CS Machine Learning Qualifying Exam

Georgia Institute of Technology

March 30, 2017

The exam is divided into four areas: Core, Statistical Methods and Models, Learning Theory, and Decision Processes. There are three questions in each area. You must answer *two* out of three questions in the Core area. You must choose two of the remaining areas and answer *two* out of three questions in each one.

1 Core

Question 1. **Bayes Error**

Let (X, Y) be a pair of random variables taking their values in R^d and $\{0, 1\}$, respectively. We are interested in classification, namely predicting Y from X. Let $g^*(X)$ denote the *Bayes classifier* and $L^* = P\{g^*(X) \neq Y\}$ denote the probability of misclassification, otherwise known as the *Bayes error*. The Bayes classifier can be viewed as the output of the *optimal* learning algorithm for (X, Y). Given any other classifier $g(X)$ with associated probability of error L, we have $L^* \leq L$.

(a) Show that any feature transform T which maps X into a transformed random variable $T(X)$ can only increase the Bayes error (meaning that the Bayes error for $T(X)$ will never be less than L^*). Does this result have any consequences for deep learning?

(b) Prove that $L^* \leq \min(p, 1 - p)$, where $p = P(Y = 1)$. Show that equality is attained if X and Y are independent. Construct a distribution where X is *not* independent of Y but $L^* = \min(p, 1 - p)$.

(c) In a paper from 1996, David Wolpert proved the "No Free Lunch" theorem which demonstrated that bias-free learning is futile. Specifically, if one is interested in the off-training-set error, then there is no a priori reason to prefer one learning algorithm over another (see www.no-free-lunch.org). Is there any contradiction between these results and the definition of the Bayes classifier above? Explain.

Question 2. **Naive Bayes Classifier**

Consider the task of predicting whether or not an email is spam based on the words that it contains. Let θ_{cw} be the probability that the word w occurs in emails of class c, where $c = 1$ is spam and $c = 0$ is nonspam. Define the indicator variable x_w which equals 1 if word w occurs in the email and 0 otherwise. Then the probability that the email belongs to class c is:

$$p(\mathbf{x}|c, \theta) = \prod_{w=1}^{W} \theta_{cw}^{x_w}(1 - \theta_{cw})^{1-x_w},$$

where $\mathbf{x} = (x_1, \ldots, x_W)$ is a bit vector and W is the number of words in the vocabulary.

(a) Show that this is a Naive Bayes model.

(b) Show that the class-conditional log-likelihood can be written:

$$\log p(\mathbf{x}|c, \theta) = \psi(\mathbf{x})^T \beta_c,$$

for appropriately chosen vectors $\psi(\mathbf{x})$ and β_c of length $W + 1$.

(c) Assuming that $p(c = 0) = p(c = 1) = 0.5$, write an expression for the posterior odds ratio $\log_2 \frac{p(c=1|\mathbf{x})}{p(c=0|\mathbf{x})}$. What type of classifier is this?

(d) Some words will occur in both spam and non-spam mails and therefore will not be very discriminative. The posterior odds for these words should be 50-50 (i.e. not informative in deciding the class label). For a particular word w, state the conditions on θ_{0w} and θ_{1w} such that the presence or absence of the word in the email will have no effect on the classifier decision from part (c).

Question 3. **Bias-Variance Tradeoff**

(a) What is the bias-variance tradeoff? Suppose we are training a neural network to solve a regression problem. What does the bias-variance tradeoff tell us about the training problem?

(b) Write the bias-variance decomposition of the expected mean squared error $E_D[(f(x|D) - E[y|x])^2]$ for the problem of predicting y given x using a neural network trained on data D. Draw three pictures to illustrate the roles of the bias, variance, and noise in y independent of x.

(c) Use the bias-variance tradeoff to discuss the recently-demonstrated effectiveness of neural-network-based classifiers (in the form of deep models) to solve modern prediction problems in computer vision and speech. To what extent is the observed empirical performance explained and/or contradicted by the classical bias-variance tradeoff?

2 Statistical Methods and Models

Question 1. **Logistic Regression**

Suppose you have two datasets: $D_1 = \{x_i^{(1)}, y_i^{(1)}\}_{i \in 1 \ldots N_1}$ and $D_2 = \{x_j^{(2)}, y_j^{(2)}\}_{j \in 1 \ldots N_2}$, with each $y \in \{-1, 1\}$, and each $x \in \mathbb{R}^P$. The dimension P is identical for D_1 and D_2.

- Let $w^{(1)}$ be the unregularized logistic regression (LR) coefficients from training on dataset D_1, under the model, $P(y \mid x; w) = \sigma(y(x \cdot w))$, with σ indicating the sigmoid function and $x \cdot w$ indicating the dot product of the features x and the coefficients w.
- Let $w^{(2)}$ be the unregularized LR coefficients (same model) from training on dataset D_2.
- Finally, let w^* be the unregularized LR coefficients from training on the combined dataset $D_1 \cup D_2$.

Under these conditions, prove that for any feature n,

$$w_n^* \geq \min(w_n^{(1)}, w_n^{(2)})$$
$$w_n^* \leq \max(w_n^{(1)}, w_n^{(2)}).$$

Question 2. **IO-HMM**

Figure 1 shows a typical input-output hidden Markov model (IO-HMM) that describes how a robot can interact with its environment. The environment states x_t are partially observed through the measurements y_t, and the agent's actions u_t directly affect the state. In this graphical model the actions are *exogenous*, meaning they are determined by factors outside of the system.

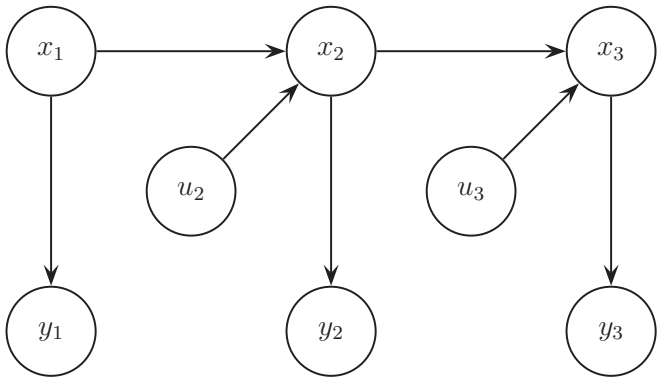

Figure 1: A dynamical system with open-loop control.

How might you learn the parameters of an IO-HMM from dataset D? Assume a "transition model" $p(x_{t+1} \mid x_t, u_{t+1})$, a "measurement model" $p(y_t \mid x_t)$, and a dataset consisting of a length-N sequence of actions and observations, e.g. $\{u_1, y_1, u_2, y_2, \ldots u_N, y_N\}$. First, describe the parameters that must be learned. Then suggest an algorithm that learns these parameters from data and describe how it works. Provide as much detail as you can.

Question 3. **Variational Inference**

Consider a probability distribution P_Φ corresponding to the pairwise Markov network (diamond-shaped) illustrated in Figure 2(a). Consider approximating it with a distribution Q that is represented by the pairwise Markov network of Figure 2(b) (plus-shaped). Derive the potentials $\psi_1(A, C)$ and $\psi_2(B, D)$ that minimize $D(Q\|P_\Phi)$.

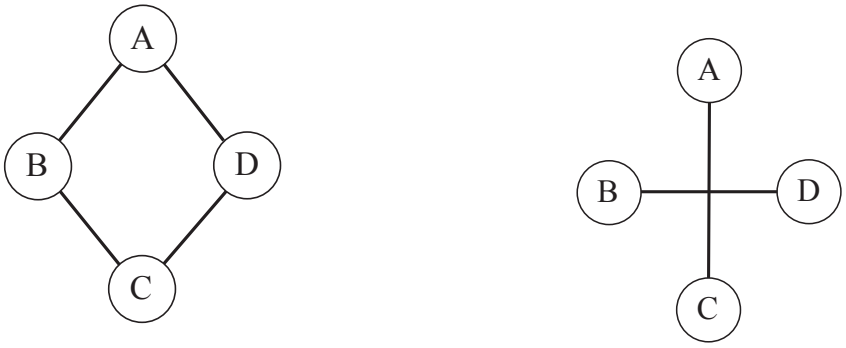

Figure 2: Two pairwise Markov models in the variables A, B, C, D in the shape of (a) *Diamond* (on left) and (b) *Plus* (on right)

3 Learning Theory

Question 1. **PAC learning of simple classes**

(a) Suppose that C is a finite set of functions from X to $\{0, 1\}$. Prove that for any distribution D over X, any target function, and any $\epsilon, \delta > 0$, if we draw a sample S from D of size

$$|S| \geq \frac{1}{\epsilon}\left[\ln(|C|) + \ln\left(\frac{1}{\delta}\right)\right],$$

then with probability $1 - \delta$, all $h \in C$ with $err_D(h) \geq \epsilon$ have $err_S(h) > 0$. Here, $err_D(h)$ is the true error of h under D and $err_S(h)$ is the empirical error of h on the sample S.

(b) Present an algorithm for PAC learning disjunctions over $\{0, 1\}^n$.

Question 2. **Expressivity of LTFs**

Assume that each example x is given by n boolean features (variables). A *decision list* is a function of the form: "if ℓ_1 then b_1, else if ℓ_2 then b_2, else if ℓ_3 then $b_3, \ldots,$ else b_m," where each ℓ_i is a literal (either a variable or its negation) and each $b_i \in \{0, 1\}$. For instance, one possible decision list is the rule: "if $\neg x_1$ then positive, else if x_5 then negative, else positive." Decision lists are a natural representation language in many settings and have also been shown to have a collection of useful theoretical properties.

(a) Show that conjunctions (like $x_1 \wedge \neg x_2 \wedge x_3$) and disjunctions (like $x_1 \vee \neg x_2 \vee x_3$) are special cases of decision lists.

(b) Show that decision lists are a special case of linear threshold functions. That is, any function that can be expressed as a decision list can also be expressed as a linear threshold function "$f(x) = +$ iff $w_1 x_1 + \ldots w_n x_n \geq w_0$," for some values $w_0, w_1, \ldots, w_n$.

Question 3. **VC Dimension and Rademacher Complexity**

(a) What is the VC-dimension d of axis-parallel rectangles in R^3? Specifically, a legal target function is specified by three intervals $[x_{min}, x_{max}]$, $[y_{min}, y_{max}]$, and $[z_{min}, z_{max}]$, and classifies an example (x, y, z) as positive iff $x \in [x_{min}, x_{max}]$, $y \in [y_{min}, y_{max}]$, and $z \in [z_{min}, z_{max}]$.

(b) Explain the importance of VC-dimension for machine learning.

(c) Explain when and why generalization bounds based on the Rademacher Complexity could be better than those based on the VC-dimension.

4 Decision Processes

Question 1. **Reinforcement Learning Methods**

(a) Explain the difference between the SARSA and Q-Learning algorithms. Under what conditions would you prefer one over the other?

(b) For the SARSA algorithm, write the formula for the weight update in the case of a continuous state space and eligibility traces. Explain the meaning of each term in the formula.

(c) For the Q-learning algorithm, show that the Bellman equation in a discrete state space without eligibility traces is given by:

$$Q(x,u) = \sum_{x'} p(x'|x,u)\{r(x,x',u) + \gamma \max_{u'} Q(x',u')\}$$

(d) In many real-world applications, the action space for a robot may be continuous instead of discrete. For example, in autonomous driving the steering angle and acceleration are continuous. How can continuous actions be handled within an RL framework? Describe the strengths and weaknesses of your proposed approach.

(e) What are the advantages of *policy gradient* methods over TD methods? Give one example of a scenario where a policy gradient approach would be attractive.

Question 2. **Markov Reward Process**

Consider an undiscounted Markov reward process with two states A and B. The transition process and reward functions are unknown, but you have observed two episodes:

$(A, +3) \rightarrow (A, +2) \rightarrow (B, -4) \rightarrow (A, +4) \rightarrow (B, -3) \rightarrow$ Terminate
$(B, -2) \rightarrow (A, +3) \rightarrow (B, -3) \rightarrow$ Terminate

In the above episodes, the tuples denote a sample state transition and associated reward at each step. For example, $(A, +2) \rightarrow (B, -4)$ denotes a transition from state A to state B with a reward of $+2$. Answer the following:

- Using the first-visit Monte Carlo method, estimate the value function $V(A), V(B)$
- Using the every-visit Monte Carlo method, estimate the value function $V(A), V(B)$
- Draw a diagram of the Markov reward process that best explains these two episodes from a maximum likelihood perspective (you do not need to prove that you have the ML estimate). Show rewards and transition probabilities on your diagram.
- Solve the Bellman equation to give the true value function $V(A), V(B)$. (Hint: Solve the Bellman equations directly rather than iteratively).
- What value function would batch TD(0) find, if it was applied repeatedly to these two episodes?
- What value function would batch TD(1) find, using accumulated eligibility traces?

Question 3. **Scaling up Reinforcement Learning**

Machine learning algorithms have traditionally had difficulty scaling to large problems. In classification and traditional supervised learning this problem arises with data that exist in very high dimensional spaces or when there are many data points for computing, for example, estimates of conditional densities. In reinforcement learning this is also the case, arising when, for example, there are many, many states or very fine-grained actions spaces.

- Typical approaches to addressing such problems in RL include function approximation and problem decomposition. Compare and contrast these two approaches. What problems of scale do these approaches address? What are their strengths and weaknesses? Are they orthogonal approaches? Can they work well together?
- What are the differences between hierarchical and modular reinforcement learning? Explain both the theoretical and practical limits of these approaches.