What to know for midterm #2

1 Asymptotics and connections to non-Bayesian approaches (BDA 4)

1.1 Consistency

- Know the definition of what it means for an estimator to be consistent.
- Know the definition of what it means for a posterior distribution to be consistent.
- Understand the concept of frequentist analysis of Bayesian methods, in which we ask what properties a given Bayesian procedure would have if the data were generated from some true distribution P_0 (which may or may not be a member of the assumed model class).
- Know that Doob's theorem provides very general conditions under which posterior consistency holds, if the model is correct specified and identifiable. (You do not need to know the details.)
- Know that when the model is misspecified, the posterior will typically concentrate at the point θ^* minimizing the KL divergence.

1.2 Asymptotic normality

- Know what it means that the posterior is asymptotically normal, and be able to write down the formula expressing this. In particular, know the mean and covariance matrix of the normal approximation.
- Be able to derive the formula for asymptotic normality of the posterior, from the Taylor approximation (without rigorous details).

- In simple cases with univariate θ, be able to analytically compute the mean and variance of the asymptotic normal approximation, for a given likelihood.
- (Exercise 4 from homework 1) Understand why the posterior on ϕ is (typically) asymptotically normal when $\phi = f(\theta)$, and know how the mean and variance of the asymptotic normal distribution change under this transformation.
- Be able to give an argument for why asymptotic normality of the posterior, plus consistency of the MLE, typically will imply posterior consistency.
- Understand some of the ways in which posterior consistency and asymptotic normality can fail, and be able to give examples.

1.3 Frequentist coverage

- Know the definition of the coverage probability of a confidence region, and have a good intuitive understanding of what it means.
- Understand why having good frequentist coverage is a desirable property.
- In simple cases, be able to analytically compute coverage probability.
- Understand the definition of a posterior credible region (e.g., a 90% or 95% credible region).
- Know the definition of an equal-tailed posterior credible interval.
- In simple cases, be able to analytically compute an equal-tailed posterior credible interval.
- Know that posterior credible regions often (but not always) have good frequentist coverage properties.
- Understand why, if the prior and likelihood are exactly correct, posterior credible regions have frequentist coverage equal to their posterior probability (exercise 15a from homework 1).

2 Model checking and cross-validation (BDA 6 & 7)

2.1 Posterior predictive checking

- Understand the idea behind posterior predictive checks.
- Know how to perform a posterior predictive check and compute a posterior predictive p-value.
- In simple cases, be able to analytically compute a posterior predictive p-value.
- Know the definition of the posterior predictive distribution for replicate data sets, and know how to sample from it based on posterior samples.
- Know how to interpret the results of a posterior predictive check.
- Understand that, ideally, p-values are uniformly distributed, so sometimes we will see p-values close to zero or one simply by chance. If we were to compute a large number of p-values, know how many we would expect to see outside a given range.
- Know that (unfortunately) posterior predictive p-values are not "true" p-values in the sense that they are not uniformly distributed, even if the model is correct.
- Understand why some test statistics/quantities will always be well captured by a given model (and thus are not very informative about model fit), especially in the case of exponential families.
- Realize that one needs to be careful when modifying the model based on the results of posterior predictive checks, since this can lead to overfitting.
- Realize that posterior predictive checks represent a sort of internal consistency check, but that they are not an ideal way of evaluating model fit, because they are "using the data twice".

2.2 Cross-validation

- Understand the idea behind cross-validation—why does it make sense?
- Know the definition of leave-one-out cross-validation.
- Know the definition of k-fold cross-validation.
- Know these common choices of loss function for cross-validation: log posterior predictive, 0-1 loss, square loss.
- Be able to derive the expected loss for a given loss function (taking care to compute it with respect to the true distribution!)
- Know how to compute a cross-validation estimate of generalization performance.
- Understand why cross-validation will typically provide a better assessment of performance compared to posterior predictive checks, since CV is not evaluating the model on the same data that was used to fit the model.
- Understand that if cross-validation is used to choose among multiple models, then in order to assess the performance of the chosen model, it needs to be evaluated on a further held-out set (disjoint from the set of data used for cross-validation).

3 Modeling accounting for data collection (BDA 8)

- Be able to recognize situations in which the data collection process is biased in a way that will affect your inferences.
- Be able to give specific examples of situations in which it is important to model the data collection process.
- Know the definition of ignorability, as well as the intuitive interpretation of it.
- Be able to explain the interpretation of the "potential outcomes" y and the observation indicators I.

- Know the definition of the "complete-data likelihood" in the general setup we considered.
- Know what distribution to use for posterior inferences about θ when ignorability does not hold.
- Be able to derive a formula for this posterior in simple cases.
- Given a verbal description of a distribution on potential outcomes and a data collection process, be able to write down a reasonably appropriate probabilistic model for it.
- Know the definitions of the following conditions: missing at random (MAR), missing completely at random (MCAR), strong ignorability, and distinct parameters.
- Be able to give examples in which these conditions hold or do not hold.
- Know what implications hold between these different conditions.
- Be able to prove that strong ignorability implies ignorability.
- Be able to prove that MAR + distinct parameters implies ignorability.

4 Graphical models (PRML 8)

4.1 Directed graphical models (DGMs)

- Know what the following terms mean: directed graph, directed acyclic graph (DAG), parents of a vertex.
- Understand what it means for a probability distribution to respect a given DAG.
- Be able to write down the factorization implied by a given DAG.
- Be able to draw the DAG corresponding to a given factorization of a distribution.
- In simple cases, be able to determine whether a given distribution respects a given DAG.

- Understand why, for any given probability distribution on $n \geq 2$ variables, there is always more than one DAG respected by the distribution.
- Understand why, for any given DAG, there is more than one probability distribution that respects it.
- Be able to give an example of a DAG that is respected by any distribution on n variables.
- Be able to give an example of a distribution that respects any DAG on n variables.
- Understand that the directionality of the edges in a directed graphical model does not necessarily reflect causality or temporal order!!!

4.2 Undirected graphical models (UGMs)

- Know what the following terms mean: undirected graph, clique, maximal clique.
- Know what it means for a probability distribution to respect a given undirected graph.
- Understand why this is equivalent to saying that the distribution is proportional to a product over maximal cliques.
- Understand why this is equivalent to saying that the distribution is equal to a product over all cliques.
- In simple cases, be able to determine whether a given distribution respects a given undirected graph.
- In simple cases, given a distribution, be able to draw an undirected graph that is respected by the distribution.

4.3 DGMs vs UGMs

- Understand why using a DGM is sometimes preferable to using a UGM, and be able to give an example.

- Understand why using a UGM is sometimes preferable to using a DGM, and be able to give an example.

- Understand that there are some distributions whose conditional independence properties **are not** fully captured by any member of either class of graphical models (directed or undirected).

- Understand that there are some distributions whose conditional independence properties **are** fully captured by a member of each class of graphical models. Be able to give an example and justify it.

- Given a drawing of a DAG G and an undirected graph G', and given the knowledge that p respects G, be able to determine whether this implies that p respects G'.

- Given a drawing of a DAG G and an undirected graph G', and given the knowledge that p respects G', be able to determine whether this implies that p respects G.

4.4 Conditional independence criteria

- Understand that a graphical model only implies independence properties—it does not necessarily imply any dependence properties.

- Understand the separation criterion for undirected graphical models.

- Given a UGM and disjoint subsets of vertices A, B, C, be able to determine whether the UGM implies that $X_A \perp X_B \mid X_C$.

- Understand that if the separation criterion does not apply, then we cannot determine (from the graph alone) whether $X_A \perp X_B \mid X_C$. In particular, this does not imply that the variables are necessarily dependent!

- Know the definition of the moralization (or moral graph) of a DAG.

- Given a DAG, be able to draw the corresponding moral graph.

- Know the meaning of the following terms: set of ancestors, subgraph of ancestors.

- Understand the moral ancestral separation criterion for directed graphical models.

- Given a DGM and disjoint subsets of vertices A, B, C, be able to determine whether the DGM implies that $X_A \perp X_B \mid X_C$.

- Understand that if the moral ancestral separation criterion does not apply, then we cannot determine (from the graph alone) whether $X_A \perp X_B \mid X_C$. In particular, this does not imply that the variables are necessarily dependent!

- Understand the proof of the separation criterion for UGMs (from your homework).

- Understand the proof of the moral ancestral separation criterion for DGMs (from your homework).

5 Hidden Markov Models (PRML 13)

5.1 Setup

- Know the definition of a Markov chain.

- Know the DGM and UGM corresponding to the definition of a Markov chain. Be able to show that the definition of a Markov chain is equivalent to respecting this DGM, and is also equivalent to respecting this UGM.

- Know the definition of a hidden Markov model (HMM) in terms of the factorization of the distribution.

- Know the DGM and UGM corresponding to the factorization definition of a HMM. Be able to show that the factorization definition of a HMM is equivalent to respecting this DGM, and is also equivalent to respecting this UGM.

- Understand the kinds of applications in which HMMs are useful.

- Know the meaning of following terms: hidden state, initial distribution, transition matrix, emission distributions.

- Be able to give examples of HMMs and be able to identify when a given model is an HMM.

5.2 Viterbi algorithm

- Understand the goal of the Viterbi algorithm.
- Be able to give examples of applications in which the Viterbi algorithm would be useful.
- Know the computational complexity of the Viterbi algorithm, and understand why it is advantageous over the naive approach of maximizing over all sequences.
- Understand how to derive the Viterbi algorithm, and be able to derive it from the general principle of writing down the formula for maximizing the probability and looking for recursions. (NOTE: Just memorizing the derivation is not sufficient—you must understand how to derive it so that you can apply the same principles to new situations.)
- Be able to derive both the part for computing the max as well as for computing an argmax.
- Understand why the algorithm is guaranteed to find an argmax (a maximizing sequence).
- Once you have identified the recursions, be able to write down the algorithm in pseudocode.
- Be able to derive the computational complexity of the algorithm, based on the description of the algorithm. (Again, it is insufficient to memorize it, you need to understand how it is derived.)
- Understand the issue of arithmetic underflow/overflow.
- Understand how to fix the issue of arithmetic underflow/overflow using logs.
- Be able to derive the modified algorithm using logs.

5.3 Forward-backward algorithm

- Understand the goal of the forward-backward algorithm (i.e., the forward and backward algorithms, together).
- Understand how the results of the forward and backward algorithms enable you to compute many quantities of interest regarding the conditional distribution on the hidden variables given the observed data.
- Understand how to sample from the conditional distribution on the hidden variables given the observed data, using the results of the forward and backward algorithms.
- Understand how the forward algorithm enables you to predict future data, and be able to write down the posterior predictive distribution in terms of the results from the forward algorithm.
- Understand how to derive both the forward algorithm and the backward algorithm, and be able to derive them from the general principle of writing down the formula for computing the normalization constant and looking for recursions. (NOTE: Again, just memorizing the derivation is not sufficient—you need to understand how to derive it.)
- Once you have identified the recursions, be able to write down the algorithm in pseudocode.
- Be able to derive the computational complexity of the algorithm, based on the description of the algorithm.
- Understand the issue of arithmetic underflow/overflow in the forward and backward algorithms.
- Understand how to derive the log-sum-exp trick.
- Understand why the log-sum-exp trick solves the issue of arithmetic underflow/overflow.