CS 188 Midterm 2

Probability

Product Rule:

$$P(y)P(x|y) = P(x,y)$$

Chain Rule:

$$P(x_1, x_2, x_3) = P(x_1)P(x_2|x_1)P(x_3|x_1, x_2)$$

Bayes Rule:

$$P(x,y) = P(x|y)P(y) = P(y|x)P(x)$$

$$P(x|y) = \frac{P(y|x)}{P(y)}P(x)$$

$$P(\texttt{Cause}|\texttt{Effect}) = \frac{P(\texttt{Effect}|\texttt{Cause})P(\texttt{Cause})}{P(\texttt{Effect})}$$

Independence:

Two variable are independent in a joint distribution if:

$$P(x,y) = P(x)P(y)$$

$$X \perp\!\!\!\perp Y$$

$$\forall x, yP(x,y) = P(x)P(y)$$

Conditional Independence

X is conditionally independent of Y given Z:

$$X \perp\!\!\!\perp Y|Z$$

if and only if:

$$\forall x, y, z : P(x,y|z) = P(x|z)P(y|z)$$

or equivalently, if and only if:

$$\forall x, y, z : P(x|z,y) = P(x|z)$$

Inference: Calculating some useful quantity from a joint probablility distribution.

Example:
Posterior probability

$$P(Q|E_1 = e_1, \ldots E_k = e_k)$$

Most likely explanation:

$$\texttt{argmax}_q P(Q = q|E_1 = e_1 \ldots)$$

Bayes' Net Sampling

Prior Sampling P
Rejection Sampling $P(Q|e)$
Likelihood Weighting $P(Q|e)$

Gibbs Sampling $P(Q|e)$

Step 1: Fix evidence
Step 2: Initialize other variables randomly
Step 3: Repeat

Expected Utility

$$\sum_a P(a)U(a,t)$$

Max Expected Utility:

$$MEU(\emptyset) = max_a EU(a)$$

Value of Perfect Information

$$VPI(E'|e) = \left(\sum_{e'} P(e'|e)MEU(e,e')\right) - MEU(e)$$

Assume we have evidence E=e. Value if we act now:

$$MEU(e) = max_a \sum_s P(s|e)U(s,a)$$

Assume we see E'=e'. Value if we act now:

$$MEU(e,e') = max_a \sum_s P(s|e,e')U(s,a)$$

but E' is a random variable whose value is unknown, so we don't know what e' will be.
Expected value if E' is revealed and then we act:

$$MEU(e,E') = \sum_{e'} P(e',e)MEU(e,e')$$

Value of information: how much MEU goes up by revealing E' first then acting, over acting now:

$$VPI(E'|e) = MEU(e,E') - MEU(e)$$

Properties:

Nonnegative:

$$\forall E', e : VPI(E'|e) \geq 0$$

Nonadditive:

$$VPI(E_j, E_k|e) \neq VPI(E_j|e) + VPI(E_k|e)$$

Order-independent:

$$VPI(E_j, E_k|e) = VPI(E_j|e) + VPI(E_k|e, E_j)$$
$$= VPI(E_k|e) + VPI(E_j|e, E_k)$$

Markov Models

Forward Algorithm:

$$\texttt{given } P(x_1) = \texttt{known}, \; P(x_t) = \sum_{x_{t-1}} P(x_t|x_{t-1})P(x_{t-1})$$

$$\texttt{satisfying}, \; P_\infty(X) = P_{\infty+1}(X) = \sum_x P_{t+1|t}(X|x)P_\infty(x)$$

Hidden Markov Models

Assume we have current belief P(X — evidence to date):

$$B(X_t = P(X_t|e_{1:t})$$

Then after one time step passes (Elapse Time):

$$B'(X_{t+t}) = \sum_{x_t} P(X'|x)B(x_t)$$

Result of an observation, given current Belief:

$$B'(X_{t+1}) = P(X_{t+1}|e_{1:t})$$

Dynamic Bayes Net Particle Filters

Initialize: Generate prior samples for the t=1 Bayes net

$$\texttt{Example particle:} \quad G_1^a = (3,3), G_1^b = (5,3)$$

Elapse time: Sample a successor for each particle

$$\texttt{Example successor:} \quad G_2^a = (2,3), G_2^b = (6,3)$$

Observe: Weight each entire sample by the likelihood of the evidence conditioned on the sample

$$\texttt{Likelihood:} \quad P(E_1^a|G_1^a) * P(E_1^b|G_1^b)$$

Resample: Select prior samples (tuples of values) in proporation to their likelihood

Parameter Estimation

Empirical Rate (Maximum Likelihood Estimate):

$$P_{ML}(x) = \frac{\texttt{count}(x)}{\texttt{total samples}}$$

Likelihood of the data:

$$L(x, \Theta) = \prod_i P_\Theta(x_i)$$

Laplace Smoothing:

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

Laplace for conditionals:

$$P_{LAP,k}(x|y) = \frac{c(x,y) + k}{c(y) + k|X|}$$

Linear Interpolation:

$$P_{LIN}(x|y) = \alpha\hat{P}(x|y) + (1.0 - \alpha)\hat{P}(x)$$

Perceptrons

Linear Classifiers:

$$\texttt{activation}_w(x) = \sum_i w_i * f_i(x) = w * f(x)$$

Learning: Binary Perceptron:

Start with weights = 0
classify with the current weights:

$$y = \begin{cases} +1 & \texttt{if } w * f(x) \geq 0 \\ -1 & \texttt{if } w * f(x) < 0 \end{cases}$$

Multiclass Perceptron

Start with all weights = 0
Predict with current weights:

$$y = \texttt{arg max}_y w_y * f(x)$$

If correct make no change
If wrong: lower score of wrong answer, raise score of right answer

$$w_y = w_y - f(x)$$
$$w_{y*} = w_{y*} + f(x)$$

Properties of the perceptron

Separability: true if some parameters get the training set perfectly correct
Convergence: if the training is separable, perceptron will eventually converge (binary case)
Mistake Bound: the maximum number of mistakes (binary case) related to the margin or degree of separability.

Problems with the perceptron

Noise: if the data isn't separable, weights might thrash
Mediocre generalization: finds a "barely" separating solution
Overtraining: test/held-out accuracy usually rises, then falls.
Is a kind of overfitting.

MIRA*: fixing the perceptron

$$min_w \frac{1}{2} \sum_y ||w_y - w'_y||^2$$

$$w_{y*} * f(x) \geq w_y * f(x) + 1$$