Multiclass Perceptron:

Predict : $\hat{y} = \arg\max_y W_y \cdot \varphi(x)$

If correct, no change

If wrong: lower score of wrong answer, raise right ans

$W_{\hat{y}} = W_{\hat{y}} - \varphi(x)$ , $W_{y^*} = W_{y^*} + \varphi(x)$

If the training set is separable, perceptron will converge

$$P(y_i|x_i) = \frac{e^{w_{y_i} \cdot x_i}}{\sum_y e^{w_y \cdot x_i}}$$

$$ll(w) = \sum_i \log P_w(y_i|x_i) = \sum_i w_{y_i} \cdot x_i - \log \sum_y e^{w_y \cdot x_i}$$

$$\frac{d}{dw_y} \log P_w(y_i|x_i) = x_i(I(y = y_i) - P(y|x_i))$$

Gradient descent

initialize $w$ (e.g., randomly)

repeat for $K$ iterations:

for each example $(x_i, y_i)$:

compute gradient $\Delta_i = -\nabla_w \log P_w(y_i|x_i)$

compute gradient $\nabla_w \mathcal{L} = \sum_i \Delta_i$

$w \leftarrow w - \alpha \nabla_w \mathcal{L}$

Stochastic Gradient Descent

initialize $w$ (e.g., randomly)

repeat for $K$ iterations:

for each example $(x_i, y_i)$:

compute gradient $\Delta_i = -\nabla_w \log P_w(y_i|x_i)$

$w \leftarrow w - \alpha \Delta_i$

Neural Network

$$z_i^{(k)} = g\left(\sum_j W_{i,j}^{(k-1,k)} z_j^{(k-1)}\right)$$

g = nonlinear activation function

Gibbs sampling: only CPTs that have resampled variable need to be considered and join together.

Maximum-likelihood sampling: weights are product of P(observation | parents(observation))

HMM two steps

① predict $P(X_{t+1}|e_{1:t})$

看新模型，有多的 given evidence 就在表达式中加上

② update $P(X_{t+1}|e_{1:t}, e_{t+1}) \propto$

多出来的 evidence 在 Bayes net 里的 P $\times$ $P(X_{t+1}|e_{1:t})$

(如果有 new evidence 但是 unknown，要求和)

Naive Bayes: $P(Y, W_1, \dots, W_n) = P(Y) \prod_{i=1}^{n} P(W_i|Y)$

Laplace Smoothing: $P_{LAP,k}(x) = \frac{c(x)+k}{N+k|X|}$

$\alpha \models \beta$ : models($\alpha$) $\subseteq$ models($\beta$)

$\alpha \Rightarrow \beta = \neg\alpha \vee \beta$ , $\alpha \Leftrightarrow \beta = (\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$

valid: true in every model

satisfiable: true in some models