CS/CNS/EE 253: Advanced Topics in Machine Learning
Topic: When precisely does active learning help? **Lecturer:** Daniel Golovin
Scribe: Doris Xin **Date:** Feb 10, 2010

11.1 Myopic Strategy

11.1.1 Review

Definition 11.1.1 (Version Space)

$$\mathcal{V} = \{h \in \mathcal{H} : h \text{ consistent with all observed labels}\} \tag{11.1.1}$$

Definition 11.1.2 (Volume of a version space) *The volume of the version space is the total prior probability mass of its hypotheses. Thus, given a prior* $\mathbf{Pr}[\cdot]$ *on* $\mathcal{H}$*, we have*

$$\text{vol}(\mathcal{V}) = \sum_{h \in \mathcal{V}} \mathbf{Pr}[h] \tag{11.1.2}$$

Recall from last lecture that the goal of the myopic strategy is to maximize the expected shrinkage of the version space. Formally, this can be done by picking

$$\max_i \min_{b \in \{-1,1\}} \{\text{vol}(\mathcal{V}_{(i,b)})\} \tag{11.1.3}$$

where $\mathcal{V}_{(i,b)}$ denotes the version space after labeling x_i with b, i.e., $\mathcal{V}_{(i,b)} := \{h : h \in \mathcal{V}, h(x_i) = b\}$. As shown in the last lecture, if a random hypothesis $h \in \mathcal{H}$ is drawn according to the prior $\mathbf{Pr}[\cdot]$, then the the myopic strategy makes at most $4 \log \frac{1}{\min \mathbf{Pr}[h]}$ times as many queries in expectation as any possible strategy makes in expectation.

Example in 1D: Suppose our data lie in $[0, 1]$. Consider the following threshold hypotheses

$$\mathcal{H} := \{h_\tau : \tau \in [0, 1]\}$$

where

$$h_\tau(x) := \begin{cases} 1 & \text{if } x \geq \tau \\ -1 & \text{otherwise} \end{cases}$$

Suppose we have a uniform prior on hypotheses. Refer to the accompanying figures. If we have evenly spaced unlabeled data points along the interval $[0, 1]$, then the myopic strategy is just ordinary binary search.

Another Example in 1D:

$$\mathcal{H}_c : \left\{x \mapsto \text{sgn}\left(\textstyle\sum_{i=0}^{d} a_i x^i + cx\right) : a \in \{-1, 1\}^{d+1}\right\}$$

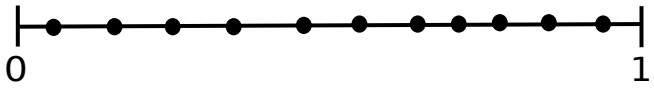

Figure 11.1.1: Example input data points.

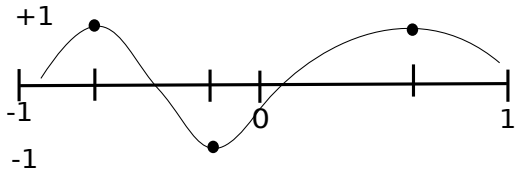

Figure 11.1.2: Example input data points and hypothesis.

where c is some constant, and the inputs are uniformly distributed. As before, suppose we have a uniform prior on hypotheses.

The VC dimension of this example is at most $d+1$ because a polynomial of degree d can shatter at most $d+1$ points.

To develop some intuition on how to pick points to label in order to maximize shrinkage of the version space, let's look at the following case:

Let $c = d/2$

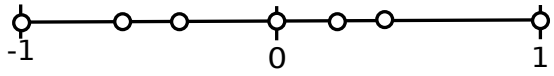

Figure 11.1.3: Example input data points.

We want to get the label of a point at 0, which reveals a_0 to us, and thus shrinks the hypothesis space exactly in half. Sampling points at -1 or 1 provides almost no meaningful information, because with very high probability $\text{sgn}\left(\sum_{i=0}^{d} a_i x^i + cx\right) = \text{sgn}(cx)$ at $x = -1$ and $x = 1$. Since we know c, we can guess the value at those points with high confidence, even without looking at any labels.

More generally, we want to query points located in the middle of the version space as opposed to off to the sides in order to maximize shrinkage. This example also illustrates the difficulty inherent in a heuristic suggested in class, namely

> Query the label of a point which allows us to infer the label of as many other unlabelled data points as possible, in expectation.

The problem with this heuristic on this data set, is that the first several queries don't allow you to infer the labels of any unlabelled data points, no matter what queries you make.

11.1.2 What can go wrong with the myopic strategy?

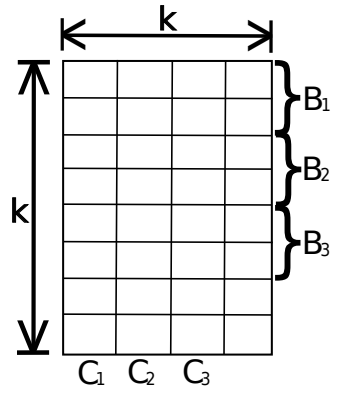

Figure 11.1.4: Grid of hypothesis with uniform distribution.

Consider the hypothesis space in Figure 11.1.4.

Three different types of queries are allowed for learning this hypothesis space:

- Block queries: "Is $h^* \in B_i$?", where block B_i consists of two rows, and each hypothesis is in exactly one block.
- Column queries: "Is $h^* \in C_i$?"
- Column-Interval queries (a, b, c): "Is $h^* \in \{(x, c) : a \leqslant x \leqslant b\}$?"

Intuitively, block queries contain more grids than column queries, which contain more grids than column-interval queries. Thus, the myopic strategy will attempt to locate the correct grid by using queries in this order.

Myopic Strategy:

1. Pick a block.
2. Find the correct block using block queries.
3. Use column-interval queries to find the correct grid.

$$\mathbf{E}[\text{Myopic cost}] = \underbrace{\frac{1}{2}\left(\frac{k}{2}\right)}_{\text{find block}} + \underbrace{\frac{1}{2}(k)}_{\text{find column}} + \underbrace{1}_{\text{find row}} = \frac{3k}{4} + 1 \tag{11.1.4}$$

whereas

$$\mathbf{E}[\text{OPT}] \leqslant \underbrace{\frac{1}{2}(k)}_{\text{find column}} + \underbrace{\lceil \log k \rceil}_{\text{find row}} \tag{11.1.5}$$

Clearly, the myopic strategy in this case is not the most cost efficient. This example can be tweaked to such that the myopic strategy requires $\Omega(\frac{\log|\mathcal{H}|}{\log\log|\mathcal{H}|})$ more queries than the optimal strategy, in expectation.

11.2 MaxMin Margin Approach

The myopic strategy cannot always be implemented because computing $\text{vol}(\mathcal{V})$ can be very difficult and expensive.

Tong & Koller[1] presented a method of approximating $\text{vol}(\mathcal{V})$ by margin of the best linear separator (max margin) with labeled points only.

Note that this approach is only applicable to homogeneous linear separators. As seen in a previous lecture, nonhomogeneous linear separators can be made homogeneous by adding a dimension to the weight vectors.

In this approach, the version space is approximated by the largest hypersphere that can fit inside the current version space. As shown in Figure 11.2.6, the hypersphere serves as a good approximation for the volume of the version space if the version space is described by a regular polytope. The hypersphere does not serve as a good approximation when the version space is highly irregular.

The SVM unit vector $\mathbf{w}_i$ is the center of the hypersphere, and the radius m_i of the hypersphere is proportional to the size of the margin of $\mathbf{w}_i$. Suppose we have a candidate unlabeled instance x in the pool. We can estimate the relative size of the resulting version space V^- by labeling x as -1, finding the SVM obtained from adding x to our labeled training data and looking at the size of its margin m^- . We can perform a similar calculation for V^+ by relabeling x as class $+1$ and finding the resulting SVM to obtain margin m^+ . Since we want an equal split of the version space in order to maximize shrinkage, we want to pick the point x for which $\min(m^-, m^+)$ is the greatest.

The class noticed that points whose SVM unit vectors form the same angle with a given unit vector but are not equidistant from the origin require a scaling factor that takes out the bias towards points that are located closer to the origin but do not actually shrink the version space more. See Figure 11.2.7. Alternately, we can project the points onto the unit sphere, so that all points have unit distance from the origin.

11.3 Splitting Index

We have seen in the last lecture that there exist problems for which active learning cannot effectively half the hypothesis space. We need to characterize problems to determine whether active learning will help. To do so, we introduce a "non-Bayesian framework." Let $\mathcal{H}$ be the hypothesis class whose probability distribution is not known, $\mathcal{D}$ be the distribution of the input data. For a fixed ε, find

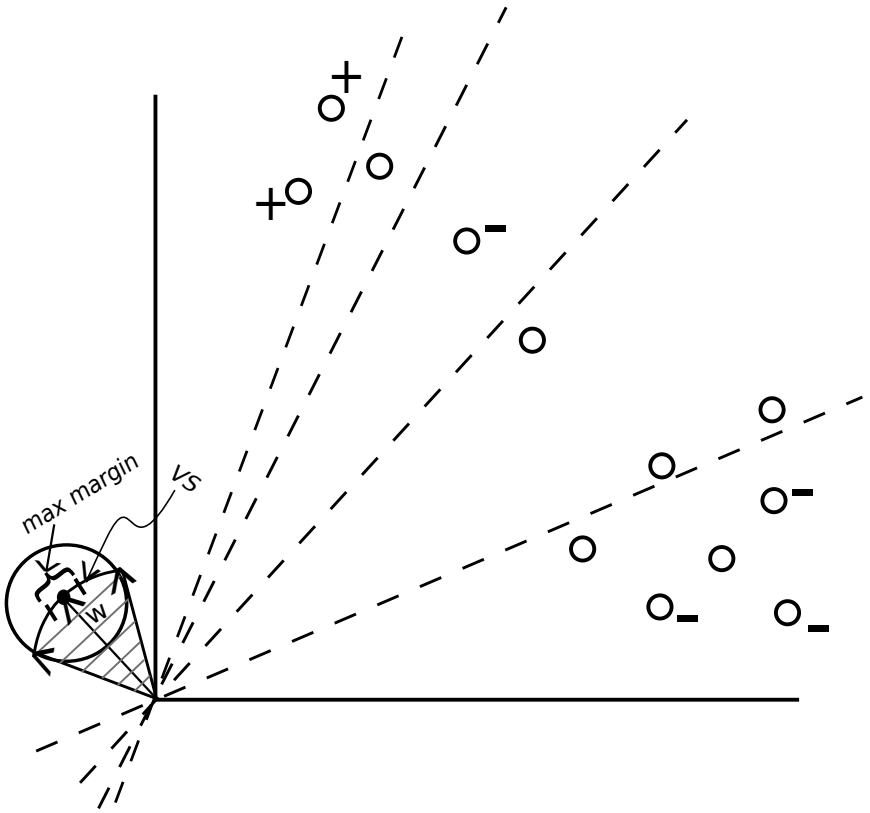

Figure 11.2.5: MaxMin margin.

$h \in \mathcal{H}$ such that $\text{err}(h) \leqslant \varepsilon$ over samples drawn from $\mathcal{D}$. In this approach, there is no notion of $\text{vol}(\mathcal{V})$.

Definition 11.3.1 (Distance between two hypotheses)

$$d(h, h') = \mathbf{Pr}_{x \sim \mathcal{D}}[h(x) \neq h'(x)] \tag{11.3.6}$$

We want to shrink the version space until $\forall\ h, h' \in \mathcal{V},\ d(h, h') \leqslant \varepsilon$. In other words, we want to cut the version space to shrink its *diameter*, the maximum distance between two hypotheses in $\mathcal{V}$. See Figure 11.3.8. Towards this end we imagine a graph with $\mathcal{V}$as vertices and $\{(h, h') \in \binom{\mathcal{V}}{2} : d(h, h') > \epsilon\}$ as edges, and try to eliminate all edges by labelling points and deleting vertices from the graph (i.e., hypotheses from $\mathcal{V}$).

11.3.1 Splitting Index

Definition 11.3.2 *For some $Q \subseteq \mathcal{H} \times \mathcal{H}$ let*

$$Q_\varepsilon \equiv \{(h, h') \in Q : d(h, h') > \varepsilon\}$$

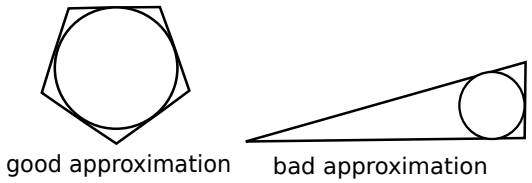

Figure 11.2.6: Approximating $\text{vol}(\mathcal{V})$ using hyperspheres.

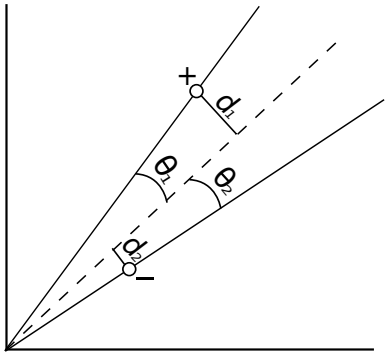

Figure 11.2.7: While $d_1 > d_2$, $\theta_1 = \theta_2$, which implies that both data points shrink the version space by the same amount. This suggests that the margins should be weighted by the data points' distance from the origin.

Definition 11.3.3 *Fix a finite set $Q \subseteq \mathcal{H} \times \mathcal{H}$ of* edges. *For some data point x_i, we say x_i* ρ-splits *Q if for all $b \in \{-1, 1\}$, labeling x_i with b eliminates a ρ fraction of the edges in Q. That is,*

$$\textit{For all } b \in \{-1, 1\},\ |Q \cap \mathcal{H}_{(i,b)} \times \mathcal{H}_{(i,b)}| \leqslant (1 - \rho)|Q|$$

Here, $\mathcal{H}_{(i,b)} := \{h \in \mathcal{H} : h(x_i) = b\}$.

Note that if we observe label b for x_i then we eliminates all edges in Q incident on hypotheses that are not consistent with the new observation of $h(x_i) = b$.

Definition 11.3.4 *A subset of hypotheses $S \subset \mathcal{H}$ is $(\rho, \varepsilon, \tau)$-splittable if for all finite edge-sets $Q \subset S \times S, \mathbf{Pr}_{x \sim \mathcal{D}}[x$ ρ-splits $Q_\varepsilon] \geqslant \tau$.*

- *ε is the quality measure on the hypotheses.*
- *ρ measures how fast we can eliminate bad pairs (h, h') relative to the speed of active learning given "good" data points.*
- *τ is the fraction of "good" points.*

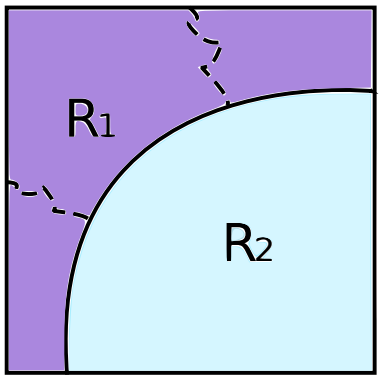

Figure 11.3.8: While the split decreases the diameter of R_2, the diameter of R_1 is the same as that of the original $\mathcal{V}$. The dotted lines show the locations of splits that help shrink the diameter of R_1.

Q should be an ε_0-cover of $\mathcal{H}$ for some $\varepsilon_0 < \frac{\varepsilon}{2}$, which means that $\forall\ h \in \mathcal{H}\ \exists\ h' \in Q$ such that $d(h, h') \leqslant \varepsilon_0$.

Theorem 11.3.5 *If Q is $(\rho, \varepsilon, \tau)$ -splittable, then there is an active learning algorithm that learns h with $err(h) \leqslant \varepsilon_o$ using $\tilde{O}\left(\frac{1}{\varepsilon} + \frac{1}{\rho\tau}\right)$ unlabeled points, and $\tilde{O}\left(\frac{1}{\rho}\right)$ labeled points.[2]*

Example. Let us return to the simple example with data in $[0, 1]$, simple threshold hypotheses $\mathcal{H} := \{h_\tau : \tau \in [0, 1]\}$, a uniform prior over $\mathcal{H}$, and unlabeled data points drawn uniformly at random from $[0, 1]$. This example is $\left(\rho = \frac{1}{2}, \varepsilon, \varepsilon\right)$-splittable for all $\varepsilon > 0$. To see this, first note the distance is $d(h_\tau, h_{\tau'}) = |\tau - \tau'|$, and also the probability that we draw an unlabeled data point which, if labeled, will cut the edge $(h_\tau, h_{\tau'})$ is $|\tau - \tau'|$. Moreover, if we consider any finite set $S \subset \mathcal{H}$, and let $Q = S \times S$, then picking a point x "near the middle" (e.g., any $x \in [a, b]$ where a is such that exactly half the edges have left endpoints less than a, and $b = a + \epsilon$). allows us to eliminate at least half of the edges.

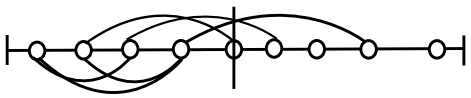

Figure 11.3.9: The line with simple threshold functions is $\left(\rho = \frac{1}{2}, \varepsilon, \varepsilon\right)$-splittable for all $\varepsilon > 0$.

References

[1] S. Tong, D. Koller, *Support Vector Machine Active Learning with Applications to Text Classification*, JMLR 2001.

[2] S. Dasgupta, and O. Patashnik, *Coarse Sample complexity bounds for active learning*, NIPS 2005.