15780: GRADUATE AI (SPRING 2018)

Practice Final
(Solutions)

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.

Solution: Consider an arbitrary player j whose peak is at x^j. And let $x^1, \ldots, x^{j-1}, x^{j+1}, \ldots x^n$ be arbitrary peaks reported by other players. On truly reporting peak x^j, let the location chosen for the library be $y^\star = (\text{med}\{x_1^i\}, \text{med}\{x_2^i\})$. Note that, the reported value of x_1^j does not affect $y_2^\star$ and the reported value of x_2^j does not affect $y_1^\star$. WLOG, let us consider the effect of misreporting the first coordinate. We have two cases possible:

Case 1: $x_1^j \leq y_1^\star$
If j misreports x_1^j by reporting any value smaller than or equal to $y_1^\star$, then $y^\star$ remains the median. Hence, j's utility does not change. On the other hand, if j reports a value larger than $y_1^\star$, the median of the first coordinate will either remain the same or strictly increase; let the new median be $\hat{y}_1^\star$. The new utility for j is then $-d(x^j, (\hat{y}_1^\star, y_2^\star)) = -\sqrt{(\hat{y}_1^\star - x_1^j)^2 + (y_2^\star - x_2^j)^2} \leq -\sqrt{(y_1^\star - x_1^j)^2 + (y_2^\star - x_2^j)^2} = -d(x^j, y^\star)$, where the inequality holds because $x_1^j \leq y_1^\star \leq \hat{y}_1^\star$. Hence, player j cannot increase their utility in this case.

Case 2: $x_1^j > y_1^\star$
If j misreports x_1^j by reporting any value bigger than or equal $y_1^\star$, then $y^\star$ remains the median. Hence, j's utility does not change. On the other hand, if j reports a value smaller than $y_1^\star$, the median of the first coordinate will either remain the same or strictly decrease; let the new median be $\hat{y}_1^\star$. The new utility for j is then $-d(x^j, (\hat{y}_1^\star, y_2^\star)) = -\sqrt{(x_1^j - \hat{y}_1^\star)^2 + (x_2^j - y_2^\star)^2} \leq -\sqrt{(x_1^j - y_1^\star)^2 + (x_2^j - y_2^\star)^2} = -d(x^j, y^\star)$, where the inequality holds because $x_1^j > y_1^\star \geq \hat{y}_1^\star$. Hence, player j cannot increase their utility in this case.

Therefore, whatever is reported for x_2^j, player j's utility is maximized by truly reporting x_1^j. Similarly, we can show that whatever is reported for x_1^j, player j's utility is maximized by truly reporting x_2^j. Hence, any player j cannot increase their utility by reporting a peak that is different of x^j.

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.

Solution: If any of the data points is not in $[-2\alpha, \alpha]$ (i.e., $\exists x \in X$ s.t. $x \notin [-2\alpha, \alpha]$), then we observe X with 0 likelihood. If all the data points are within $[-2\alpha, \alpha]$, then the likelihood of observing X is $\left(\frac{1}{3\alpha}\right)^m$. To maximize this likelihood, we want to find the minimum α such that all the data points are in $[-2\alpha, \alpha]$. All the data points are in $[-2\alpha, \alpha]$ if and only if $\alpha \geq \frac{-\min(X)}{2}$ ($\Leftrightarrow -2\alpha \leq \min(X)$) and $\alpha \geq \max(X)$. Thus,

$$\mathcal{L}(\alpha) = \begin{cases} \left(\frac{1}{3\alpha}\right)^m & \text{if } \alpha \geq \max(\max(X), \frac{-\min(X)}{2}) \\ 0 & \text{otherwise.} \end{cases}$$

The maximum likelihood estimator of α is $\hat{\alpha} = \max(\max(X), \frac{-\min(X)}{2})$ because for $\alpha < \hat{\alpha}$, $\mathcal{L}(\alpha) = 0 < \mathcal{L}(\hat{\alpha})$. Furthermore, for $\alpha > \hat{\alpha}$, $\mathcal{L}(\alpha) = \left(\frac{1}{3\alpha}\right)^m < \mathcal{L}(\hat{\alpha}) = \left(\frac{1}{3\hat{\alpha}}\right)^m$.

(b) [8 points] Given a collection of observed (independent) data points $X = \{x^{(1)}, \ldots x^{(m)}\}$ 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** $p(\alpha|X) \propto p(X|\alpha)p(\alpha)$**).**

Solution: If any of the data points is not in $[0, e^{\alpha}]$ (i.e., $\exists x \in X$ s.t. $x \notin [0, e^{\alpha}]$), then $p(X|\alpha) = 0$, and from the hint, $p(\alpha|X) = 0$. If all the data points are within $[0, e^{\alpha}]$, then

$$p(\alpha|X) \propto p(X|\alpha)p(\alpha) \propto \left(\frac{1}{e^{\alpha}}\right)^m e^{-\alpha^2} = e^{-\alpha^2 - m\alpha}. \tag{1}$$

Since $\exp(x)$ is a monotonically increasing function, we want to find α maximizing $f(\alpha) = -\alpha^2 - m\alpha$, while keeping all the data points in $[0, e^{\alpha}]$. Note that $f(\alpha)$ is a quadratic function maximized at $\alpha = -m/2$. If $\max(X) \leq e^{-m/2}$, then $\alpha = -m/2$ maximizes $f(\alpha)$ while satisfying $\forall x \in X$, $x \in [0, e^{\alpha}]$. If $\max(X) > e^{-m/2}$, then it means that $\alpha = -m/2$ is too small and we need to find a larger α. Since for $\alpha > -m/2$, $f'(\alpha) = -2\alpha - m < 0$ (i.e., f is strictly decreasing), $\hat{\alpha} = \log(\max(X))$ maximizes $f(\alpha)$ while satisfying $\forall x \in X$, $x \in [0, e^{\alpha}]$. For any $\alpha > \hat{\alpha}$, we would have $f(\alpha) < f(\alpha')$ and hence a smaller posterior probability. Hence,

$$\hat{\alpha} = \begin{cases} -m/2, & \text{if } \max(X) \leq e^{-m/2} \\ \log(\max(X)), & \text{otherwise,} \end{cases}$$

is the estimator of α maximizing $p(\alpha|X)$.

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.

Solution: First, we prove that iterative elimination of strictly dominated strategies never removes an action that is in the support of any Nash equilibrium. Assume for the sake of contradiction that an action s_i^* in a Nash equilibrium s^* is removed, and, furthermore, that it is the first action in any Nash equilibrium that is removed. This means that there exists some s_i such that $u_i(s_i, s'_{-i}) > u_i(s_i^*, s'_{-i}),\ \forall s'_{-i} \in S_{-i}$ at this time. We know that because s_i^* is the first action in any Nash equilibrium to be removed, the rest of s^* has not been removed, so $s^*_{-i} \in S_{-i}$. Then, $u_i(s_i, s^*_{-i}) > u_i(s_i^*, s^*_{-i})$, which contradicts our assumption that s^* is a Nash equilibrium.

Now, by the hint and Nash's theorem, we are done: because there must exist at least one Nash equilibrium, and because IESDS never removes an action that is in the support of any Nash equilibrium, then this must be the unique Nash equilibrium in the game.

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.

Solution: Consider any $x, y \in I_\beta$. Any point on the line between x and y is expressible as $(1-t)x+ty$ for some $t \in [0, 1]$. We need to show that $(1-t)x+ty \in I_\beta$, or equivalently, $f((1-t)x+ty) \leq \beta$. By convexity of f, we have

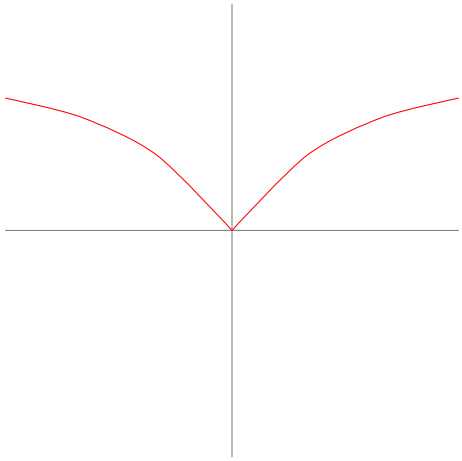
$$f\left((1-t)x+ty\right) \leq (1-t)f(x)+tf(y)$$

And because $x, y \in I_\beta$, we have

$$(1-t)f(x)+tf(y) \leq (1-t)\beta+t\beta = \beta.$$

(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.)

Solution: A function like this would qualify:



The function is not convex: Take two distinct points on the function with strictly positive x-values. The line joining them lies below the function.

The level sets are convex: For any $\beta > 0$, the corresponding level set $\{f(x) \leq \beta\}$ is just an interval in $\mathbb{R}$. For $\beta = 0$, the corresponding level set is just $\{0\}$. For $\beta < 0$, the corresponding level set is $\emptyset$. In all cases, these are convex sets.

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.)

Solution: For AND, $W = (1, 1), b = -1$. For OR, $W = (1, 1), b = 0$.

(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.

Solution: To extend the work in part (a), we can use a unit to do either do an AND over multiple variables, or an OR over multiple variables. The OR over all variables is simply the sum of the variables (when the variables correspond to positive literals) with bias 0. The AND is simply summing up all variables and the bias is the negative of one less than the number of variables. To deal with negative literals, we would add $(1 - x_i)$ for every negative literal which means the coefficient is -1 rather than 1 for the variable and additionally a 1 is added to the bias. Every function can be represented as a CNF formula, so we assume the input is in CNF. Then, the hidden units implement the clauses of the CNF with a multivariable AND over all of the relevant input literals. The output unit is a multivariable OR over the hidden units.

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.

Solution:

First, we show that ϵ can take the value 1. For any i, consider $\Delta^{(i)}$ such that its first co-ordinate equals $-y^{(i)}$ and all other co-ordinates are zero. Clearly, $\|\Delta^{(i)}\|_\infty = 1$. Furthermore, $h_{\theta^a}(x^{(i)} + \Delta^{(i)}) \cdot y^{(i)} = (\theta^a \cdot x^{(i)}) \cdot y^{(i)} + (\theta^a \cdot \Delta^{(i)}) \cdot y^{(i)} = 1 + \Delta_1^{(i)} \cdot y^{(i)} = 1 - (y^{(i)})^2 = 0$.

Next, we show that this is the smallest value of ϵ possible. In particular, if $\epsilon < 1$, we have

$$\begin{aligned} h_{\theta^a}(x^{(i)} + \Delta^{(i)}) \cdot y^{(i)} &= (\theta^a \cdot x^{(i)}) \cdot y^{(i)} + (\theta^a \cdot \Delta^{(i)}) \cdot y^{(i)} \\ &= 1 + \Delta_1^{(i)} \cdot y^{(i)} \\ &\geq 1 - |\Delta_1^{(i)}| \\ &\geq 1 - \epsilon > 0. \end{aligned}$$

Hence, $\epsilon = 1$ is the smallest value possible.

(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 $1/D$.

Solution:

First, we show that ϵ can take the value $1/D$. For any i, consider $\Delta^{(i)}$ such that all its co-ordinates equal $-\frac{y^{(i)}}{D}$. Clearly, $\|\Delta^{(i)}\|_\infty = \frac{1}{D}$. Furthermore, $h_{\theta^b}(x^{(i)} + \Delta^{(i)}) \cdot y^{(i)} = (\theta^b \cdot x^{(i)}) \cdot y^{(i)} + (\theta^b \cdot \Delta^{(i)}) \cdot y^{(i)} = 1 - \sum_{j=1}^{D} \frac{1}{D}(y^{(i)})^2 = 0$.

Next, we show that this is the smallest value of ϵ possible. In particular, if $\epsilon < \frac{1}{D}$, we have that

$$\begin{aligned} h_{\theta^b}(x^{(i)} + \Delta^{(i)}) \cdot y^{(i)} &= (\theta^b \cdot x^{(i)}) \cdot y^{(i)} + (\theta^b \cdot \Delta^{(i)}) \cdot y^{(i)} \\ &= 1 + y^{(i)} \sum_{j=1}^{D} \Delta_j^{(i)} \theta_j^b \\ &\geq 1 - \max_j \left|\Delta_j^{(i)}\right| \sum_{j=1}^{D} \left|\theta_j^b\right| \\ &\geq 1 - D\epsilon > 0. \end{aligned}$$

Hence, $\epsilon = \frac{1}{D}$ is the smallest value possible.

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):

$$\underset{\theta \in \mathbb{R}^n}{\text{minimize}} \ \frac{1}{m} \sum_{i=1}^{m} \ell(\theta^T x^{(i)},\ y^{(i)}) \tag{2}$$

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 (2), 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.")

Solution: The integer program can be formulated as follows:

$$\begin{aligned} \underset{\theta \in \mathbb{R}^n, z \in \{0,1\}^m}{\text{minimize}} \quad & \frac{1}{m} \sum_{i=1}^{m} z_i \\ \text{subject to} \quad & y^{(i)}(\theta^T x^{(i)}) \geq 1 - z_i M. \end{aligned}$$

Constraint: If $z_i = 0$, then it must be that $y^{(i)}(\theta^T x^{(i)}) \geq 1$ is satisfied, i.e., we need to classify the example correctly. But if $z_i = 1$, then we need not correctly classify the example, because we choose M large enough so that the inequality is satisfied no matter the value of $y^{(i)}(\theta^T x^{(i)})$. Thus, our constraint enforces the condition given in the second hint.

The sum of the z_i's is then an upper bound on the number of mistakes. Since at the optimum, we will set $z_i = 0$ whenever possible, minimizing this sum is exactly equivalent to minimizing the number of classification mistakes, i.e., the 0/1 loss.

Note (for fun): The modified 0/1 loss formulation presented is equivalent to the standard 0/1 loss formulation $\ell_{0/1}(\theta^T x,\ y) = \mathbf{1}\{y \cdot (\theta^T x) \leq 0\}$. To see why this is equivalent, first note that if we have perfect classification under $\ell_{0/1}$, then $y^{(i)}(\theta^T x^{(i)}) > 0$ (strictly greater than) by definition of the 0/1 loss. Therefore, we could scale θ to some θ' that satisfies $y^{(i)}(\theta'^T x^{(i)}) \geq 1$, matching the definition of our modified 0/1 loss.