15780: GRADUATE AI (SPRING 2018)

Practice Final

May 2, 2018

Topic	Total Score	Score
Social Choice	14	
Probabilistic Modeling	12	
Game Theory	14	
Convex Optimization	12	
Deep Learning	16	
Adversarial Attacks	16	
Integer Programming	16	
Total	100	

1 Social Choice: Strategyproofness [14 points]

Consider the library allocation problem discussed in class, where we pick the location to set up a library. For this problem, we will consider the real plane ($\mathbb{R}^2$) as opposed to the real line ($\mathbb{R}$). Recall that each player has a true preference for the location of the library, which we will refer to as a *peak*.

Assume that the utility function of a player whose peak is $x \in \mathbb{R}^2$ is $-d(x, y)$ for a facility located at y, where d denotes Euclidean distance. Given player peaks $x^1, ..., x^n$, consider the mechanism that locates the library at $(\text{med}\{x_1^i\}, \text{med}\{x_2^i\})$. Prove that this mechanism is strategyproof, i.e., player i cannot increase their utility by reporting a peak that is different from x^i, regardless of the reports of other players.

Note: For simplicity, you can assume that the number of voters n is odd.

2 Probabilistic Modeling: MLE and MAP [12 points]

(a) [4 points] Given a collection of observed (independent) data points $X = \{x^{(1)}, \ldots, x^{(m)}\}$ from a uniform distribution over $[-2\alpha, \alpha]$ (for $\alpha > 0$), derive the maximum likelihood estimator of α, which maximizes the probability of observing X.

(b) [8 points] Given a collection of observed (independent) data points $X = \{x^{(1)}, ...x^{(m)}\}$ from a uniform distribution over $[0, e^{\alpha}]$ where α follows a prior distribution

$$p(\alpha) \propto e^{-\alpha^2},$$

derive the estimator of α that maximizes the posteriori probability $p(\alpha|X)$. **(Hint: use** $p(\alpha|X) \propto p(X|\alpha)p(\alpha)$**).**

3 Game Theory: IESDS [14 points]

One method of simplifying the search for Nash equilibria is through the iterated elimination of strictly dominated strategies (IESDS). We say that a player's pure strategy s_i' is strictly dominated by another pure s_i if $\forall s_{-i} \in S_{-i},\ u_i(s_i', s_{-i}) < u_i(s_i, s_{-i})$. In other words, s_1 dominates s_2 if, no matter what the other players do, player i always does strictly better by playing s_1 rather than s_2.

IESDS proceeds by repeatedly eliminating one strictly dominated strategy per round, until there are no more dominated strategies to eliminate. For example, IESDS on the following game proceeds as follows.

	North	East	South	West
Top	2,3	1,-1	4,0	3,-3
Middle	7,2	-2,0	5,2	6,7
Bottom	8,2	0,1	6,-1	4,0

- Column eliminates East, as playing North is strictly better.
- Row eliminates Top, as playing either Middle or Bottom is strictly better now that Column has eliminated East.
- Column eliminates South, as playing West is strictly better now.
- No more strategies can be eliminated; this leaves Row: [Middle, Bottom] and Column: [North, West] as the surviving strategies.

Prove the following: If IESDS eliminates all but one of the strategies of each player, then there is a unique Nash equilibrium in the game.

Hints:

- Start by proving that IESDS will never remove an action s_i that appears (with nonzero probability) in any Nash equilibrium.
- Conclude by applying Nash's Theorem: In any (finite) game, there exists at least one (possibly mixed) Nash equilibrium.

4 Convex Optimization [12 points]

Recall that we covered two distinct but similar notions of convexity in class: convexity of sets, and convexity of functions. These two definitions are not directly comparable, but we can establish a relationship between them as follows.

(a) [6 points] The level set I_β of a function is the subset of all points in its domain for which the function takes a value at most β i.e., for $f : D \to \mathbb{R}$ with some domain D, $I_\beta = \{x \in D \mid f(x) \leq \beta\}$. Prove that when f is a convex function, for every β, the level set I_β is convex.

(b) [6 points] Find an example where the converse is not true, i.e. a non-convex function for which **for every** β the level set I_β (as defined above) is convex. (A pictoral proof with a brief justification is fine.)

5 Deep Learning: Neural Networks and Boolean Functions [16 points]

In this question, you will explore the representational power of neural networks. We will assume the inputs $x \in \{0,1\}^n$ are binary vectors of length n. We will also use the true binary threshold as the activation function, i.e., $f(z) = 1$ if $z > 0$ and 0 otherwise. We will consider only networks with a 1-unit output layer, and thus the output will be either 0 or 1. We can think of using such a neural network to implement boolean functions.

(a) [8 points] Suppose $n = 2$ i.e. the input is a pair of binary values. Suppose we have a neural network with **no hidden units** and just a single output unit, i.e. $y = f(W^T x + b)$ is the entire network. What should W and b be if we want to implement boolean AND (i.e. $y = 1$ only when $x = (1, 1)$). What about boolean OR? (No justification is needed.)

(b) [8 points] In fact, for any number of input boolean variables, a **single hidden layer** is enough to represent any boolean function. We can use a scheme known as *conjunctive normal form* (CNF) to do this. A formula is in CNF if it is being expressed as an OR over multiple ANDs. The ANDs are defined on the input variables, and are known as *clauses*. For instance, $(x_1 \wedge x_2 \wedge \neg x_3) \vee (\neg x_1 \wedge x_2 \wedge x_3)$ is a valid CNF on the input variables x_1, x_2, x_3.

Any boolean function can be represented by a CNF formula. Describe how to build a network to implement any boolean function in this way.

6 Adversarial Attacks [16 points]

Assume we are given a set of m training points $S = \{(x^{(i)}, y^{(i)}) \in \mathbb{R}^D \times \{-1, +1\} \mid i = 1, \dots m\}$. Consider a monotonically decreasing classification loss $\mathcal{L} : \mathbb{R} \to \mathbb{R}$ and a hypothesis function $h_\theta(x) = \theta^T x$ mapping from $\mathbb{R}^D$ to $\mathbb{R}$ for $\theta \in \mathbb{R}^D$.

For this problem, assume that the training data is such that for every i, the first co-ordinate of $x^{(i)}$ equals its label and all other co-ordinates are zero i.e., $x_1^{(i)} = y^{(i)}$, and $x_j^{(i)} = 0$ for $j > 1$. Consider values θ^a and θ^b of the parameter, that perfectly classify the training data:

$$\theta^a = (1, \overbrace{0, 0, \dots, 0}^{D-1 \text{ zeros}})$$
$$\theta^b = (1, 1, 1, \dots, 1).$$

We can see that for all i, $h_{\theta^a}(x^{(i)}) \cdot y^{(i)} = h_{\theta^b}(x^{(i)}) \cdot y^{(i)} = 1$, leading to perfect classification.

(a) [8 points] **Robustness of θ^a to adversarial attacks.** Consider ϵ such that for every sample i, there exists an adversarial perturbation $\Delta^{(i)}$ satisfying $\|\Delta^{(i)}\|_\infty \leq \epsilon$ and $h_{\theta^a}(x^{(i)} + \Delta^{(i)}) \cdot y^{(i)} \leq 0$? Show that the smallest value ϵ can take is 1.

(b) [8 points] **Robustness of θ^b to adversarial attacks.** Consider ϵ such that for every sample i, there exists an adversarial perturbation $\Delta^{(i)}$ satisfying $\|\Delta^{(i)}\|_\infty \leq \epsilon$ and $h_{\theta^b}(x + \Delta^{(i)}) \cdot y^{(i)} \leq 0$. Show that the smallest value ϵ can take is $1/D$.

7 Integer Programming [16 points]

Consider a linear binary classification setting (i.e. $h_\theta(x) = \theta^T x,\ y \in \{-1, 1\}$) where we would like to minimize a modification of the standard 0/1 loss (i.e. number of mistakes):

$$\underset{\theta \in \mathbb{R}^n}{\text{minimize}} \ \frac{1}{m} \sum_{i=1}^{m} \ell(\theta^T x^{(i)},\ y^{(i)}) \tag{1}$$

where

$$\ell(\theta^T x,\ y) = \mathbf{1}\{y \cdot (\theta^T x) < 1\}.$$

This machine learning problem can be formulated as a mixed integer program. Construct a mixed integer program that is equivalent to Equation (1), and briefly justify why they are equivalent.
Hints:

- Introduce an additional optimization variable $z \in \{0, 1\}^m$.
- Construct a constraint enforcing that for a given θ, z_i is allowed to be 0 only if we have correctly classified example $x^{(i)}$ under θ. Equivalently, your constraint must ensure that when $x^{(i)}$ has been misclassified for a particular θ, then the only feasible value of z_i is 1.
- To implement the previous hint, introduce an arbitrarily large constant M and note that $z_i M = 0$ iff $z_i = 0$. (You do not need to be precise about the definition of M, but you will need to justify why it must be "large enough.")