y_i}^T \mathbf{y_j}$ subject to $\alpha_i \geq 0 \wedge \sum_{i=1}^n z_i \alpha_i = 0$. The optimal hyperplane $\mathbf{w}^*, w_0^*$ is given by $w^* = \sum_{i=1}^n \alpha_i^* z_i \mathbf{y_i}$, $w_0^* = -\frac{1}{2}(\min_{i:z_i=1} \mathbf{w^*}^T \mathbf{y_i} + \max_{i:z_i=-1} \mathbf{w^*}^T \mathbf{y_i})$

Sparsity Optimal Margin: $w^T w = \sum_{i \in SV} \alpha_i^*$.

Soft-Margin SVM Introduce slack variables ξ_i. New problem: Minimize $\frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^n \xi_i$. New constraint: $\xi_i \geq 0$.

For the dual problem, the only difference turns out that we have $\forall i\ C \geq \alpha_i \geq 0$ instead of $\forall i\ \alpha_i \geq 0$.

nonlinear SVM Substitute $\mathbf{y_i}^T \mathbf{y_j} = \phi^T(\mathbf{x_i})\phi(\mathbf{x_j})$ by a kernel function $K(\mathbf{x_i}, \mathbf{x_j})$, such that the discriminant function is $g(\mathbf{x}) = \sum_{i=1}^n \alpha_i z_i K(\mathbf{x_i}, \mathbf{x})$.

K is a kernel function iff the matrix $K(i,j) = K(\mathbf{x_i}, \mathbf{x_j})$ is positive semi-definite. Addition, scaling, multiplication, plugging input into functions, applying polynomial/exp functions on the kernel results in a new kernel.

Ensemble Methods

Use weak learners (stumps, decision trees, multi-layer perceptrons, RBFs) c_b to build a weighted classifier $\hat{c}_B(x) = \text{sgn}(\sum_{b=1}^B \alpha_b c_b(x))$.

Bagging Train classifiers by different bootstrap samples. Random forests: bagging + decision trees (pick the best split-point among p variables again and again). Weights are chosen uniformly.

Boosting Uses data-reweighting. AdaBoost minimizes the exponential loss $e^{-yF(x)}$. Apply weight $w_i^{(b)}$ to training data (x_i, y_i) at the bth boosting step. $w_i^{(1)} = \frac{1}{n}$, then $\epsilon_b = \sum_{i=1}^n \frac{w_i^{(b)}}{\sum_{i=1}^n w_i^{(b)}} \mathbb{I}_{\{c_b(x_i) \neq y_i\}}, \alpha_b = \log\frac{1-\epsilon_b}{\epsilon_b}, w_i^{(b+1)} = w_i^{(b)} \exp(\alpha_b \mathbb{I}_{\{c_b(x_i) \neq y_i\}})$

Regression

Least-Squares Minimize $RSS(\beta) = ||\mathbf{y} - \mathbf{X}\beta||^2$, $\mathbf{X}$ are the points to fit row-wise. Differentiate wrt. β and set to 0: For nonsingular $\mathbf{X}^T\mathbf{X}$: $\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$ and $\hat{\mathbf{y}} = \mathbf{X}\hat{\beta}$.

If we assume additive gaussian noise ϵ around 0 with variance σ^2, this is also the MLE optimum: $\hat{\beta} \sim \mathcal{N}(\beta, (\mathbf{X}^T\mathbf{X})^{-1}\sigma^2)$. Has the smallest variance among all linear unbiased estimates (Gauss Markov Theorem var$(a^T\beta) \leq$ var$(c^T y)$).

Ridge $RSS(\beta) = ||\mathbf{y} - \mathbf{X}\beta||^2 + \lambda||\beta||_2^2$. Solution: $\hat{\beta}^{\text{ridge}} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}$. Small eigenvalues are repressed: Shrinkage factor $\frac{d_j^2}{d_j^2 + \lambda}$ is small for small singular values.

LASSO Favors sparseness, $RSS(\beta) = ||\mathbf{y} - \mathbf{X}\beta||^2 + \lambda||\beta||$.

Bias-Variance Dilemma Split the error into three components: $\mathbb{E}_D\mathbb{E}_{X,Y}(\hat{f}(X)-Y)^2 = \mathbb{E}_D\mathbb{E}_X(\hat{f}(X)-\mathbb{E}(Y|X))^2 + \mathbb{E}_{X,Y}(Y-\mathbb{E}(Y|X))^2 = \mathbb{E}_X\mathbb{E}_D(\hat{f}(X)-\mathbb{E}_D\hat{f}(X))^2 + \mathbb{E}_X(\mathbb{E}_D\hat{f}(X)-\mathbb{E}(Y|X))^2 + \mathbb{E}_{X,Y}(Y-\mathbb{E}(Y|X))^2 = \text{variance} + \text{bias}^2 + \text{noise}$
Small dataset, large hyp class: Large var, small bias - Large dataset, small hyp class: Small var, large bias. Ensemble methods keep bias fixed and lower variance, but RCLB is a lower-bound of variance.

Combining B regressors Use $\hat{f}(x) = \frac{1}{B}\sum_{i=1}^B \hat{f}_i(x)$

Bias $\mathbb{E}_D\hat{f}(x) - \mathbb{E}(Y|x) = \frac{1}{B}\sum_{i=1}^B \mathbb{E}_D\hat{f}_i(x) - \mathbb{E}(Y|x) = \frac{1}{B}\sum_{i=1}^B \text{bias}(\hat{f}_i(x))$ →Unbiased estimators remain unbiased

Variance
$\mathbb{V}(\hat{f}(x)) = \frac{1}{B^2}\sum_{i=1}^B \mathbb{V}(\hat{f}_i(x)) + \frac{1}{B^2}\sum\sum_{i\neq j}\text{Cov}(\hat{f}_i(x),\hat{f}_j(x))$
If we assume small covariances and similar variances ($\mathbb{V}(\hat{f}_i(x)) \approx \sigma^2$), then $\mathbb{V}(\hat{f}(x)) \approx \frac{\sigma^2}{B}$ (reduction by factor of B)

Gaussian processes Extend lin. regression by defining a prior over the regression coefficients. "kernelized lin. regression"
Assume $y = x^T\beta + \epsilon$, ϵ normally distr. around 0, calculate expectation and covariance, get "confidence bands" around function by covariance

Unsupervised Learning

Nonparametric Density Estimation Always induce a classifier by selecting the class with the highest density at a point.

Historgrams Problem: Reliable estimate needs exponentially many samples with the dimension.

Parzen Estimators Some window function ϕ with d-dimensional volume $V_n = h_n^d$. Estimate the density $\hat{p}_n(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^n \frac{1}{V_n}\phi(\frac{\mathbf{x}-\mathbf{x_i}}{h_n})$. Convergence: Variance goes to 0, converges to real density if $\lim_{n\to\infty} V_n = 0$ and $\lim_{n\to\infty} nV_n = \infty$. Convolution of empirical density with the window function $\frac{1}{V_n}\phi(\mathbf{x})$ (low-pass if ϕ Gaussian). Problems: V_0 too small means noise (not smooth enough), V_0 too large means loss of detail, V_n is data independent (doesn't scale differently to regions)

k-NN V_n grows until k samples included: $\hat{p}(x) = \frac{1}{V_k(x)}$ where $V_k(x)$ =minimal volume around x with k neighbors. Induced classifier corresponds to majority vote. Error rate converges in L_1 to $P_1 \leq P^*(2-\frac{C}{C-1}P^*) \leq 2P^*$ with P^* Bayes error rate.
Problems: Complexity increases with dimension (compute norms) and sample size

Clustering

k-Means Assignment function $c(\mathbf{x})$, centroids $\mu_c \in \mathcal{Y}$, find c and $\mathcal{Y}$ that minimize $\sum_{\mathbf{x}}||\mathbf{x}-\mu_{c(\mathbf{x})}||^2$. Approximation:

1. keep $\mathcal{Y}$ fixed, $c(x) := \text{argmin}_{c\in\{1,\dots,k\}}||\mathbf{x}-\mu_c||^2$
2. keep $c(\mathbf{x})$ fixed, $\mu_\alpha = \frac{1}{n_\alpha}\sum_{\mathbf{x}:c(\mathbf{x})=\alpha}\mathbf{x}$

Mixture Models Assume data distributed according to not one density $p(\mathbf{x}|\theta)$, but a mixture of densities $\sum_{c\in\{1..k\}}\pi_c\, p(\mathbf{x}|\theta_c)$.
Task: Estimate $\hat{\theta}$ such that it maximizes the likelihood of $\mathcal{X}$: $\prod_{\mathbf{x}\in\mathcal{X}}\sum_{c\in\{1..k\}}\pi_c\, p(\mathbf{x}|\theta_c)$. Log-likelihood: $\sum_{\mathbf{x}\in\mathcal{X}}\log\sum_{c\in\{1..k\}}\pi_c\, p(\mathbf{x}|\theta_c)$. Optimizing the sum within log is hard.

Gaussian Mixtures
$p(\mathbf{x}|\mu,\mathbf{\Sigma}) = \frac{1}{\sqrt{2\pi}^d\sqrt{|\Sigma|}}\exp(-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)))$

EM Introduce "latent variables" $\mathcal{X}_{\mathcal{L}}$: Is x assigned to class c?

1. Expectation: Calculate $Q(\theta,\theta^{(j)}) = \mathbb{E}_{\mathcal{X}_L}[L(\mathcal{X},\mathcal{X}_L|\theta)|\mathcal{X},\theta^{(j)}]$
2. Maximization: Choose $\theta^{(j+1)} = \text{argmax}_{\theta}Q(\theta,\theta^{(j)})$

EM for Mixture Models Estimation: We call the latent variables $M_{\mathbf{x}c}$ and insert the log-likelihood for $L(\mathcal{X},M|\theta)$. Then we pull the expectation through to the $M_{\mathbf{x}c}$ and call it $\gamma_{\mathbf{x}c} = \mathbb{E}_M[M_{\mathbf{x}c}|\mathcal{X},\theta^{(j)}]$. Case distinction on M (either 0 or 1) yields that $\gamma_{\mathbf{x}c} = P(c|\mathbf{x},\theta^{(j)})$. Use Bayes to get $\frac{P(\mathbf{x}|c,\theta^{(j)})P(c|\theta^{(j)})}{P(\mathbf{x}|\theta^{(j)})}$.
Maximization: Derive $Q-\lambda(\sum_{\bar{c}=1}^k\pi_c - 1) = 0$, for Gaussians w.r.t to mixture weights π_c and μ_c and Σ_c. We get $\pi_c = \frac{1}{|\mathcal{X}|}\sum_{x\in\mathcal{X}}\gamma_{\mathbf{x}c}$, $\mu_c = \frac{\sum_{\mathbf{x}\in\mathcal{X}}\gamma_{\mathbf{x}c}\mathbf{x}}{\sum_{\mathbf{x}\in\mathcal{X}}\gamma_{\mathbf{x}c}}$, $\Sigma_c = \sigma_c^2\mathbb{I}$ with $\sigma_c^2 = \frac{\sum_{\mathbf{x}\in\mathcal{X}}\gamma_{\mathbf{x}c}(\mathbf{x}-\mu_c)^2}{\sum_{\mathbf{x}\in\mathcal{X}}\gamma_{\mathbf{x}c}}$.
Repeat estimation then maximization until convergence.

Convergence Log-likelihood increases with each iteration.

Problems Hard to analyze (cost fn changes), influence of hidden variables not completely understood, local minima (usually not practical problem, choose smaller step), dependent on initial values

Time Series Sequence of random variables $(X_t)_{t\in\mathcal{T}} = X_1, X_2, \dots$. Drawing from these, ⇒ trajectory (assign an x_t to each t). Stationary process: Joint distribution of any subsequence is invariant under index shifts. Markov process: Outcome of each draw depends only on the previous draw. Stationary markov chains have single transition matrix (no changes).

HMM Each state can generate different symbols b of random variable S, each with a certain probability in state k: $e_k(s_t) = P(S_t = s_t|X_t = x_k)$. The probability of drawing the path $\mathbf{x}$ and generating the string $\mathbf{s}$ is $P(\mathbf{s},\mathbf{x}) = a_{x_0x_1}\cdot\prod_{t=1}^n e_{x_t}(s_t)a_{x_t}x_t+1$.

Evaluation problem Known transition+emission probabilities $a_{ij}, e_k(s_t)$, given sequence. Compute probability that a sequence $\mathbf{s}$ was emitted. Forward algorithm: Recursion: $f_l(s_t)$ total probability of subsequence $s_1,\dots,s_t$ if the t-th state is x_l. $f_l(s_{t+1}) = e_l(s_{t+1})\sum_k f_k(s_t)a_{k_l}$. Then compute $P(\mathbf{s}) = \sum_k f_k(s_n)a_{k\epsilon}$, ϵ being the final state. The backward algorithm does the same by computing probabilities for suffix strings instead of prefix strings.

Decoding problem Known $a_{ij}, e_k(s_t)$. compute most likely path $x_1,\dots,x_n$ responsible for a sequence $\mathbf{s}$. Viterbi algorithm. We know that $P(x_l,t|\mathbf{s}) = \frac{f_l(s_t)b_l(s_t)}{P(\mathbf{s})}$. For each symbol s_t, for all states x_i alculate $v_l(s_{t+1}) = e_l(s_{t+1})\max_k(v_k(s_t)a_{kl})$ as the probability of reaching state x_l in step $t+1$. Set a pointer to the most probable predecessor state. Find the most probable path by following the pointers backward.

Learning problem Known sequences $\mathbf{s}^1,\dots,\mathbf{s}^n$, compute the model ($a_{ij}$ and $e_k(s_t)$) and the path $\mathbf{x}$. Baum-Welch algorithm. E-Step: For each sequence $\mathbf{s}^j$ Compute the f, b by the backward-algorithm. Then compute A (expected number of times transition $x_k \mapsto x_l$ is made) and E (expected number of times b is emitted by x_k) for all states and symbols b. M-Step: Compute parameter estimates a_{kl} and $e_k(b)$.

Maths

- $P(X,Y) = \frac{P(X,Y)P(Y)}{P(Y)} = P(X|Y)P(Y)$
- $P(X) = \sum_{y\in\mathcal{Y}}P(x,y);\;\; p(x) = \int_{\mathcal{Y}}p(x,y)\,dy$
- independence: $P(Y|X) = P(Y)$, $P(X,Y) = P(X)P(Y)$
- $E[X] = \int_{\mathcal{X}}x\cdot p(x)\,dx$, $E[x|y] = \int xp(x|y)\,dx$
- $E[f(x)] = \int_{\mathcal{X}}f(x)p(x)\,dx$
- $E[a+bX] = a+bE[X]$, $Var[a+bX] = b^2Var[X]$
- $Var[X] = \int(x-\mu_X)^2\cdot p(x)\,dx = \sum_{i=1}(x_i-\mu_X)^2\cdot p(x_i) \geq 0$
- $Var[X] = E[(X-E[X])^2] = E[X^2]-E[X]^2$
- $Var[X+Y] = Var[X]+Var[Y]+2Cov(X,Y)$
- $Cov(X,Y) = E_{X,Y}[(X-\mu_x)(Y-\mu_y)]$
- $Cov(X,Y) = E[XY]-E[X]E[Y] = 0$ if independent
- $Corr(X,Y) = Cov(X,Y)/\sigma_X\sigma_Y$
- $\Sigma = E[(x-\mu)(x-\mu)^T] = \int(x-\mu)(x-\mu)^Tp(x)\,dx$
- $\frac{d}{dx}\log(ax+b) = \frac{a}{ax+b}$

Lagrange multipliers How to minimize $f(\mathbf{w})$ s.t. $g_i(\mathbf{w}) \leq 0$ and $h_j(\mathbf{w}) = 0$ for all i, j.
Primal form: Minimize $L(\mathbf{w},\alpha,\beta) = f(\mathbf{w}) + \sum_i\alpha_ig_i(\mathbf{w}) + \sum_j\beta_jh_j(\mathbf{w})$.
Dual form: The value of the dual form is ≤ the value of the primal form, for SVMs it is =.

Kuhn-Tucker conditions Necessary and sufficient conditions for a point $\mathbf{w}^*$ to be an optimum: Optimum if there exist α^* and β^* s.t. $\frac{\partial L(\mathbf{w}^*,\alpha^*,\beta^*)}{\partial\mathbf{w}} = 0$ and $\frac{\partial L(\mathbf{w}^*,\alpha^*,\beta^*)}{\partial\beta} = 0$, and for all i: $\alpha_ig_i(\mathbf{w}^*) = 0$ and $g_i(w^*) \leq 0$ and $\alpha_i^* \geq 0$