15780: GRADUATE AI (SPRING 2018)

Practice Midterm 1
(Solutions)

March 1, 2018

Heuristic Search	20	
Learning Theory	25	
Optimization and ML	30	
Linear Programming	25	
Total	100	

1 Heuristic Search (20 points)

Consider the following search problem. There is a set of *operations* $O = \{o_1, \ldots, o_n\}$, and a set of *conditions* $C = \{C_1, \ldots, C_m\}$. Each operation $o_i \in O$ has a set of *preconditions* $P_i \subseteq C$, and a set of *effects* $E_i \subseteq C$. A state is defined by a subset of conditions $S \subseteq C$. An operation $o_i \in O$ can be applied at state S if and only if $P_i \subseteq S$, and it leads to the state $S \cup E_i$. The goal state is C, i.e., the state that contains all conditions. The initial state is the empty set (so initially only operations o_i that have an empty P_i can be applied).

We define the following heuristic function h for this search problem. Given a state S, $h(S)$ computes the optimal path to the goal state, in the modified problem where every operation o_i is replaced with the operation o'_i, which has the same set of effects E_i, but an empty set of preconditions. Informally, any of the "old" operations can be applied at any state. (The perceptive student may have noticed that computing $h(S)$ is equivalent to solving the Minimum Set Cover problem, that is, computing $h(S)$ happens to be computationally hard, so this is a pretty bad heuristic.)

Prove that A* graph search with the heuristic h is optimal (it always finds the shortest sequence of operations that leads to the goal state). You may rely on any theorem stated in class.

Solution: We know from class that A* graph search with a heuristic h is optimal if h is consistent. We will now prove that h is consistent; i.e., that $h(x) \leq h(y) + c(x, y)$.

Let $H(x, y)$ be the minimum number of moves necessary to get from x to y under the conditions of the heuristic function. Note that $H(x, t) = h(x)$. Because h is a relaxation of the search problem, we know that $H(x, y) \leq c(x, y)$.

If we consider any x and y, we can see that $h(x)$ is at most the number of moves under h to get from x to y plus the number of moves under h to get from y to t; in other words,

$$h(x) \leq H(x, y) + h(y).$$

However, we also know that $H(x, y) \leq c(x, y)$, which means

$$h(x) \leq c(x, y) + h(y),$$

as desired.

2 Learning Theory (25 points)

Q1. (10 pt) For a finite function class F, show that VC-dim$(F) \leq \log_2(|F|)$.

Solution: To shatter a set d points, F needs at least 2^d classes. Therefore, that is, $|F| \geq 2^d$.

Q2. (5 pt) Give an example of an input space X and a function class F such that VC-dim$(F) = \log_2(|F|)$.

Solution: $X = \{1\}$, F contains two functions, one that labels 1 positive, and one that labels 1 negative.

Q3. (10 pt) Give an example of an input space X and two function classes F_1 and F_2 such that VC-dim$(F_i) = 0$ for $i = 1, 2$, but VC-dim$(F_1 \cup F_2) = 1$.

Solution: $X = \{1\}$, F_1 contains only the function that labels 1 positive, and F_2 contains only the function that labels 1 negative.

3 Optimization and ML (30 points)

Q1. (10 pt) Consider the regression problem of minimizing the sum of absolute losses using a linear hypothesis function, that is

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

where $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_+$ is given by $\ell(\hat{y}, y) = |\hat{y} - y|$ and $h_\theta(x) = \theta^T x$. Show that this is a convex optimization problem in θ.

Solution: By the definition of convexity, we can show that $|\theta^T x - y|$ is convex. Specifically, note that given some $\theta_1, \theta_2 \in \mathbb{R}^n$ and $\alpha \in [0, 1]$

$$|(\alpha\theta_1 + (1-\alpha)\theta_2)^T x - y| = |\alpha\theta_1^T x - \alpha y + (1-\alpha)\theta_2^T x - (1-\alpha)y| \le \alpha|\theta_1^T x - y| + (1-\alpha)|\theta_2^T x - y| \tag{2}$$

Since a sum of convex functions is convex, we therefore know that the optimization problem is convex.

Q2. (10 pt) Prove that we can find the solution of the absolute loss linear regression problem by solving the following linear program

$$\begin{aligned} \underset{\theta \in \mathbb{R}^n, z \in \mathbb{R}^m}{\text{minimize}} \quad & \sum_{i=1}^{m} z_i \\ \text{subject to} \quad & -z_i \le \theta^T x^{(i)} - y^{(i)} \le z_i \end{aligned} \tag{3}$$

Solution: First note that the constraint $-z_i \le \theta^T x^{(i)} - y^{(i)} \le z_i$ is equivalent to the constraint that $|\theta^T x^{(i)} - y^{(i)}| \le z_i$, so the sum of the z_i terms are an *upper bound* on the sum of absolute losses. Second, note that if we had $|\theta^T x^{(i)} - y^{(i)}| < z_i$ (strictly less than) at any solution point, we could simply instead choose $z_i = |\theta^T x^{(i)} - y^{(i)}|$ and obtain a solution that still satisfies the constraints while having strictly lower objective value. Thus, at the optimal solution we know that we must have $z_i = |\theta^T x^{(i)} - y^{(i)}|$, meaning the optimization problem has minimized the sum of absolute losses, which is precisely the problem stated above.

Q3. (10 pt) Prove that we can find the linear classifier that minimizes 0/1 loss using the following *binary integer* programming problem, for a large enough value of M.

$$\begin{aligned} \underset{\theta\in\mathbb{R}^n, z\in\{0,1\}^m}{\text{minimize}} \quad & \sum_{i=1}^{m} z_i \\ \text{subject to} \quad & y^{(i)}\theta^T x^{(i)} \geq 1 - z_i M \end{aligned} \tag{4}$$

Solution: First note that if we have perfect classification, then $y^{(i)}\theta^T x^{(i)} > 0$ (strictly greater than) by definition of the 0/1 loss. Therefore, we could scale θ to also satisfy $y^{(i)}\theta^T x^{(i)} \geq 1$. If $z_i = 0$, then this inequality has to be 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 $y^{(i)}\theta^T x^{(i)}$. Because we are minimizing the sum of the z_i terms in the objective, this is exactly equivalent to minimizing the number of classification mistakes, i.e., the 0/1 loss.

4 Linear Programming (25 points)

4.1 Standard Form (10 points)

Recall that a linear program is in the *standard form* if it is expressed as follows:

$$\begin{aligned} &\text{minimize } c^T x \\ &\text{subject to } Ax = b \\ &\qquad x \geq 0 \end{aligned}$$

with optimization variable $x \in \mathbb{R}^n$, and problem data $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$.
Convert the following problem to the standard form:

$$\begin{aligned} &\text{maximize } x_1 + 2x_2 \\ &\text{subject to } x_1 + 3x_2 \leq 12 \\ &\qquad -2x_1 - x_2 \geq -8 \\ &\qquad 1 \leq x_1 \\ &\qquad 0 \leq x_2 \leq 4. \end{aligned}$$

Specifically, what is c, A, and b in the converted problem?

Solution: The converted problem is as follows:

$$\begin{aligned} &\text{minimize } -x_1 - 2x_2 \\ &\text{subject to } x_1 + 3x_2 + x_3 = 12 \\ &\qquad 2x_1 + x_2 + x_4 = 8 \\ &\qquad -x_1 + x_5 = -1 \\ &\qquad x_2 + x_6 = 4 \\ &\qquad x_1, x_2, x_3, x_4, x_5, x_6 \geq 0 \end{aligned}$$

Thus,

$$c = \begin{bmatrix} -1 & -2 & 0 & 0 & 0 & 0 \end{bmatrix}, \; A = \begin{bmatrix} 1 & 3 & 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}, \text{ and } b = \begin{bmatrix} 12 & 8 & -1 & 4 \end{bmatrix}.$$

4.2 Simplex Algorithm (15 points)

The following is **part** of the simplex algorithm for solving a linear program in the standard form:

Repeat:

1. Given index set $\mathcal{J}$ such that $x_{\mathcal{J}} = A_{\mathcal{J}}^{-1} b \geq 0$.
2. Find $j \notin \mathcal{J}$ for which $\bar{c}_j = c_j - c_{\mathcal{J}}^T A_{\mathcal{J}}^{-1} A_j < 0$.
3. Compute step direction $d_{\mathcal{J}} = -A_{\mathcal{J}}^{-1} A_j$ and determine index to remove
$$i^\star = ?$$
4. Update index set: $\mathcal{J} \leftarrow \mathcal{J} - \{i^\star\} \cup \{j\}$.

Choose **one** correct answer for each of the following statements:

Q1. (5 pt) In the second step of the algorithm, no $j \notin \mathcal{J}$ satisfies $c_j - c_{\mathcal{J}}^T A_{\mathcal{J}}^{-1} A_j < 0$.

This means [① a solution is found, ② the problem is infeasible, ③ the problem is unbounded].

Solution: ① a solution is found.

Q2. (5 pt) In the third step of the algorithm, $i^\star$ should be set to

$$\left[① \arg\min_{i \in \mathcal{J}: d_i < 0} x_i/d_i,\ ② \arg\max_{i \in \mathcal{J}: d_i < 0} x_i/d_i,\ ③ \arg\min_{i \in \mathcal{J}: d_i \geq 0} x_i/d_i,\ ④ \arg\max_{i \in \mathcal{J}: d_i \geq 0} x_i/d_i \right].$$

Solution: ② $\arg\max_{i \in \mathcal{J}: d_i < 0} x_i/d_i$.

Q3. (5 pt) In the third step of the algorithm, every $i \in \mathcal{J}$ satisfies $d_i \geq 0$.

This means [① a solution is found, ② the problem is infeasible, ③ the problem is unbounded].

Solution: ③ the problem is unbounded.