Homework 6 — Convex programs and generalization theory

By the due date (midnight on Wed Feb 28), upload the PDF of your typewritten solutions to gradescope.

1. We are given a set of $m$ equations in $n$ unknowns $x_1, \ldots, x_n$:

\[
\begin{align*}
    a_{11}x_1 + \cdots + a_{1n}x_n &= b_1 \\
    a_{21}x_1 + \cdots + a_{2n}x_n &= b_2 \\
    &\vdots \\
    a_{m1}x_1 + \cdots + a_{mn}x_n &= b_m
\end{align*}
\]

It might not be possible to satisfy all these equations exactly; what we want is to find a solution $x = (x_1, \ldots, x_n)$ such that the maximum deviation

\[
\max_{1 \leq i \leq m} \left| b_i - \sum_{j=1}^n a_{ij}x_j \right|
\]

is as small as possible. Write this as a linear program.

2. A halfspace in $\mathbb{R}^d$ is specified by a vector $w \in \mathbb{R}^d$ and an offset $b \in \mathbb{R}$, and is defined as $\{x : w \cdot x \leq b\}$.

   (a) Now suppose we have a collection of halfspaces, given by $w_1, w_2, \ldots$ and $b_1, b_2, \ldots$, respectively. There might be infinitely many of them. Show that their intersection is a convex set.

   (b) Can you express the unit ball $\{x \in \mathbb{R}^d : \|x\|_2 \leq 1\}$ as the intersection of infinitely many halfspaces?

3. We are given two polyhedra $P_1, P_2 \subseteq \mathbb{R}^d$, each specified as the intersection of finitely many halfspaces. We would like to find the distance between these two bodies: the smallest possible value $\|x_1 - x_2\|$, where $x_1 \in P_1$ and $x_2 \in P_2$. Show how to express this as a convex program.

4. Let $\mathcal{X} = \{0,1\}^d$. The class $\mathcal{H}$ of monotone disjunctions consists of classifiers $h$ that are given by a disjunction (logical OR) of some subset of the $d$ features. For instance, the classifier

\[
h(x) = x_1 \lor x_3 \lor x_8
\]

assigns label 1 to points $x \in \mathcal{X}$ for which any of the features $x_1, x_3, x_8$ are set; and assigns label 0 otherwise. Suppose we obtain a training set of $n$ points, drawn i.i.d. from an unknown underlying distribution, and we find a monotone disjunction $h \in \mathcal{H}$ that is correct on all $n$ points. We would like to give a bound on the true error of $h$.

   (a) What is $|\mathcal{H}|$? Your answer should be a function of $d$.

   (b) Give a bound on the true error of $h$ that holds with probability at least $1 - \delta$ over the choice of training data.
(c) What bound could you give if instead we looked at the smaller class \(H_k \subset H\) of \(k\)-sparse monotone disjunctions: that is, monotone disjunctions consisting of at least 1 and at most \(k\) variables?

5. Estimating the bias of a coin. A coin of bias 3/4 is tossed 300 times and an empirical estimate \(\hat{p}\) of the bias is obtained. Use the central limit theorem to come up with an interval in which \(\hat{p}\) will lie, with 95% probability.

6. Determine the VC dimension of the following concept classes. Justify your answers.
   (a) Intervals on the line. \(X = \mathbb{R}\) and \(H = \{h_{a,b} : a, b \in \mathbb{R}, a < b\}\) where \(h_{a,b}(x) = 1(a \leq x \leq b)\).
   (b) Axis-aligned rectangles in the plane. Each \(h \in H\) is given by an axis-aligned rectangle in \(\mathbb{R}^2\), where points inside the rectangle are labeled 1, and points outside are labeled 0.

7. Isotonic regression. In a line fitting problem, we have a data set consisting of pairs \((x_1, y_1), \ldots, (x_n, y_n) \in \mathbb{R}^2\) and we want to draw a line through them. More precisely, we want to find parameters \(a, b \in \mathbb{R}\) such that \(f(x) = ax + b\) passes as close as possible to the points. We have already seen a least-squares formulation of this.

In isotonic regression, we allow a more general function \(f\). It doesn’t have to be a line: it just needs to be monotonically increasing, that is, \(f(x) \geq f(x')\) whenever \(x \geq x'\).

(a) Here is a training set of six points \((x_i, y_i)\):
   
   \[
   (4, 20), \ (2, 5), \ (5, 9), \ (3, 7), \ (1, 10), \ (6, 12).
   \]

   Plot these points, and sketch a function \(f: \mathbb{R} \rightarrow \mathbb{R}\) that is monotonically increasing and that passes through as many of these points as possible.

Let’s sort the data points so that \(x_1 \leq x_2 \leq \cdots \leq x_n\). Monotonicity then means

\[
f(x_1) \leq f(x_2) \leq \cdots \leq f(x_n).
\]

In fact, we can choose any \(f(x_i)\) values that meet this requirement; and we can fill in the rest of the \(f\)-curve by, say, linearly interpolating between these points.

How shall we evaluate candidate functions \(f\)? In part (a), we used the loss function

\[
L_o(f) = \# \text{ of training points that } f \text{ does not pass through}.
\]

Finding the optimal such \(f\) is called the longest increasing subsequence problem in computer science, and can be solved efficiently. However, we typically prefer to use a different, least-squares loss.

Here is a least-squares formulation of our problem: given \(x_1 \leq x_2 \leq \cdots \leq x_n\) and corresponding values \(y_1, \ldots, y_n\), find \(f_1, f_2, \ldots, f_n \in \mathbb{R}\) such that \(f_1 \leq f_2 \leq \cdots \leq f_n\) and such that the squared loss

\[
L(f) = \sum_{i=1}^{n}(y_i - f_i)^2
\]

is minimized. (Here \(f_i\) corresponds to \(f(x_i)\).)

(b) Show that this is a convex optimization problem.

An elegant approach to solving this problem is the pool adjacent violators algorithm. It starts by simply setting \(f_i = y_i\) for all \(i\), and then repeatedly fixes any monotonicity violations: any time it finds \(f_i > f_{i+1}\), it resets both of them to the average of \(f_i, f_{i+1}\) and merges points \(x_i\) and \(x_{i+1}\).

Here is the algorithm, given a set of \(x\) values and their corresponding \(y(x)\).
• Let $S$ be the sorted list of $x$-values

• For all $x$ in $S$:
  – Set $f(x) = y(x)$
  – Assign weight $w(x) = 1$

• While there adjacent values $x < x'$ in $S$ with $f(x) > f(x')$:
  – Remove $x'$ from $S$ and set a pointer from it to $x$
  – Let $f(x) = \frac{w(x)f(x) + w(x')f(x')}{w(x) + w(x')}$
  – Let $w(x) = w(x) + w(x')$

At the end, each of the original $x$-points either lies in the list $S$, in which case it receives value $f(x)$, or leads to some $\tilde{x}$ in list $S$ by following pointers, in which case it receives value $f(\tilde{x})$.

(c) Run this algorithm on the small data set of six points from part (a). What values of $f$ does it yield?