Machine Learning. Spring 2013. Homework 4.

Due: Thursday 6/June. 2013. (By email, or put hardcopy in my mailbox).

Question 1. Inference Algorithms for Probability Distributions

Consider a Markov Random Field (MRF) defined over a graph $\mathcal{V}, \mathcal{E}$. Binary valued variables $x_i \in \{0, 1\}$ are defined at each node $i \in \mathcal{V}$. The probability distribution is given by:

$$P(\vec{x}) = \frac{1}{Z} \exp\{-\sum_{i \in \mathcal{V}} \theta_i x_i - \sum_{(i,j) \in \mathcal{E}} T_{ij} x_i x_j\}, \tag{1}$$

where $\{\theta_i : i \in \mathcal{V}\}$, $\{T_{ij} : (i, j) \in \mathcal{E}\}$ are constants.

For what types of graphs can dynamic programming (DP) be applied to estimate $\vec{x}^* = \arg\max P(\vec{x})$? Describe DP for these cases.

Question 2. Hidden Markov Models.

A Hidden Markov Model (HMM) has hidden states $q_t \in \{s_1, ..., s_n\}$ and observed states $o_t \in \{v_1, ..., v_m\}$. Define the probability model $P(O|Q)$ for generating a sequence of outputs $O = \{o_1, ..., o_t\}$ from a sequence of inputs $Q = \{q_1, ..., q_t\}$. Define the prior model $P(Q)$. What is the graph structure?

What are the three tasks that an HMM is designed for? Describe algorithms for performing these three tasks. What happens to these tasks if the hidden states are observed?

Question 3. Multi-Class max-margin and Latent SVM.

Describe how structure max-margin (structure SVM) can be applied to learning the parameters for a multi-class problem where the output is of form $\vec{y}^* = \arg\max_{\vec{y}} \lambda \cdot \phi(\vec{d}, \vec{y})$, where $\vec{d}$ is the input and $\vec{y}$ is the output.

Discuss how the complexity of $\lambda \cdot \phi(\vec{d}, \vec{y})$ (e.g. the graph structure) affects the difficulty of the learning. How does this relate to learning the parameters of an exponential regression model $P(\vec{y}|\vec{d}, \lambda) = \frac{1}{Z[\lambda, \vec{x}]} \exp\{\lambda \cdot \phi(\vec{d}, \vec{y})\}$?

Describe how to learn latent SVMs where some output variables $\vec{y}$ are observed,

some variables $\vec{h}$ are hidden/latent, and the input data is $\vec{d}$. The solution is of form: $\vec{y}^*, \vec{h}^* = \arg\max_{\vec{y},\vec{h}} \lambda \cdot \phi(\vec{d}, \vec{y}, \vec{h})$.

How will the learning depend on the complexity of the model and on the initial conditions? How does this relate to the EM algorithm for a distribution $P(\vec{y}, \vec{h}|\vec{x}, \lambda) \propto \exp\{\lambda \cdot \phi(\vec{d}, \vec{y}, \vec{h})\}$?