# Inference and Representation, Fall 2015

## Problem Set 2: Undirected graphical models & Modeling exercise

**Due: Friday, Oct. 2, 2015 at 5pm** (as a PDF document uploaded in NYU Classes.)

**Important:** *See problem set policy on the course web site.*

---

1. Recall that an Ising model is given by the distribution

$$p(x_1, \cdots, x_n) = \frac{1}{Z} \exp\Big( \sum_{(i,j)\in E} w_{i,j} x_i x_j - \sum_{i \in V} u_i x_i \Big), \tag{1}$$

where the random variables $X_i \in \{-1, +1\}$. Related to the Ising model is the *Boltzmann machine*, which is parameterized the same way (i.e., using Eq. 1), but which has variables $X_i \in \{0, 1\}$. Here we get a non-zero contribution to the energy (i.e. the quantity in the parentheses in Eq. 1) from an edge $(i, j)$ only when $X_i = X_j = 1$.

Show that a Boltzmann machine distribution can be rewritten as an Ising model. More specifically, given parameters $\vec{w}, \vec{u}$ corresponding to a Boltzmann machine, specify new parameters $\vec{w}', \vec{u}'$ for an Ising model and prove that they give the same distribution $p(\mathbf{X})$ (assuming the state space $\{0, 1\}$ is mapped to $\{-1, +1\}$).

2. Give a procedure to convert any Markov network on discrete variables into a pairwise Markov random field. In particular, given a distribution $p(\mathbf{X})$, specify a new distribution $p'(\mathbf{X}, \mathbf{Y})$ which is a pairwise MRF, such that $p(\mathbf{x}) = \sum_{\mathbf{y}} p'(\mathbf{x}, \mathbf{y})$, where $\mathbf{Y}$ are any new variables added.

*Clarification:* Assume that the input is specified as full tables specifying the value of the potential for every assignment to the variables for each potential. The new pairwise MRF must have a description which is polynomial in the size of the original MRF.

*Hint*: First consider a simple case, such as a MRF on 3 binary variables with a single potential function for the 3 variables, i.e. $p(\mathbf{X}) \propto \psi_{123}(X_1, X_2, X_3)$. Introduce a new variable $Y$ with $2^3 = 8$ states and let $p'(\mathbf{X}, Y) \propto \psi_Y(Y)\psi_{1Y}(X_1, Y)\psi_{2Y}(X_2, Y)\psi_{3Y}(X_3, Y)$. Figure out how to set the new potential functions $\psi_Y(Y), \psi_{1Y}(X_1, Y), \psi_{2Y}(X_2, Y)$ and $\psi_{3Y}(X_3, Y)$ so as to have $p(\mathbf{x}) = \sum_y p'(\mathbf{x}, y)$ for all assignments $\mathbf{x}$.

3. **Exponential families** (see Chap. 9). Probability distributions in the exponential family have the form:

$$p(\mathbf{x}; \eta) = h(\mathbf{x}) \exp\{\eta \cdot \mathbf{f}(\mathbf{x}) - \ln Z(\eta)\}$$

for some scalar function $h(\mathbf{x})$, vector of functions $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_d(\mathbf{x}))$, canonical parameter vector $\eta \in \mathbb{R}^d$ (often referred to as the *natural parameters*), and $Z(\eta)$ a constant (depending on $\eta$) chosen so that the distribution normalizes.

   (a) Determine which of the following distributions are in the exponential family, exhibiting the $\mathbf{f}(\mathbf{x})$, $Z(\eta)$, and $h(\mathbf{x})$ functions for those that are.

      i. $N(\mu, I)$—multivariate Gaussian with mean vector $\mu$ and identity covariance matrix.

ii. $\mathrm{Dir}(\alpha)$—Dirichlet with parameter vector $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_K)$ (see Sec. 2.5.4).

iii. log-Normal distribution—the distribution of $Y = \exp(X)$, where $X \sim N(0, \sigma^2)$.

iv. Boltzmann distribution—an undirected graphical model $G = (V, E)$ involving a binary random vector $\mathbf{X}$ taking values in $\{0,1\}^n$ with distribution $p(\mathbf{x}) \propto \exp\big\{\sum_i u_i x_i + \sum_{(i,j)\in E} w_{i,j} x_i x_j\big\}$.

(b) *Conditional models.* One can also talk about conditional distributions being in the exponential family, being of the form:

$$p(\mathbf{y} \mid \mathbf{x}; \eta) = h(\mathbf{x}, \mathbf{y}) \exp\{\eta \cdot \mathbf{f}(\mathbf{x}, \mathbf{y}) - \ln Z(\eta, \mathbf{x})\}.$$

The partition function $Z$ now depends on $\mathbf{x}$, the variables that are conditioned on. Let $Y$ be a binary variable whose conditional distribution is specified by the logistic function,

$$p(Y = 1 \mid \mathbf{x}; \alpha) = \frac{1}{1 + e^{-\alpha_0 - \sum_{i=1}^{n} \alpha_i x_i}}$$

Show that this conditional distribution is in the exponential family.

4. **Conjugacy and Bayesian prediction** (generalization of Bernoulli ex. from 9.2.5.5)

(a) Let $\theta \sim \mathrm{Dir}(\alpha)$. Consider discrete random variables $(X_1, X_2, \ldots, X_N)$, where $X_i \sim \mathrm{Cat}(\theta)$ for each $i$ (thus the $X_i$ are conditionally independent of one another given $\theta$). Show that the posterior $p(\theta \mid x_1, \ldots, x_N, \alpha)$ is given by $\mathrm{Dir}(\alpha')$, where

$$\alpha'_k = \alpha_k + \sum_{i=1}^{N} 1[x_i = k].$$

This property, that the posterior distribution $p(\theta \mid \mathbf{x})$ is in the same family as the prior distribution $p(\theta)$, is called *conjugacy.* The Dirichlet distribution (see Sec. 2.5.4) is the *conjugate prior* for the Categorical distribution. Every distribution in the exponential family has a conjugate prior. For example, the conjugate prior for the mean of a Gaussian distribution can be shown to be another Gaussian distribution.

(b) Now consider a random variable $X_{\text{new}} \sim \mathrm{Cat}(\theta)$ that is assumed conditionally independent of $(X_1, X_2, \ldots, X_N)$ given $\theta$. Compute:

$$p(x_{\text{new}} \mid x_1, x_2, \ldots, x_N, \alpha)$$

by integrating over $\theta$.

*Hint*: Your result should take the form of a ratio of gamma functions.

This is called *Bayesian* prediction because we put a prior distribution over the parameters $\theta$ (in this case, a Dirichlet) and are thus able to take into consideration our initial uncertainty over (and prior knowledge of) the parameters together with the evidence we observed (samples $x_1, \ldots, x_N$) when giving our predictions for $x_{\text{new}}$.

5. In this question, you will be using PyMC3, a framework for probabilistic programming. Begin by reading the "Getting started" guide[1] and Chapter 1 of the book *Probabilistic Programming and Bayesian Methods for Hackers*.[2]

   A PyMC3 environment has been set up on CIMS compute nodes (students who did not already have a CIMS account should have received an e-mail on Tuesday to their NYU e-mail address with access information). To use PyMC3 on the CIMS machines (specifically, we recommend using the `crunchy` machines[3]), first run the following command:

   ```
   module load python-2.7
   ```

   Afterward, you can run any Python program using PyMC3 by importing `pymc3`.

   First, run the examples discussed in the above reading, and experiment with what happens if you change the models and parameters (this is simply warm-up; do not hand in).

   This question will involve working with latent variable models. We will provide some data, and you have to model it as a Bayesian network with hidden (unobserved) parameters. You will assign prior distributions over these hidden parameters and use PyMC3's `Metropolis`[4] function that takes the observed data and the probabilistic model and generates samples that estimate the posterior distributions of the latent variables (in this case, the hidden parameters). You can essentially treat `Metropolis` function as a black-box; details will be covered in Lecture 6. The `summary` function in PyMC3 reports the mean, standard deviation and quantiles for all parameters.

   Using the ideas from the following examples,

   - Example 1: Coal mining disasters case study
   - Example 2: Text messages data analysis example
   - Example 3: Example code for arbitrary determinsitics

   analyse the data in `text_data.csv`, a sythetic dataset provided with the assignment, using PyMC3's `Metropolis` function for approximate inference. Unlike in Example 2 above, in this dataset there are *two* switchpoints. Can you find them?

   Find the points where the distribution of text message count changes, and the average rate of text messages in each of these regions with different distribution. Report the results from the `summary` function and briefly describe your findings.

[1] `https://pymc-devs.github.io/pymc3/getting_started/`

[2] `http://nbviewer.ipython.org/github/CamDavidsonPilon/Probabilistic-Programming-and-Bayesian-Methods-for-Hackers/blob/master/Chapter1_Introduction/Chapter1.ipynb` – note, this tutorial uses PyMC2, which has slightly different syntax from PyMC3

[3] `https://www.cims.nyu.edu/webapps/content/systems/resources/computeservers`

[4] One of the examples uses the `Slice` function from PyMC3, which is similar to `Metropolis`, except that it works only for continuous variables.