Review

Expected cost, P(z=k|x), LDA & QDA
Generative versus discriminative
Generative

- ML or Bayesian
- Density-based
- PNN, kNN

Discriminative

- Logistic (binary) or softmax (multi-class), IRLS
- Perceptrons
- Decision Trees

Decision Trees:

- Tests: single attribute tests; hyperplane tests
- Continuous tests: Ranges
- Select t such that mutual information IG(z,t) is maximum

$$IG(z,t) = H(z) - H(z\,|\,t) = H(t) - H(t\,|\,z)$$

- You could consider pairs of tests at a time (JMI): IG(z,t_i, t_j). There is nothing prevents you form considering triples, quadruples, etc., but computing mutual information becomes complex!
- Other criteria: Gini index, MAP error,…
- Decision Trees have Low Bias and High Variance
- Ways to address it: Cost-complexity pruning and Error pruning, Bagging and Random Forests, Boosting: AdaBoost and Gradient Boosting … do 10-fold cross-validation
- Bagging: Construct lots of decision trees using Bootstrap samples and average the classification decisions/regression estimates. Does not work well.
- Random Forests: Similar to bagging, but now sample features as well. Works well in practice.
- Boosting: Weak learners in series. Downstream learners use data on which upstream learners made errors for training. You can do this using weighted bootstrap sampling…. AdaBoost
- Idea: $\hat{z}_M^n = \mathrm{sgn}[\sum_{m=1}^{M} \alpha_m f_m(\underline{x}^n)]; M \approx 10-20; n = 1,2,..,N;\ f\ \textit{is a weak classifier}$

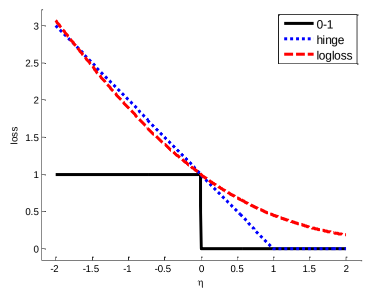
  In a binary classifier, it makes correct decisions at least 50% or more.
- AdaBoost has a nice way of computing the weights $\{\alpha_m\}$
- We need some preliminaries. Suppose at step *m* have

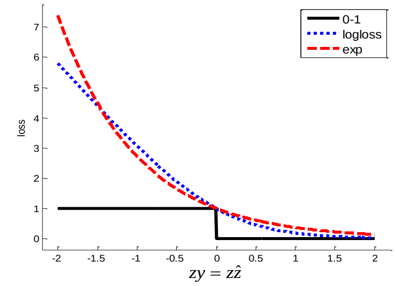
$$\hat{z}_{m-1}^n = \mathrm{sgn}[\sum_{j=1}^{m-1} \alpha_j f_j(\underline{x}^n)]; M \approx 10-20; n = 1,2,..,N$$

$$want\ \hat{z}_m^n = \mathrm{sgn}[\sum_{j=1}^{m-1} \alpha_j f_j(\underline{x}^n) + \alpha_m f_m(\underline{x}^n)]; M \approx 10-20; n = 1,2,..,N$$

But this is non-differentiable. Loss function: $L(z,\hat{z}) = 1 - \mathrm{sgn}(z\hat{z})$. Plot the function versus $z\hat{z}$.

Discuss Bounds: Exponential loss, log loss, hinge loss





$L(z,\hat{z}) \le L_h(z,\hat{z}) = [1 - z\hat{z}]_+ = \max(0, 1 - z\hat{z})$ hinge loss used in SVM

$L(z,\hat{z}) \le L_e(z,\hat{z}) = e^{-z\hat{z}}$ Exponential loss used in AdaBoost

$L(z,\hat{z}) \le L_l(z,y) = \ln(1 + e^{-zy}) / \ln 2$ Log loss

$y = \hat{z} = \underline{w}^T \underline{x}$ *or* $y = \hat{z} = \underline{w}^T \phi(\underline{x})$

$$L_m(f_m) = \sum_{n=1}^{N} \exp\{-z^n[\hat{z}_{m-1}(\underline{x}^n) + \alpha_m f_m(\underline{x}^n)]\} = \sum_{n=1}^{N} w_n^{(m)} e^{-\alpha_m z^n f_m(\underline{x}^n)}$$

$$w_n^{(m)} = \exp\{-z^n \hat{z}_{m-1}(\underline{x}^n)\} = \text{weight on data item } n \textit{....know these}!$$

$$L_m(f_m) = e^{-\alpha_m} \sum_{n=1}^{N} w_n^{(m)} \delta_{z^n f_m(\underline{x}^n)} + e^{\alpha_m} \sum_{n=1}^{N} w_n^{(m)} (1 - \delta_{z^n f_m(\underline{x}^n)}) = \underbrace{\left(e^{\alpha_m} - e^{-\alpha_m}\right) \sum_{n=1}^{N} w_n^{(m)} (1 - \delta_{z^n f_m(\underline{x}^n)})}_{\textit{depends on classifier } f_m} + e^{-\alpha_m} \sum_{n=1}^{N} w_n^{(m)}$$

$$\Rightarrow \text{Fit weak classifier } f_m \text{ by minimizing error function } J_m = \sum_{n=1}^{N} w_n^{(m)} (1 - \delta_{z^n f_m(\underline{x}^n)})$$

....Bootstrap sampling with weights $w_n^{(m)}$

$$\text{Optimal } \alpha_m \text{ from } e^{2\alpha_m} = \frac{\sum_{n=1}^{N} w_n^{(m)} \delta_{z^n f_m(\underline{x}^n)}}{\sum_{n=1}^{N} w_n^{(m)} (1 - \delta_{z^n f_m(\underline{x}^n)})}$$

$$= \frac{\sum_{n=1}^{N} w_n^{(m)} \delta_{z^n f_m(\underline{x}^n)} / \sum_{n=1}^{N} w_n^{(m)}}{\sum_{n=1}^{N} w_n^{(m)} (1 - \delta_{z^n f_m(\underline{x}^n)}) / \sum_{n=1}^{N} w_n^{(m)}} = \frac{1 - e_m}{e_m} \Rightarrow \alpha_m = \frac{1}{2} \ln\left(\frac{1 - e_m}{e_m}\right)$$

$$\Rightarrow w_n^{(m+1)} = w_n^{(m)} e^{-\alpha_m z^n f_m(\underline{x}^n)} = \begin{cases} w_n^{(m)} e^{-\alpha_m} & if \;\; f_m(\underline{x}^n) = z^n \\ w_n^{(m)} e^{\alpha_m} & if \;\; f_m(\underline{x}^n) \ne z^n \end{cases}$$

$$\hat{z} = \operatorname{sgn}[\sum_{m=1}^{M} \alpha_m f_m(\underline{x}^n)]; M \approx 10 - 20$$

Show via example

- Gradient Boosting: Idea originated from iterative solution of linear equations and least squares problems

$$\underline{y} = A\underline{x} \to \underline{x}_1 \to r_1 = \underline{y} - A\underline{x}_1 \to \underline{e}_1 \to \underline{x}_2 = \underline{x}_1 + \underline{e}_1 \to r_2 = \underline{y} - A\underline{x}_2 \to \underline{e}_2 \to \underline{x}_3 = \underline{x}_2 + \underline{e}_2, etc.$$

$$L_m(f_m) = \sum_{n=1}^{N} L(z^n, \hat{z}_{m-1}(\underline{x}^n) + f_m(\underline{x}^n)) + \Omega(f_m) \ldots \Omega(.) \text{ regularization term}$$

$$\approx \sum_{n=1}^{N} \{L(z^n, \hat{z}_{m-1}(\underline{x}^n)] + g_n f_m(\underline{x}^n) + \frac{1}{2} h_n [f_m(\underline{x}^n)]^2\} + \Omega(f_m);$$

$$g_n = \frac{\partial L(z^n, \hat{z})}{\partial \hat{z}} \mid \hat{z} = \hat{z}_{m-1}(\underline{x}^n); h_n = \frac{\partial^2 L(z^n, \hat{z})}{\partial \hat{z}^2} \mid \hat{z} = \hat{z}_{m-1}(\underline{x}^n)$$

$$\Omega(f_m) = \gamma T + \frac{\lambda}{2} \sum_{j=1}^{T} w_j^2; T = \# \text{ of leaves}$$

w_j = weight of leaf j (e.g., expected cost of path in the tree leading to leaf j)

$f_m(\underline{x}^n) = w_j$ if $\underline{x}^n$ is mapped to leaf j, i.e., $q(\underline{x}^n) = j; q$ is the mapping fuunction

$$L_m(f_m) \approx \sum_{n=1}^{N} \{L(z^n, \hat{z}_{m-1}(\underline{x}^n)] + g_n f_m(\underline{x}^n) + \frac{1}{2} h_n [f_m(\underline{x}^n)]^2\} + \gamma T + \frac{1}{2} \lambda \sum_{j=1}^{T} w_j^2$$

$$= \sum_{j=1}^{T} \{ \underbrace{\left(\sum_{n \in I_j} g_n \right)}_{G_j} w_j + \frac{1}{2} (\underbrace{\sum_{n \in I_j} h_n}_{H_j} + \lambda) w_j^2 + \gamma \} + \text{constant..... separable in } w_j \text{.... but huge possibilities for } \{I_j\}!$$

For known $\{I_j\}$, *optimal* $w_j = -\frac{G_j}{H_j + \lambda}; L_m(f_m) = \frac{1}{2} \sum_{j=1}^{T} (\frac{-G_j^2}{H_j + \lambda} + 2\gamma)$

$$Gain = \frac{1}{2} [\frac{G_{Lj}^2}{H_{Lj} + \lambda} + \frac{G_{Rj}^2}{H_{Rj} + \lambda} - \frac{G_j^2}{H_j + \lambda}] - \gamma; G_j = G_{Lj} + G_{Rj}; H_j = H_{Lj} + H_{Rj}$$

$$\hat{z}_m = \hat{z}_{m-1} + \varepsilon f_m; \varepsilon \approx 0.1$$

Percepron is an incremental gradient (stochastic gradient descent (SGD)) algorithm:

$$\underline{w} \leftarrow \underline{w} + \eta e_n \underline{x}_n$$

Why gradient algorithms work even if step size is off?

Consider $\min_{x,y} f(x, y) = \frac{h_1}{2} (x - 4)^2 + \frac{h_2}{2} (y - 2)^2 \Rightarrow x^* = 4, y^* = 2$

Suppose we start at (0,0)

$$\nabla f = \underline{g} = \begin{bmatrix} h_1(x-4) \\ h_2(y-2) \end{bmatrix}; \nabla^2 f = \begin{bmatrix} h_1 & 0 \\ 0 & h_2 \end{bmatrix}; h_i = \text{a measure of curvature>0}$$

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \end{bmatrix} - \eta \begin{bmatrix} h_1(x_k-4) \\ h_2(y_k-2) \end{bmatrix} \Rightarrow \begin{bmatrix} x_{k+1}-4 \\ y_{k+1}-2 \end{bmatrix} = \begin{bmatrix} (1-\eta h_1)(x_k-4) \\ (1-\eta h_2)(y_k-2) \end{bmatrix}$$

$$\Rightarrow \textit{need } -1 < 1-\eta h_i < 1 \Rightarrow \eta > 0 \;\&\; \eta < \min_i\left(\frac{2}{h_i}\right) = \frac{2}{\max(h_i)}$$

$$\textit{optimal } \eta : \frac{h_1}{2}(1-\eta h_1)^2(x_k-4)^2 + \frac{h_2}{2}(1-\eta h_2)^2(y_k-2)^2$$

$$\Rightarrow -h_1^2(1-\eta h_1)(x_k-4)^2 - h_2^2(1-\eta h_2)(y_k-2)^2 = 0$$

$$\Rightarrow \eta = \frac{h_1^2(x_k-4)^2 + h_2^2(y_k-2)^2}{h_1^3(x_k-4)^2 + h_2^3(y_k-2)^2}$$

$$\textit{If } h_1 >> h_2, \eta \approx \frac{1}{h_1} \ldots\ldots \eta \text{ can be off by a factor of 2 and still get function reduction!}$$

LMS:

$$\underline{w}^{(n+1)} = \underline{w}^{(n)} + \eta^n e_n \underline{x}^n ; \eta^n = \frac{\lambda}{\left\|\underline{x}^n\right\|^2}$$ $0<\lambda<2$. $\lambda=1$ implies projection.

$$\min_{\underline{w}} \frac{1}{2}\left\|\underline{w} - \underline{w}^{(n)}\right\|^2 \qquad \text{s.t. } \underline{w}^T \underline{x}^n = z^n$$

$$L(\underline{w}, \mu) = \frac{1}{2}(\underline{w} - \underline{w}^{(n)})^T(\underline{w} - \underline{w}^{(n)}) + \mu\left[z^n - \underline{w}^T \underline{x}^n\right]$$

$$\nabla_{\underline{w}} L = \underline{w} - \underline{w}^{(n)} - \mu \underline{x}^n = 0 \qquad \text{using } \underline{w}^T \underline{x}^n = z^n$$

$$\Rightarrow \mu = \frac{z^n - \underline{w}^{(n)T} \underline{x}^n}{\left(\underline{x}^n\right)^T \underline{x}^n}$$

$$\Rightarrow w^{(n+1)} = w^{(n)} + \frac{(z^n - \underline{w}^{(n)T} \underline{x}^n)}{\left\|\underline{x}^n\right\|^2} \underline{x}^n$$

Convergence of LMS: Convergence in the mean; convergence in mean square

Momentum: $\underline{w}^{(n+1)} = \underline{w}^{(n)} + \eta(z^n - \underline{w}^{(n)T} \underline{x}^n)\underline{x}^n + \mu(\underline{w}^{(n)} - \underline{w}^{(n-1)})$

$$\left.\begin{array}{l} \Delta \underline{w}^{(n)} = \mu \Delta \underline{w}^{(n-1)} + \eta e_n \underline{x}^n \\ \textit{or } \underline{d}^{(n)} = \mu \underline{d}^{(n-1)} - \eta \underline{g}^{(n)} \\ \text{clearly need } \mu < 1 \end{array}\right\}$$

Nesterov: $\underline{d}^{(n)} = \mu \underline{d}^{(n-1)} - \eta \underline{g}^{(n)} \big|_{\underline{w}^{(n)} + \mu \underline{d}^{(n-1)}}$ ...*gradient evaluated at* $\underline{w}^{(n)} + \mu \underline{d}^{(n-1)}$

Bold-driver: $\eta^{new} = \begin{cases} \rho\ \eta^{old} & \text{if } \Delta J < 0 \ \Rightarrow \text{ improved} \\ \sigma\ \eta^{old} & \text{if } \Delta J > 0 \end{cases} \quad \rho = 1.1 \quad \sigma \approx 0.5$

AdaGrad:

$$w_i^{(n+1)} = w_i^{(n)} - \eta_i^{(n)} g_i^{(n)}; \eta_i^{(n)} = \frac{\eta}{\sqrt{\sum_{j=1}^{n} \left(g_i^{(j)}\right)^2 + \varepsilon}} = \frac{\eta}{\sqrt{G_n + \varepsilon}}; \varepsilon \approx 10^{-8}$$

$$G_n = G_{n-1} + \left(g_i^{(n)}\right)^2; G_0 = 0$$

RMSprop:

$$G_n = \gamma G_{n-1} + (1-\gamma) \| \underline{g}^{(n)} \|_2^2; G_0 = 0; \gamma \approx 0.9$$

$$\underline{w}^{(n+1)} = \underline{w}^{(n)} - \frac{\eta}{\sqrt{G_n + \varepsilon}} \underline{g}^n; \eta \approx 0.001$$

Adam:

$$\overline{g}^{(n)} = \theta\ \overline{g}^{(n-1)} + (1-\theta) g^{(n)}; \overline{g}^{(0)} = 0$$

$$G_n = \gamma G_{n-1} + (1-\gamma) \| \underline{g}^{(n)} \|_2^2$$

$$\underline{w}^{(n+1)} = \underline{w}^{(n)} - \frac{\eta^{(n)}}{\sqrt{G_n + \varepsilon}} \underline{g}^n; \eta^{(n)} = \eta \frac{\sqrt{1-\gamma^t}}{1-\theta^t}$$

Quick prop: $w_i^{(n+1)} - w_i^{(n)} = \dfrac{g_i^{(n)}}{g_i^{(n-1)} - g_i^{(n)}} \left[w_i^{(n)} - w_i^{(n-1)} \right]$

Single layer network

Incremental Newton and RLS

Modified RLS = Gauss-Newton = EKF

Fisher's Linear Discriminant