**EECS E6720: Bayesian Models for Machine Learning**

**Columbia University, Fall 2016**

**Instructor: John Paisley**

**MIDTERM EXAM** (150 points)

# Exam details

- This is a take-home exam. It is open book, but you are not allowed to consult with anyone else on this exam.
- This exam is due by **5:00pm** on **Saturday, October 29, 2016 through Courseworks**.
- This exam counts 150 points (equivalent to 30%) towards your final grade. Late submissions will have **5 points deducted for each minute late**.
- Submission time is non-negotiable. I will only grade your last submission to Courseworks. **Under no circumstances will I accept a late test after 5:30pm.**
- You must submit your answers in a **single PDF file** that is **no more than 5MB** in size. Failure to do so will result in points being deducted.
- Show your work for full credit. Illegible work won't receive full credit. Photographs of your work that don't show up clearly will not receive full credit.

**Question 1. Bayes rule and predictive distributions (25 + 25 points)**
We have observations $x_1, \ldots, x_n$ with each $x \in \{0, 1, 2, \ldots\}$. We choose to model this as $x_i \overset{iid}{\sim} p(x|\pi, r)$, where

$$p(x|\pi, r) = \binom{x + r - 1}{x} \pi^x (1 - \pi)^r.$$

We want to learn $\pi$, so we place a beta prior on it, $\pi \sim \text{Beta}(a, b)$.

a) Calculate the posterior distribution of $\pi$, $p(\pi|x_1, \ldots, x_n)$.

b) What is the predictive distribution of a new $x$? That is, calculate $p(x_{n+1}|x_1, \ldots, x_n)$ under this modeling assumption.

**Question 2. Expectation-maximization algorithm (50 points)**

You are given a data set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, where $x \in \mathbb{R}^d$ and $y \in \mathbb{R}$. You model this using Bayesian linear regression with the following prior structure,

$$y_i \overset{ind}{\sim} \text{Normal}(x_i^T w, \alpha^{-1}), \quad w \sim \text{Normal}(0, \lambda^{-1} I), \quad \lambda \sim \text{Gamma}(a, b).$$

Derive an Expectation-Maximization algorithm for optimizing $\ln p(y, \lambda | x)$ over $\lambda$, where the vector $w$ functions as the variable being integrated out.

Please note the following about what I am looking for:

- It is not necessary to show all work in deriving $q(w)$ using Bayes rule, but it must be clear that you know how Bayes rule is used here, and also what the solution of $q(w)$ is.
- It must be clear that you understand what constitutes the "E" and the "M" steps. Partial credit will be given for correct algorithms without a clear path to the solution.
- You must give pseudo-code for optimizing $\lambda$ including the equations that you would implement in a coding language (similar to the algorithms outlined in the notes).
- If any expectations remain in your final algorithm, you should indicate what they are equal to in the pseudo-code.

**Question 3. Variational inference (50 points)**

You are given a data set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, where $x \in \mathbb{R}^d$ and $y \in \mathbb{R}$. You model this using Bayesian linear regression with the following prior structure,

$$y_i \overset{ind}{\sim} \text{Normal}(x_i^T w, \alpha^{-1}), \quad w \sim \text{Normal}(0, \lambda^{-1} I), \quad \alpha \sim \text{Gamma}(a, b), \quad \lambda \sim \text{Gamma}(e, f).$$

Derive a variational inference algorithm for learning $q(w, \alpha, \lambda) \approx p(w, \alpha, \lambda | y, x)$ using the factorization $q(w, \alpha, \lambda) = q(w)q(\alpha)q(\lambda)$. For your $q$ distributions, use

$$q(w) = \text{Normal}(\mu', \Sigma'), \quad q(\alpha) = \text{Gamma}(a', b'), \quad q(\lambda) = \text{Gamma}(e', f').$$

Again, please note the following about what I am looking for:

- You are free to use the "direct method" from the notes, but the "optimal method" is much easier and faster. The $q$ distributions above are the optimal ones.
- Therefore, you do not need to explicitly calculate the variational objective function to receive full credit ("$\mathcal{L}$" in the notes). But again, you are free to take this approach.
- What I am looking for is an equation-based algorithm (not a gradient-based algorithm) for learning the pairs $(a', b'), (e', f'), (\mu', \Sigma')$. Since it's technically possible to calculate $\mathcal{L}$ and optimize with gradient methods, you will be given partial credit if you do this.
- For full credit, your work must show a clear path to your answer.
- Summarize your algorithm for optimizing $(a', b'), (e', f'), (\mu', \Sigma')$ using pseudo-code similar to how is done in the notes. (This will make your final results easier to find and read for the graders.)
- If any expectations remain in your final algorithm, you should indicate what they are equal to in the pseudo-code.