Gaussian NB

$$\hat{\mu}_{ik} = \frac{1}{\sum_j \delta(Y^j = y_k)} \sum_j X_i^j \delta(Y^j = y_k) \quad \hat{\sigma}_{ik}^2 = \frac{1}{\sum_j \delta(Y^j = y_k) - 1} \sum_j (X_i^j - \hat{\mu}_{ik})^2 \delta(Y^j = y_k) \quad \hat{\sigma}^2_{MLE} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{\mu})^2$$

$$P(x \mid \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \qquad P(\mathcal{D} \mid \mu, \sigma) = \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^N \prod_{i=1}^{N} e^{\frac{-(x_i-\mu)^2}{2\sigma^2}}$$

MAP

$$\hat{\theta} = \arg\max_{\theta} P(\theta \mid \mathcal{D}) \qquad \hat{\mu}_{MLE} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

MLE

$$\hat{\theta} = \arg\max_{\theta} \; P(\mathcal{D} \mid \theta)$$

Derivatives

$$(cf)' = cf'(x)$$
$$(f \pm g)' = f'(x) \pm g'(x)$$
$$(fg)' = f'g + fg' \; \textbf{– Pr}$$
$$\left(\frac{f}{g}\right)' = \frac{f'g - fg'}{g^2} \; \textbf{– Qu}$$

$$\frac{d}{dx}(a^x) = a^x \ln(a)$$
$$\frac{d}{dx}(\mathbf{e}^x) = \mathbf{e}^x$$
$$\frac{d}{dx}(\ln(x)) = \frac{1}{x}, \; x > 0$$
$$\frac{d}{dx}(\ln|x|) = \frac{1}{x}, \; x \neq 0$$
$$\frac{d}{dx}(\log_a(x)) = \frac{1}{x \ln a}, \; x > 0$$

SVM

$$g(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b \qquad g(\mathbf{x}) = \sum_{i \in SV} \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b$$

$$\alpha_i \left(y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 \right) = 0 \qquad \mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = \sum_{i \in SV} \alpha_i y_i \mathbf{x}_i$$

minimize $\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i$

such that

$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i$

$\xi_i \geq 0 \qquad 0 \leq \alpha_i \leq C$

minimize $L_p(\mathbf{w}, b, \alpha_i) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i \left(y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 \right)$

s.t. $\alpha_i \geq 0$

maximize $\sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$

s.t. $\alpha_i \geq 0$, and $\sum_{i=1}^{n} \alpha_i y_i = 0$

Perceptron

AND $\quad f(X_1, \ldots, X_n) = X_1 \wedge \ldots \wedge X_k \wedge \neg X_{k+1} \wedge \ldots \wedge \neg X_n$

$w_0 = -k + 0.5; \quad w_1 = \ldots = w_k = 1 \quad w_{k+1} = \ldots = w_n = -1.$

$w_0 = -n + 0.5;$

OR $\quad g(X_1, \ldots, X_n) = X_1 \vee \ldots \vee X_k \vee \neg X_{k+1} \vee \ldots \vee \neg X_n$

$w_0 = n - k - 0.5; \quad w_0 = n - 0.5 \qquad +1$ to a and -1 to b.

$w_0 = \frac{a+b}{2}$ and $w_1 = \frac{a-b}{2}$.

Neural Net

- Initialize w_i's and w_0 to random values.
- Until Convergence do
  - $\Delta w_0 = 0$
  - $\Delta w_i = 0$ for $i = 1$ to n
  - For each training example indexed by j do
    - Compute $o_j = w_0 + \sum_{i=1}^{n} w_i(x_i + x_i^{1.5})$
  - For each training example indexed by j do
    - $\Delta w_0 = \Delta w_0 + (y_j - o_j)^2$
    - $\Delta w_i = \Delta w_i + (y_j - o_j)^2(x_{i,j} + x_{i,j}^{1.5})$ for $i = 1$ to n
  - $w_0 = w_0 + \eta \Delta w_0$
  - $w_i = w_i + \eta \Delta w_i$ for $i = 1$ to n

Backpropagation Algorithm

Initialize all weights to small random numbers
Until convergence, Do
For each training example, Do

1. Input it to network and compute network outputs
2. For each output unit k
   $\delta_k \leftarrow o_k(1 - o_k)(t_k - o_k)$
3. For each hidden unit h
   $\delta_h \leftarrow o_h(1 - o_h) \sum_{k \in outputs} w_{h,k}\delta_k$
4. Update each network weight $w_{i,j}$
   $w_{i,j} \leftarrow w_{i,j} + \Delta w_{i,j}$
   where $\Delta w_{i,j} = \eta \delta_j x_{i,j}$

Linear

$$\frac{\partial E}{\partial w_i} = \sum_d (t_d - o_d)(-x_{i,d})$$

$$\frac{\partial E}{\partial w_i} = \frac{\partial}{\partial w_i}\frac{1}{2}\sum_d (t_d - o_d)^2 = \frac{1}{2}\sum_d \frac{\partial}{\partial w_i}(t_d - o_d)^2 = \frac{1}{2}\sum_d 2(t_d - o_d)\frac{\partial}{\partial w_i}(t_d - o_d) = \sum_d (t_d - o_d)\frac{\partial}{\partial w_i}(t_d - \vec{w} \cdot \vec{x_d})$$

Sigmoid

$$\frac{\partial E}{\partial w_i} = -\sum_{d \in D}(t_d - o_d)o_d(1 - o_d)x_{i,d}$$

$$\delta_h \leftarrow o_h(1 - o_h)\sum_{k \in outputs} w_{h,k}\delta_k$$

- Training rule for linear unit: $o = \sum_{i=0}^{n} w_i x_i$

$$w_i = w_i + \Delta w_i$$

where

$$\Delta w_i = -\eta\frac{\partial E}{\partial w_i} = \eta(t - o)x_i$$

- Training rule for Sigmoid unit $o = Sig(\sum_{i=0}^{n} w_i x_i)$

$$\Delta w_i = -\eta\frac{\partial E}{\partial w_i} = \eta(t - o)o(1 - o)x_i$$

Minkowski distance

$$L_k(x_i, x_j) = \left(\sum_{a=1}^{d} |x_{i,a} - x_{j,a}|^k\right)^{\frac{1}{k}}$$

$$L_\infty(x_i, x_j) = \max_a |x_{i,a} - x_{j,a}|$$

tanh -> $f'(z) = 1 - (f(z))^2$

tanh(z) rescaled sigmoid [-1,1] instead of [0,1]

Linear reg

$$w_1 = \frac{m\sum_{i=1}^{m} x_i y_i - \sum_{i=1}^{m} x_i \sum_{i=1}^{m} y_i}{m\sum_{i=1}^{m} x_i^2 - (\sum_{i=1}^{m} x_i)^2}$$

$$w_0 = \frac{\sum_{i=1}^{m} y_i - w_1\sum_{i=1}^{m} x_i}{m}$$

$$JR(\mathbf{w}) = \sum_{i=1}^{m}\left(y_i - \sum_{j=1}^{n} w_j x_{i,j}\right)^2 + \frac{\lambda}{2}\sum_{j=1}^{m}|w_j|^q$$

$$J(\mathbf{w}) = \sum_{i=1}^{m}(y_i - \mathbf{w}^T x_i)^2$$

$$\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y} \qquad w_j = w_j - \alpha\frac{\partial}{\partial w_j}J(\mathbf{w})$$

$$w_i \leftarrow w_i + \eta\sum_l X_i^l(Y^l - \hat{P}(Y^l = 1|X^l, W)) - \eta\lambda w_i$$

$$W \leftarrow \arg\max_W \sum_l \ln P(Y^l|X^l, W) - \frac{\lambda}{2}||W||^2$$

Logistic Reg

$$W \leftarrow \arg\max_W \sum_l \ln P(Y^l|X^l, W) \qquad P(Y = 1|X) = \frac{\exp(w_0 + \sum_{i=1}^{n} w_i X_i)}{1 + \exp(w_0 + \sum_{i=1}^{n} w_i X_i)}$$

$$w_i \leftarrow w_i + \eta\sum_l X_i^l(Y^l - \hat{P}(Y^l = 1|X^l, W)) \qquad P(Y = 0|X) = \frac{1}{1 + \exp(w_0 + \sum_{i=1}^{n} w_i X_i)}$$

$$l(W) = \sum_l Y^l(w_0 + \sum_i^n w_i X_i^l) - \ln(1 + \exp(w_0 + \sum_i^n w_i X_i^l))$$

$$\frac{\partial l(W)}{\partial w_i} = \sum_l X_i^l(Y^l - \hat{P}(Y^l = 1|X^l, W)) \qquad p(\mathbf{w}) = \prod_i \frac{1}{\kappa\sqrt{2\pi}} e^{\frac{-w_i^2}{2\kappa^2}}$$

Pruning & regularization increases bias reduces variance
Linear regressin: Loss +penalty (bcz minimizing)
Log regression: Loss – penalty (bcz maximizing)
Under zero mean, conditional independent variance assumption,
LogR = Gaussian NB
NB: Features independent given class ! assumption on P(**X**|Y)
LR: Functional form of P(Y|**X**), no assumption on P(**X**|Y)
therefore LR expected to outperform GNB
Naïve Bayes needs O(log n) samples
Logistic Regression needs O(n) samples
LR- conditional likelihood

KMean

$$\phi(\{x_i\},\{a_i\},\{c_k\}) = \sum \text{dist}(x_i, c_{a_i})$$

HMM

Filtering: $\mathbf{P}(\mathbf{X}_t|\mathbf{e}_{1:t})$
belief state—input to the decision process of a rational agent

Prediction: $\mathbf{P}(\mathbf{X}_{t+k}|\mathbf{e}_{1:t})$ for $k > 0$
evaluation of possible action sequences;
like filtering without the evidence

Smoothing: $\mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:t})$ for $0 \le k < t$
better estimate of past states, essential for learning

Most likely explanation: $\arg\max_{\mathbf{x}_{1:t}} P(\mathbf{x}_{1:t}|\mathbf{e}_{1:t})$
speech recognition, decoding with a noisy channel

$$\begin{aligned}\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t+1}) &= \mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t}, \mathbf{e}_{t+1})\\ &= \alpha\mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1}, \mathbf{e}_{1:t})\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t}) \quad \text{Bayes Rule}\\ &= \alpha\mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1})\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t}) \quad \text{Markov assumption}\end{aligned}$$

I.e., prediction + estimation. Prediction by summing out $\mathbf{X}_t$:

$$\begin{aligned}\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t+1}) &= \alpha\mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1})\Sigma_{\mathbf{x}_t}\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{x}_t, \mathbf{e}_{1:t})P(\mathbf{x}_t|\mathbf{e}_{1:t})\\ &= \alpha\mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1})\Sigma_{\mathbf{x}_t}\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{x}_t)P(\mathbf{x}_t|\mathbf{e}_{1:t})\end{aligned}$$

Forward–backward algorithm: cache forward messages along the way
Time linear in t (polytree inference), space $O(t|\mathbf{f}|)$

$\mathbf{f}_{1:t+1} = \text{FORWARD}(\mathbf{f}_{1:t}, \mathbf{e}_{t+1})$ where $\mathbf{f}_{1:t} = \mathbf{P}(\mathbf{X}_t|\mathbf{e}_{1:t})$
Time and space **constant** (independent of t)
Divide evidence $\mathbf{e}_{1:t}$ into $\mathbf{e}_{1:k}$, $\mathbf{e}_{k+1:t}$:

$$\begin{aligned}\mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:t}) &= \mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:k}, \mathbf{e}_{k+1:t})\\ &= \alpha\mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:k})\mathbf{P}(\mathbf{e}_{k+1:t}|\mathbf{X}_k, \mathbf{e}_{1:k}) \quad \text{Bayes Rule}\\ &= \alpha\mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:k})\mathbf{P}(\mathbf{e}_{k+1:t}|\mathbf{X}_k) \quad \text{Markov assumptic}\\ &= \alpha\mathbf{f}_{1:k}\mathbf{b}_{k+1:t}\end{aligned}$$

Backward message computed by a backwards recursion:

$$\begin{aligned}\mathbf{P}(\mathbf{e}_{k+1:t}|\mathbf{X}_k) &= \Sigma_{\mathbf{x}_{k+1}}\mathbf{P}(\mathbf{e}_{k+1:t}|\mathbf{X}_k, \mathbf{x}_{k+1})\mathbf{P}(\mathbf{x}_{k+1}|\mathbf{X}_k)\\ &= \Sigma_{\mathbf{x}_{k+1}}P(\mathbf{e}_{k+1:t}|\mathbf{x}_{k+1})\mathbf{P}(\mathbf{x}_{k+1}|\mathbf{X}_k)\\ &= \Sigma_{\mathbf{x}_{k+1}}P(\mathbf{e}_{k+1}|\mathbf{x}_{k+1})P(\mathbf{e}_{k+2:t}|\mathbf{x}_{k+1})\mathbf{P}(\mathbf{x}_{k+1}|\mathbf{X}_k)\end{aligned}$$

ADA BOOST

- α_t :
  - No errors: ε_t=0 → α_t=∞
  - All errors: ε_t=1 → α_t=−∞
  - Random: ε_t=0.5 → α_t=0

VC

c belongs to C
$$m \ge \frac{1}{\epsilon}(\ln(1/\delta) + \ln(|H|))$$

c not known
$$m \ge \frac{1}{2\epsilon^2}(\ln(1/\delta) + \ln(|H|))$$

infinite |H|
$$m \ge \frac{1}{\epsilon}(4\log_2(2/\delta) + 8VC(H)\log_2(13/\epsilon))$$

Entropy

$$GainRatio(S,A) \equiv \frac{Gain(S,A)}{SplitInformation(S,A)}$$

$$H(V) = \sum_{v=0}^{1} -P(H=v)\lg P(H=v).$$

$$SplitInformation(S,A) \equiv -\sum_{i=1}^{c} \frac{|S_i|}{|S|}\log_2\frac{|S_i|}{|S|}$$

$$Gain(S,A) = H(S) - \sum_{v \in Values(A)} P(v)H(S_v)$$

Beta Prior

Prior: $Beta(\beta_H, \beta_T)$
Data: α_H heads and α_T tails

$$\widehat{\theta} = \arg\max_{\theta} P(\theta \mid \mathcal{D}) = \frac{\alpha_H + \beta_H - 1}{\alpha_H + \beta_H + \alpha_T + \beta_T - 2}$$

- Likelihood function: $P(\mathcal{D} \mid \theta) = \theta^{\alpha_H}(1-\theta)^{\alpha_T}$
- Posterior: $P(\theta \mid \mathcal{D}) \propto P(\mathcal{D} \mid \theta)P(\theta)$

$$\begin{aligned}P(\theta \mid \mathcal{D}) &\propto \theta^{\alpha_H}(1-\theta)^{\alpha_T}\ \theta^{\beta_H-1}(1-\theta)^{\beta_T-1}\\ &= \theta^{\alpha_H+\beta_H-1}(1-\theta)^{\alpha_T+\beta_T-1}\\ &= Beta(\alpha_H+\beta_H, \alpha_T+\beta_T)\end{aligned}$$

Mixture Model

- Given a dataset: $D = \{\underline{x}_1, \ldots, \underline{x}_N\}$
- **Mixture model**: $\Theta = \{\alpha_1, \ldots, \alpha_K, \theta_1, \ldots, \theta_K\}$

$$p(\underline{x}|\Theta) = \sum_{k=1}^{K} \alpha_k p_k(\underline{x}|z_k, \theta_k)$$

The $p_k(\underline{x}|z_k, \theta_k)$ are *mixture components*, $1 \le k \le K$

$z = (z_1, \ldots, z_K)$ is a vector of K binary indicator variables

Note: only one of them equals 1 at any given point. Each point is assumed to be generated from exactly one mixture component!

Mixture Weights. $\alpha_k = p(z_k)$ $\quad \sum_{k=1}^{K} \alpha_k = 1.$

the "membership weight" of data point $\underline{x}_i$ in cluster k, given parameters Θ

$$w_{ik} = p(z_{ik} = 1|\underline{x}_i, \Theta) = \frac{p_k(\underline{x}_i|z_k, \theta_k) \cdot \alpha_k}{\sum_{m=1}^{K} p_m(\underline{x}_i|z_m, \theta_m) \cdot \alpha_m}$$

Gaussian Mixture Models (GMMs)

$$p_k(\underline{x}|\theta_k) = \frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} e^{-\frac{1}{2}(\underline{x}-\underline{\mu}_k)^t\Sigma_k^{-1}(\underline{x}-\underline{\mu}_k)}$$

EM

Solution: As discussed in class, you can either update the sufficient statistics using the variable elimination algorithm for each example or complete each example. The complexity is the minimum of the two.

The complexity of completing the dataset is $O(2n \times 9)$ because there are two possible completions for each instance and there are nine functions which we have to multiply to compute the probability or weight for each example. Since the maximum CPT size is $O(\exp(3))$, we will need $O(9\exp(3))$ to normalize the CPTs at the end. Thus the overall time complexity is $O(18n + 9\exp(3))$.