TTIC 31290: Machine Learning for Algorithm Design (Fall 2025)
Avrim Blum and Dravyansh Sharma

Lecture 10: 10/30/25 Lecturer: Dravyansh Sharma

Outline for today

- Graph-based Semi-supervised Learning
- Linear Regression

1 Graph-based Semi-supervised Learning

Recall that in our PAC learning setup, we had access to labels for all the examples (in both realizable and agnostic settings) in our "training set". Perhaps a more natural learning setting is one where we have access to some labeled examples, but also a much larger collection of unlabeled examples. For example, a child learning the concept of a "dog" perhaps initially sees some "labeled" examples in a book or real life, but also (especially in Chicago!) sees a lot of "unlabeled" examples. For another example, think of an email server trying to label emails as spam or not spam. For the large number of emails generated each day, some might be explicitly labeled by the users, but most are left unlabeled.

But how can we learn anything from unlabeled examples? Well, some unlabeled examples may be "similar" to some labeled or other unlabeled examples. Since similarity is a pairwise relation, it is natural to think of it as given by a (possibly weighted) graph over the examples.

Formally, suppose we are given a collection of (binary) labeled examples, L, and unlabeled examples, U. Suppose G is some graph with vertex set $V = L \cup U$. Since the labels are binary, any labeling of the graph corresponds to a partition of V, or equivalently a cut C of the graph. Now recall the edges of the graph correspond to pairwise similarity of the examples. Now edges in the above cut C have their end points labeled with opposite labels, despite the similarity indicated by the edge. So we would like to minimize these "similarity violations", for example by choosing a min-cut of the graph, while being consistent with the labeled examples in L. The consistency condition can be handled by adding two nodes '+' and '–', connecting all positive examples with an edge with weight ∞ with '+' and connecting all negative examples with an edge with weight ∞ with '–' (and we can find the usual min-cut on this modified graph). Note that our concept space includes all possible ways of labeling the unlabeled points in U. There are other reasonable ways to partition the vertices of the graph besides the min-cut (e.g. by rounding a continuous extension of min-cut). In this lecture we will focus on the min-cut.

But how do we design the graph G in the first place? A common scenario is that we have access to pairwise distances $d(x_i, x_j)$ between the examples (say, distance in some feature space). Given these distances, there are several ways to design the graph.

1.1 Unweighted "threshold" graphs

Perhaps the simplest way to design a graph using a distance metric $d(\cdot,\cdot)$ is to use unweighted edges, placing the edge between a pair of nodes $x_i, x_j \in V$ iff their distance $d(x_i, x_j) < \tau$, for some threshold $\tau \in \mathbb{R}_{\geq 0}$. Can we tune τ in the data-driven algorithm design framework?

Formally, a problem instance is given by a set of n nodes $V = (L, U)$, along with pairwise distances $d(\cdot,\cdot)$ between these nodes. A natural utility function is the average accuracy of the predicted labels. That is, $u_\tau(V, d)$ is the fraction of points labeled correctly when we construct the (unweighted) graph with threshold parameter τ, and label the unlabeled nodes using a min-cut of the modified graph described above. How many instances $(V_1, d_1), (V_2, d_2), \ldots$ are sufficient to learn a good parameter τ?

Theorem 1. *Let* $\mathcal{U} = \{u_\tau(\cdot,\cdot) \mid \tau \in \mathbb{R}_{\geq 0}\}$. *Then* $\mathrm{Pdim}(\mathcal{U}) = O(\log n)$.

Proof. On a fixed problem instance (V, d), as we vary τ, there are at most n^2 distinct values of τ at which the graph may change, corresponding to distinct values of $\{d(x_1, x_j) \mid x_i, x_j \in V\}$. The utility (the min-cut, and therefore the accuracy of the min-cut algorithm) is fixed once the graph is fixed. Thus, the dual utility function u_τ is piecewise constant with $O(n^2)$ pieces. Thus, $\mathrm{Pdim}(\mathcal{U}) = O(\log n)$, using a lemma from an earlier lecture. □

It turns out that the above pseudo-dimension bound is tight up to constants. That is, one can also show a lower bound $\Omega(\log n)$. We will see lower bounds in a later lecture.

1.2 Weighted graphs using polynomial or exponential kernels

A more refined (and empirically often better) approach is to create a weighted graph. As before, the problem instance consists of a set of nodes V and a distance metric $d(\cdot,\cdot)$ defined over pairs of nodes.

One way to create a weighted similarity graph is to use a *polynomial kernel*, $w(x_i, x_j) = \left(\frac{1}{d(x_i,x_j)} + \alpha\right)^k$ for some fixed positive integer k and hyperparameter α. A more popular approach is to use the *Gaussian kernel* (also called the Radial Basis Function or RBF kernel),

$$w(x_i, x_j) = \exp\left(-\frac{d(x_i, x_j)^2}{\sigma^2}\right),$$

with bandwidth hyperparameter σ. The analysis is similar in either case, so we will focus on the RBF kernel.

Let $\tilde{u}_\sigma(V, d)$ denote the fraction of points labeled correctly when we construct the weighted graph via the Gaussian kernel with bandwidth parameter σ, and label the unlabeled nodes using the min-cut approach.

Theorem 2. *Let* $\tilde{\mathcal{U}} = \{\tilde{u}_\sigma(\cdot,\cdot) \mid \sigma \in \mathbb{R}_{\geq 0}\}$. *Then* $\mathrm{Pdim}(\tilde{\mathcal{U}}) = O(n)$.

Proof. Fix a problem instance (V, d). We will show that the dual utility function $\tilde{u}^*_{V,x}$ is piecewise constant with $O(n^2 2^{2n})$ pieces. For a pair of cuts, C_1 and C_2, we have the following condition comparing the weight of their edges

$$\sum_{(x_i,x_j)\in\delta(C_1)} w(x_i,x_j) \le \sum_{(x_k,x_l)\in\delta(C_2)} w(x_k,x_l),$$

where $\delta(C)$ denote the set of edges across the cut C. Setting $y = \exp(-1/\sigma^2)$, gives us an exponential inequation in y of the form

$$\sum_i y^{a_i} \le 0,$$

where the sum consists of at most $\binom{n}{2}$ edges. We care about the critical points y (which corresponds to critical points σ) where the above holds with equality, as the set of points where the min-cut may change is contained within the set of these critical points. To this end, we have the following lemma on the number of real solutions of exponential equations.

Lemma 1. *The equation $\sum_{i=1}^n a_i e^{b_i x} = 0$ where $0 \ne a_i, b_i \in \mathbb{R}$ has at most $n-1$ distinct solutions $x \in \mathbb{R}$.*

Proof. We will use induction on n. It is easy to verify that there is no solution for $n = 1$. We assume the statement holds for all $1 \le n \le N$. Consider the equation $\phi_{N+1}(x) = \sum_{i=1}^{N+1} a_i e^{b_i x} = 0$. Since $a_1 \ne 0$, we can write

$$\phi_{N+1}(x) = \sum_{i=1}^{N+1} a_i e^{b_i x} = a_1 e^{b_1 x}\left(1 + \sum_{i=2}^{N+1} \frac{a_i}{a_1} e^{(b_i - b_1)x}\right) =: a_1 e^{b_1 x}\left(1 + g(x)\right).$$

By our induction hypothesis, $g'(0) = 0$ has at most $N-1$ solutions, and so $(1+g(x))'$ has at most $N-1$ roots. By Rolle's theorem, $(1+g(x))$ has at most N roots, and therefore $\phi_{N+1}(x) = 0$ has at most N solutions. □

For polynomial kernels, we can establish a similar lemma (using the above argument or by the Descartes' rules of signs). Using the above lemma, across all pairs of cuts C_1, C_2, we have at most $\binom{n}{2} \cdot \binom{2^n}{2}$ distinct critical points. The utility (the min-cut and therefore its accuracy) is fixed over any interval induced by these critical points. Thus, the dual utility function u_τ is piecewise constant with $O(n^2 2^n)$ pieces. Therefore, $\mathrm{Pdim}(\tilde{\mathcal{U}}) = O(n)$. □

As it turns out, this bound on the pseudo-dimension of the utility function for the exponential kernel is tight as well (up to constants).

2 Linear Regression

We will now turn our attention to a fundamental algorithm in machine learning, statistics and data science, namely *Least Squares Regression.* Given a feature matrix $X \in \mathbb{R}^{n\times d}$ and the corresponding real-valued labels $y \in \mathbb{R}^n$, *ordinary least squares* (OLS) is given by the following convex optimization problem,

$$\min_{w \in \mathbb{R}^d} \frac{1}{2}\|Xw - y\|^2.$$

Here each row of X corresponds to a d-dimensional feature vector for a single datapoint. w is essentially the best linear fit for the data, and should result in a small validation loss $\ell(X', y') = \frac{1}{2}\|X'w - y'\|^2$ on unseen data X', y' for the same problem/task.

To avoid overfitting the training set, and for additional useful properties like feature selection, one often adds some regularization penalty to the OLS objective, typically in terms of the L_p-norm of w, say for $p = 1$ or $p = 2$.

2.1 Ridge Regression

We will first look at adding the L_2 penalty, which is popularly known as *ridge regression.* Here the optimization problem has a regularization penalty hyperparameter λ_2.

$$\min_{w \in \mathbb{R}^d} \frac{1}{2}\|Xw - y\|^2 + \lambda_2\|w\|_2^2.$$

A problem instance is given by a tuple (X, y, X', y') such that we find the best w_{λ_2} by solving the (regularized) optimization problem on the training split (X, y), and evaluate the learned w_{λ_2} on the validation split (X', y'), that is $\ell_{\lambda_2}(X', y') = \frac{1}{2}\|X'w_{\lambda_2} - y'\|^2$.

We will show that the dual loss function is a rational (ratio of two polynomials in λ_2) function with degree at most d.

First, we give another useful tool for analyzing the pseudo-dimension of single parameter utility function classes (due to Balcan et al, STOC'21).

Lemma 2. *A function $h : \mathbb{R} \to \mathbb{R}$ is said to have at most B oscillations if for every $z \in \mathbb{R}$, the function $\rho \mapsto \mathbb{I}_{\{h(\rho) \geq z\}}$ is piecewise constant with at most B discontinuities. Let $\mathcal{U} = \{u_\rho : \Pi \to \mathbb{R} \mid \rho \in \mathbb{R}\}$, of which each dual function $u_x^*(\rho)$ for any $x \in \Pi$ has at most B oscillations. Then* $\mathrm{Pdim}(\mathcal{U}) = O(\log B)$.

For example, the constant function has zero oscillations, and a quadratic function has at most two oscillations.

Proof. On any fixed instance x_i, for any fixed threshold r_i, the function $\rho \mapsto \mathbb{I}_{\{u^*_{x_i}(\rho) \geq r_i\}}$ has at most B discontinuities by the definition of oscillations and therefore induces at most $B+1$ intervals on $\mathbb{R}$. Thus, for m instances, there are at most mB discontinuities across $i \in [m]$. For each of the $\leq mB+1$ intervals, the above-below pattern of all the functions in $\mathcal{U}$ is fixed. For pseudo-shattering $x_1, \ldots, x_m$, we must have $2^m \leq mB + 1$, which implies $m = O(\log B)$. □

We will now use this to analyze the family of ridge regression algorithms parameterized by λ_2.

Theorem 3. *Let $\mathcal{L}_2 = \{\ell_{\lambda_2}(\cdot, \cdot) \mid \lambda_2 \in \mathbb{R}_+\}$. Then* $\mathrm{Pdim}(\mathcal{L}_2) = O(\log d)$.

Proof. The unique solution to the ridge regression optimization, for any $\lambda_2 > 0$, is given by

$$w_{\lambda_2} = (X^T X + \lambda I)^{-1} X^T y.$$

This may be seen by setting the gradient w.r.t. w to zero and solving for w.

We now have the following simple lemma.

Lemma 3. *Let A be an $r \times s$ matrix. Consider the matrix $B(\lambda) = (A^T A + \lambda I_s)^{-1}$ and $\lambda > 0$. Each entry of $B(\lambda)$ is a rational polynomial $P_{ij}(\lambda)/Q(\lambda)$ for $i, j \in [s]$ with each P_{ij} of degree at most $s - 1$, and Q of degree s.*

Proof. Let $G = A^T A$ be the Gramian matrix. G is symmetric and therefore diagonalizable, and the diagonalization gives the eigendecomposition $G = E\Lambda E^{-1}$. Thus we have

$$(A^T A + \lambda I_s)^{-1} = (E\Lambda E^{-1} + \lambda E E^{-1})^{-1} = E(\Lambda + \lambda I_s)^{-1} E^{-1}$$

But Λ is the diagonal matrix $\text{diag}(\Lambda_{11}, \ldots, \Lambda_{ss})$, and therefore $(\Lambda + \lambda I_s)^{-1} = \text{diag}((\Lambda_{11}+\lambda)^{-1}, \ldots, (\Lambda_{ss}+\lambda)^{-1})$. This implies the desired characterization, with $Q(\lambda) = \Pi_{i\in[s]}(\Lambda_{ii} + \lambda)$ and

$$P_{ij}(\lambda) = Q(\lambda) \sum_{k=1}^{s} \frac{E_{ik}(E^{-1})_{kj}}{\Lambda_{kk} + \lambda} = \sum_{k=1}^{s} \left(E_{ik}(E^{-1})_{kj} \Pi_{i\in[s]\setminus k}(\Lambda_{ii} + \lambda) \right).$$

□

Each entry of w_{λ_2} is therefore a rational function of the form $P_i(\lambda)/Q(\lambda)$ with each P_i of degree at most $d - 1$, and Q of degree d.

Now the validation loss

$$\ell_{\lambda_2}(X', y') = \frac{1}{2} \|X' w_{\lambda_2} - y'\|^2$$

is a rational function of λ_2 with degree at most $2d$. This implies that the dual validation loss function for any fixed instance (X, y, X', y') has at most $2d$ oscillations. Lemma 2 now implies the result. □

2.2 LASSO Regression

We will now look at L_1 penalty, which is popularly known as *LASSO regression.* Here the optimization problem has a regularization penalty hyperparameter λ_1.

$$\min_{w\in\mathbb{R}^d} \frac{1}{2} \|Xw - y\|^2 + \lambda_1 \|w\|_1.$$

In this case, we have the following piecewise structure of the dual validation loss function ℓ_{λ_1} under a "general position" assumption that follows by applying the KKT optimality conditions to the above optimization problem.

Lemma 4. *Let $X_{\mathcal{E}}$ denote the submatrix of X where only the columns corresponding to the subset of features $\mathcal{E} \subseteq [d]$ are retained. We assume that $X_{\mathcal{E}}^T X_{\mathcal{E}}$ is invertible for each $\mathcal{E} \subseteq [d]$. Then the dual loss function on any fixed instance (X, y, X', y') is piecewise quadratic with at most 3^d pieces, where each piece corresponds to a solution w_{λ_1} of the form*

$$w_{\lambda_1,\mathcal{E}} = (X_{\mathcal{E}}^T X_{\mathcal{E}})^{-1}(X_{\mathcal{E}}^T y - \lambda_1 s), w_{\lambda_1,[d]\setminus\mathcal{E}} = 0,$$

where $\mathcal{E} \subseteq [d]$ is the set of non-zero coefficients of w_{λ_1}, and s is the sign vector $\{-1, 1\}^{|\mathcal{E}|}$.

We can use this lemma to establish the following bound on the pseudo-dimension.

Theorem 4. *Let $\mathcal{L}_1 = \{\ell_{\lambda_1}(\cdot,\cdot) \mid \lambda_1 \in \mathbb{R}_+\}$. Then* $\mathrm{Pdim}(\mathcal{L}_1) = O(d)$.

Proof. We will use the above piecewise structure and the general result for piecewise decomposable functions from the last lecture. The pseudo-dimension of the quadratic piece functions is $O(1)$ (constant number of oscillations, so we can use Lemma 2) and the VC dimension of the one-dimensional linear thresholds is $O(1)$. Moreover, we have at most $3^d - 1$ critical points (corresponding to distinct boundary functions). Thus, $\mathrm{Pdim}(\mathcal{L}_1) = O(1 + 1 \cdot \log 3^d) = O(d)$. □

Turns out this bound on the pseudo-dimension of LASSO loss is also asymptotically tight.

Additional Resources:

- Avrim Blum and Shuchi Chawla. “Learning from Labeled and Unlabeled Data using Graph Mincuts.” In Proceedings of the Eighteenth *International Conference on Machine Learning*, pp. 19-26. 2001.

- Maria-Florina Balcan and Dravyansh Sharma. “Data driven semi-supervised learning.” In *Advances in Neural Information Processing Systems* 34 (2021): 14782-14794.

- Maria-Florina Balcan, Mikhail Khodak, Dravyansh Sharma, and Ameet Talwalkar. “Provably tuning the ElasticNet across instances.” In *Advances in Neural Information Processing Systems* 35 (2022): 27769-27782.