DATE / / OM OT OW OT OF OS OS | NOTES

1. For Naive Bayes: consider using three features:
At training time, we have to estimate parameters:
$\theta_0 = P(Y=0)$, $\theta_1 = P(Y=1)$, $\theta_{i,j,k,t} = P(X_1=i, X_2=j, X_3=k \mid Y=t)$
where $i, j, k, t$ are binary ($\in \{0,1\}$), and essentially $j=k$.
We have to store a total of $2+2^4 = 18$ parameters, although 8 of them where $j \neq k$ are always 0.
At test time, assume we have to classify $(a, b, c)$, then $P(Y=0 \mid X_1=a, X_2=b, X_3=c)$

$$= \frac{P(Y=0) \cdot P(X_1=a, X_2=b, X_3=c \mid Y=0)}{P(X_1=a, X_2=b, X_3=c)}$$

$$\overset{X_2=X_3}{=} \frac{P(Y=0) \cdot P(X_1=a, X_2=b \mid Y=0)}{P(X_1=a, X_2=b)} = P(Y=0 \mid X_1=a, X_2$$

This is the same as if we predict this input with only two features using the NB model trained on only two features. So at test time, there's no differen
Although at training time we have to calculate 8 extra parameters which are are 0 and store them.

2. For Logistic Regression without regularization,
Consider using three features:
At training time, we have to maximize the data likelihood $L(w) := \sum_{i=1}^{m} \log p(y_i \mid x_i, w)$
where $m$ is the number of training examples and

$$p(y_i \mid x_i, w) = \frac{\exp(w_{y_i}^T f(x_i))}{\sum_y \exp(w_y^T f(x_i))}$$

$$= \frac{\exp(w_{y_0} + w_{y_1} x_1 + w_{y_2} x_2 + w_{y_3} x_3)}{\exp(w_{00} + w_{01} x_1 + w_{02} x_2 + w_{03} x_3) + \exp(w_{10} + w_{11} x_1 + w_{12} x_2 + w_{13} x_3)}$$

$$\frac{\partial L(w)}{\partial w_y} = \sum_{i=1}^{m} I(y_i = y) f(x_i) - \sum_{i=1}^{m} p(y \mid x_i, w) f(x_i) \quad y \in \{0,1\}$$

Train weights to converge, we have $\frac{\partial L(w)}{\partial w_y} = 0 \quad y \in \{0,1\}$

i.e., $\sum_{i=1}^{m} I(y_i = y) f(x_i) = \sum_{i=1}^{m} p(y \mid x_i, w) f(x_i) \quad y \in \{0,1\}$

which means each feature's predicted expectation equals its empirical expectation from training data.
So at training time, we need to do more computation.
At test time, the prediction rule is:

$$\hat{y}_i = \arg\max_{y \in Y} w_y^T f(x_i) = \arg\max_{y \in \{0,1\}} w_{y_0} + w_{y_1} x_1 + w_{y_2} x_2 + w_{y_3} x_3$$

$$= \arg\max_{y \in \{0,1\}} w_{y_0} + w_{y_1} x_1 + (w_{y_2} + w_{y_3}) x_2$$

Based on the optimization method used in training,

$w_{y2}$ and $w_{y3}$ might be different, where $y \in \{0,1\}$. but their sum should be the same as $w_{y2}'$, their conterpart when trained using only two features, to ensure the same prediction result.

3. If adding $L_2$ regularization to logistic regression, when using three features, large feature weights will be penalized, since now the likelihood function is:

$$L(w) = -k\|w\|^2 + \sum_{i=1}^{m} \log P(y_i \mid x_i, w)$$

However, since we have two duplicate features, they can "collaborate" to minimize their $L_2$ penalization while keeping their sum the same.

For example, assume we have trained a Logistic Regression model using only two features, and the weight for feature $x_2$ associated with label 0 is $W_{02} = 1$, with penalization $K$. The three features model can keep the same penalization while maximizing $(W_{02} + W_{03})$, by setting $W_{02} = \frac{\sqrt{2}}{2}$ and $W_{03} = \frac{\sqrt{2}}{2}$, yeilding the same penalization $K = K[(\frac{\sqrt{2}}{2})^2 + (\frac{\sqrt{2}}{2})^2]$. But now $W_{02} + W_{03} = \sqrt{2}$, which indicates that the model with three features will give more weights on feature $x_2$ (i.e. $x_3$) than $x_1$. And thus this model is biased and will give different predict result

1. Training an HMM is faster (in big O notation) than training a CRF.

Answer: True.

For Supervised learning, training an HMM is simply using MLE to estimate parameters. While training a CRF requires gradient computing, and forward-backward is used to compute the partition function in the likelihood as well as the marginal distributions in the gradient.

2. You can use HMM in unsupervised POS tagging but not CRFs.

Answer: False

We can apply expectation-maximization (EM) algorithm to both HMMs and CRFs to leverage unsupervised data.

3. Assume training is complete. Given a set of observations. Compute the probability of a sequence of hidden states in HMM is faster (in big-O notation) than CRF.

Answer: True.

Assume we have an underlying language model for $p(w)$. For an HMM:

$$p(s|w) = \frac{p(s,w)}{p(w)}$$

where $p(s,w) = \prod_i p(s_i|s_{i-1})\, p(w_i|s_i)$ is modeled by the HMM model, and can be computed in $O(L)$ time, $L$ is the sequence length.

$p(w)$ is the language model, and can be computed in $O(L)$ time.

So the total time complexity for HMM is $O(L)$

assume a language model for $p(w)$ is provided.

(If no language model of $p(w)$ is provided, then we must sum over all possible states to get $p(w)$, i.e., $p(w) = \sum_{s \in S} p(s,w)$, and this cost is exponential: $O(L^S)$ )

For an CRF, the model gives us $p(s|w)$ directly where $p(s|w) = \frac{1}{Z(w)} \prod_i \psi(s_i, s_{i-1}, w)$,

$$\psi(s_i, s_{i-1}, w) = \exp\left(\sum_k \alpha_{ik} f_k(s_i, s_{i-1}, w)\right),$$

$$Z(w) = \sum_{s_1, \cdots s_n} \prod_i \psi(s_i | s_{i-1}, w).$$

Since $Z(w)$ can be computed using Viterbi in $O(S^2 L)$ time, the product term remaining can be computed using $O(KL)$ time, where K is the number of features, S is the number of states, and L is the sequence length. The total time complexity for a CRF is $O(S^2L + KL)$.

4. Assume training is complete. Given a set of observations, compute the most likely sequence of hidden states in HMMs is faster (in big O notation) than it is in CRFs.

Answer: False

This is known as decoding. We can apply Viterbi algorithm to both HMM and CRF to decode a sequence in $O(S^2L)$ time, where S is the number of states, and L is the sequence length. So the time complexity should be the same.