1228 Probabilistic Graphical Models
Prof. J. Boedecker, H. Zhu, and S. Yan
Winter Semester 2025/26

Final exam solutions

1. *Multiple choice (20 points).* Please mark each listed statement as true (✓) or false (×). For each answer that is correctly marked as either true or false you will get *1* point, resulting in *4* points per question. Note, that it can also happen that all answers are true or false.

   (a) Which of the following are true about probabilistic classification problems?

   - ☐ Bayesian classifiers aim at finding the posterior distribution over the class labels given sample features.
   - ☐ All variants of Bayesian classifiers can only accept discrete sample features.
   - ☐ One can integrate expert knowledge into a Bayesian classifier via the prior on class labels.
   - ☐ The classification performance of chain classifiers is not dependent on the order of classes in the chain.

   (b) Which of the following statements are true regarding Metropolis-Hastings (MH) algorithm?

   - ☐ The target distribution has not necessarily to be in the normalized form.
   - ☐ We should have access to the analytical expression of the target distribution when applying MH.
   - ☐ A valid proposal $q(x' \mid x)$ for MH should satisfy $\mathbf{supp}(p^*) \subseteq \cup_x \mathbf{supp}(q(\cdot \mid x))$.
   - ☐ The acceptance probability $A = \min\left\{1, \frac{p^*(x')}{p^*(x)}\right\}$ holds for all types of valid proposal $q(x' \mid x)$.

   (c) Which of the following statements are true about control as probabilistic inference?

   - ☐ Under deterministic environment, the exact inference process formulates a soft version of the Bellman optimality equations.
   - ☐ Under stochastic environment, the optimal policy from the exact inference tends to have a risk-seeking behavior.
   - ☐ The objective of maximum entropy control is to maximize the expected cumulative reward as well as the entropy of the policy.
   - ☐ The optimal policy from maximum entropy control still has a risk-seeking behavior.

   (d) Which of the following statements are true regarding Hamiltonian Monte Carlo methods?

   - ☐ HMC is a purely random-walk-based method that does not use deterministic steps.
   - ☐ The step size in HMC must always remain constant throughout the entire simulation.

- ☐ HMC methods use the gradient of the target distribution to propose new states.
- ☐ HMC can be more efficient than traditional Metropolis-Hastings in high dimensional problems.

(e) Which of the following statements are true regarding Markov decision processes (MDP) and dynamical programming (DP)?

- ☐ In an MDP, the Markov property ensures that the future state depends only on the current state and action.
- ☐ Dynamic programming methods, such as value iteration and policy iteration, require a model of the transition probabilities and rewards.
- ☐ The Bellman equation provides a recursive relationship for the optimal value function in an MDP.
- ☐ Policy iteration is guaranteed to converge to the globally optimal policy in an MDP with finite states and actions.

Solution.

(a) ✓×✓×

(b) ✓×✓×

(c) ✓✓✓×

(d) ××✓✓

(e) ✓✓✓✓

---

2. *Bayesian networks (10 points).* Consider a set of discrete random variables $\{A, B, C, D, E\}$. Their conditional independence relationships and parent-child dependencies are given by:

   - $\mathbf{pa}(A) = \emptyset$,
   - $\mathbf{pa}(B) = \{A\}$,
   - $\mathbf{pa}(C) = \{A\}$,
   - $\mathbf{pa}(D) = \{B, C\}$,
   - $\mathbf{pa}(E) = \{C\}$.

   (a) Draw the *directed acyclic graph* for this Bayesian network. Based on the structure, write down the factorization of the joint probability distribution $\mathbf{P}(A, B, C, D, E)$ using the chain rule.

   (b) Using the ideas of *d-separation*, determine whether the following statements are true or false for any distribution compatible with the graph. Justify your answer by identifying the type of path, *e.g.*, chain, fork, or inverted fork (collider).

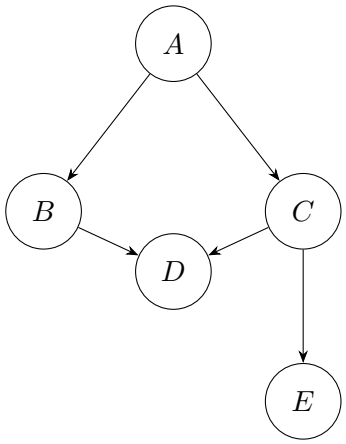
   i. $(B \perp\!\!\!\perp C \mid A)$.

   ii. $(B \perp\!\!\!\perp C \mid D)$.

   (c) For this Bayesian network, what is the Markov blanket of each variable?

Solution.

(a) The directed acyclic graph for the Bayesian network is as follows:



The factorization of the joint probability distribution is given by:

$$P(A, B, C, D, E) = P(A)P(B|A)P(C|A)P(D|B, C)P(E|C).$$

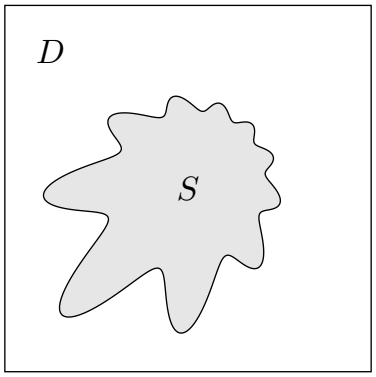
(b) i. True. Fork.

ii. False. Collider.

(c)
- $\mathbf{mb}(A) = \{B, C\}$.
- $\mathbf{mb}(B) = \{A, C, D\}$.
- $\mathbf{mb}(C) = \{A, B, D, E\}$.
- $\mathbf{mb}(D) = \{B, C\}$.
- $\mathbf{mb}(E) = \{C\}$.

---

3. *Area estimation via Monte Carlo methods (10 points).*



Consider two sets $S \subseteq D \subseteq \mathbf{R}^2$ in the 2-dimensional plane as shown in the figure above. Suppose we are given an oracle $f\colon D \to \{0,1\}$ that can determine whether a given point $x \in D$ lies in S, *i.e.*,

$$f(x) = \begin{cases} 1 & x \in S \\ 0 & \text{otherwise.} \end{cases}$$

(a) To estimate the area of S using Monte Carlo methods, is the information provided above sufficient? If not, what else do we need?

(b) Based on your answer to (a), provide a detailed procedure of estimating the area of S via Monte Carlo methods, and express the estimation results mathematically.

Solution.

(a) No, the information provided is not sufficient. In addition to the oracle f, we also need to know the area of set D, denoted as μ. *(2 points.)*

(b) To estimate the area of S, we can use the following Monte Carlo procedure: *(Each step has 2 points.)*

- Randomly sample N points $\{x_i\}_{i=1}^N$ uniformly from the set D.
- For each sampled point x_i, use the oracle f to determine whether it lies in set S.
- Let M be the number of points that fall inside set S, i.e.,

$$M = \sum_{i=1}^{N} f(x_i).$$

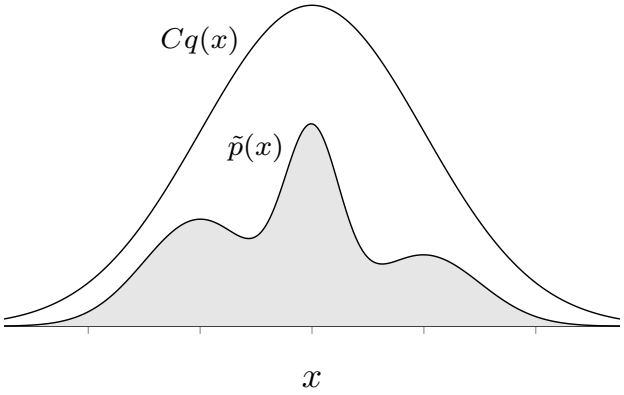
- The area of set S can then be estimated as

$$\mu \frac{M}{N}.$$

As N increases, this estimate will converge to the true area of set S.

---

4. *Rejection sampling (10 points).*



Consider a univariate probability density function $p(x) = \tilde{p}(x)/Z_p$ as shown in the figure above, where $\tilde{p}(x)$ is the unnormalized density and $Z_p = \int \tilde{p}(x)\, dx$ is the unknown normalizing constant. What we only know about p is that we can evaluate $\tilde{p}(x)$ for any given x. To sample from the target distribution p, suppose we are given a proposal distribution q that is easy to sample from, and a constant $C > 0$ such that $Cq(x) \geq \tilde{p}(x)$ for all x.

(a) Describe the rejection sampling algorithm to obtain samples from the unknown distribution p based on the information provided above.

(b) Prove that the samples obtained from rejection sampling are indeed distributed according to the target distribution p.

Solution.

(a) The rejection sampling algorithm proceeds as follows:

- Repeat until we have obtained the desired number of samples:
  - Sample a candidate point $\tilde{x}$ from the proposal distribution q.
  - Sample a uniform random variable u from the interval $[0, Cq(\tilde{x})]$.
  - If $u \leq \tilde{p}(\tilde{x})$, accept $\tilde{x}$ as a sample from the target distribution p; otherwise, reject it and repeat the process.

(5 points)

(b) To prove that the accepted samples are distributed according to p, we consider the probability of accepting a candidate point $\tilde{x}$:

$$\mathbf{P}(\text{accept } \tilde{x}) = \mathbf{P}\left(u \leq \tilde{p}(\tilde{x})\right) = \frac{\tilde{p}(\tilde{x})}{Cq(\tilde{x})}.$$

Therefore, the joint probability density of sampling $\tilde{x}$ and accepting it is given by

$$q(\tilde{x}) \cdot \mathbf{P}(\text{accept } \tilde{x}) = q(\tilde{x}) \cdot \frac{\tilde{p}(\tilde{x})}{Cq(\tilde{x})} = \frac{\tilde{p}(\tilde{x})}{C}.$$

The total probability of acceptance is

$$\int \frac{\tilde{p}(x)}{C}\, dx = Z_p/C.$$

Thus, the conditional probability density of the accepted samples is

$$p(x) = \frac{\tilde{p}(x)/C}{Z_p/C} = \frac{\tilde{p}(x)}{Z_p},$$

which is exactly the target distribution p. *(5 points)*

---

5. *Linear output hidden Markov models (10 points).* Suppose we are given a dataset

$$(x(t), y(t)), \quad t = 1, \ldots, m,$$

where $x(t) \in \mathbf{R}^n$ is the feature vector and $y(t) \in \mathbf{R}$ is the corresponding output, observed from a linear output hidden Markov model with K hidden states. It is assumed that the output $y(t)$ at each time step is generated from a linear measurement model, given by

$$y(t) = \theta_{z(t)}^T x(t) + \epsilon(t),$$

where $z(t) \in \{1, \ldots, K\}$ is the hidden state at time t, $\theta_{z(t)} \in \mathbf{R}^n$ is the coefficient vector associated with state $z(t)$, and $\epsilon(t)$ is standard Gaussian noise.

We would like to estimate the parameters of the model, including the state transition probabilities P and the linear measurement coefficients $\theta_1, \ldots, \theta_K$, based on the observed dataset. We consider a two-step procedure, which first estimates the coefficients θ_i together with the hidden states $z(t)$, so that in the second step the state transition probabilities P can be estimated based on the inferred hidden states.

(a) The first step consists in minimizing the total squared error between the observed outputs and the outputs predicted by the linear output hidden Markov model. For simplicity of notation, we may transform the hidden state labels into one-hot vectors, *i.e.*, let $z(t) \in \{e_1, \ldots, e_K\} \subseteq \mathbf{R}^K$, where e_i is the i-th standard basis vector in $\mathbf{R}^K$, which has 1 in the i-th entry and 0 elsewhere. Then we can formulate the first step of the procedure as an optimization problem, which is given by

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{t=1}^{m} z(t)^T \begin{bmatrix} \left(y(t) - \theta_1^T x(t)\right)^2 \\ \vdots \\ \left(y(t) - \theta_K^T x(t)\right)^2 \end{bmatrix} \\ \text{subject to} & z(t) \in \{e_1, \ldots, e_K\}, \quad t = 1, \ldots, m \end{array} \tag{1}$$

where the optimization variables are $\theta_1, \ldots, \theta_K$ and $z(1), \ldots, z(m)$.

i. What does each term inside the summation of the objective in the problem (1) represent? What is the role of the one-hot vector $z(t)$?

ii. The problem (1) is a combinatorial optimization problem in the two groups of variables $\{\theta_i\}_{i=1}^K$ and $\{z(t)\}_{t=1}^m$ which can be very difficult to solve directly. As a simple heuristic for approximately solving (1), we can split the problem into two subproblems so that we can alternatively optimize over each group of variables while keeping the other group fixed. Based on this idea, please describe an algorithm for approximately solving the optimization problem (1). In particular, you should explain what is the objective of each subproblem in your formulation.
Hint. You might get some inspiration from the *expectation-maximization* algorithm or the *k-means clustering*.

(b) Now suppose we have obtained an estimate of the hidden states $\tilde{z}(t) \in \{1, \ldots, K\}$ for all $t = 1, \ldots, m$. Explain how to estimate the state transition probabilities P based on the inferred hidden states. You can either express your answer in words or mathematically.

Solution.

(a) i. The term inside the summation represents the squared error between the observed output $y(t)$ and the predicted output $\theta_i^T x(t)$ for each possible hidden state i. The one-hot vector $z(t)$ serves as an indicator that selects which hidden state is active at time t, which chooses the corresponding squared error term for that time step. *(2 points.)*

ii. Consider the following algorithm:

- *Initialization.* Start with initial guesses for the coefficients $\theta_1, \ldots, \theta_K$.
- *Repeat until convergence.*
  - *E-step (Estimate hidden states).* For each time step t, assign the hidden state $z(t)$ by minimizing the squared error given the current estimates of θ_i:

$$z(t) \coloneqq e_{i^\star}, \quad i^\star \coloneqq \underset{i=1,\ldots,K}{\operatorname{argmin}} (y(t) - \theta_i^T x(t))^2.$$

  - *M-step (Maximize over the coefficients).* With the current estimates of $z(t)$, update each coefficient vector θ_i by solving a linear regression problem using only the data points assigned to state i:

$$\theta_i \coloneqq \underset{\theta}{\operatorname{argmin}} \sum_{\{t|z(t)=e_i\}} (y(t) - \theta^T x(t))^2.$$

(5 points.)

(b) To estimate the state transition probabilities P, we can count the number of transitions between each pair of states based on the inferred hidden states $\tilde{z}(t)$. Specifically, for each pair of states (i, j), we count how many times a transition from state i to state j occurs in the sequence $\tilde{z}(1), \ldots, \tilde{z}(m)$. The transition probability from state i to state j can then be estimated as

$$P_{ij} = \frac{\text{Number of transitions from state } i \text{ to state } j}{\text{Total number of transitions from state } i}.$$

This gives us the estimated state transition matrix P. *(3 points.)*