Data Mining: Learning from Large Data Sets

FS 2016

Series 25, Nov 25th, 2016 (Active Learning)

LAS Group, Institute for Machine Learning
Dept. of Computer Science, ETH Zürich
Prof. Dr. Andreas Krause
Web: `http://las.ethz.ch/teaching/dm-f16/`
Email questions to:
Josip Djolonga, josipd@inf.ethz.ch

It is not mandatory to submit solutions and sample solutions will be published after one week. If you choose to submit your solution, please send an e-mail from your `ethz.ch` **address with subject** `Exercise5` **containing a PDF (LaTeXor scan) to** `josipd@inf.ethz.ch` **until Thursday, 1 Dec 2nd 2016.**

Problem 1 (Actively learning a union of intervals):

Suppose you are given a pool $X = \{x_1, \ldots, x_n\}$ of n unlabeled examples where each $x_i \in [0, 1]$. Further suppose there are *unknown* constants $0 \leq a < b < c < d \leq 1$ such that all $x_i \in [a, b] \cup [c, d]$ are labeled with 1, whereas all remaining points are labeled with -1. We would like to develop a pool-based active learning scheme that infers the labels of all unlabeled examples. The algorithm sequentially selects one of the n examples and obtains its true label (i.e., there is no noise).

1. Show that in general, n labels are needed to infer the labels of all unlabeled examples.
2. For $x < x'$ define $E(x, x') = \big|X \cap [x, x']\big|$, i.e, the number of examples contained in the interval $[x, x']$. Suppose $E(a, b) \geq m$, $E(b, c) \geq m$ and $E(c, d) \geq m$ for some known constant $m \geq 1$.
   (a) Define an active learning scheme that selects examples given knowledge of m. How many samples are needed as a function of m and n?
   (b) Can you come up with an algorithm that works even without knowledge of m? That is, develop an algorithm that uses (approximately) the same number of labels as the algorithm in (a) with $m = \min\{E(a, b), E(b, c), E(c, d)\}$? You're allowed to use a randomized algorithm and bound the expected number of labels requested.

Solution 1:

Assume that the points are all distinct from one another. Suppose the algorithm returns -1 for the first $n - 1$ unlabeled examples queried. Then it will not be able to determine the value of the $n-$th unlabeled example without querying it. To prove this, suppose x is the last remaining unlabeled example, and $\varepsilon > 0$ so that $\varepsilon < \min_{i,j} |x_i - x_j|$. Then the algorithm will not be able to distinguish the hypotheses $x - \varepsilon < a < b < c < d < x$ (in which case x should be labeled -1) or $x - \varepsilon < a < b < c < x < d$, in which case x should be labeled 1.

Solution 2(a):

Sort X in ascending order, and rename the points so that $i < j \Rightarrow x_i \leq x_j$ for all i and j. Label every $m-$th example, starting with x_1. Also label x_n. This requires at most $\lceil n/m \rceil + 1$ labels. Let's use y_i to refer to the label of x_i. Suppose for some i, it holds that $y_{im} = x_{(y+1)m}$. Then our assumptions imply that $y_j = y_{im}$ for $im \leq j \leq (i + 1)m$. Furthermore, there are at most 4 cases where $y_{im} \neq y_{(i+1)m}$. For these cases, we can use binary search to determine j, $im \leq j \leq (i + 1)m$ such that $y_{im} = y_j \neq y_{j+1} = y_{(i+1)m}$. Thus the total number of samples is bounded by $\lceil n/m \rceil + 1 + 4\lceil \log_2 m \rceil$.

Solution 2(b):

The basic idea is to develop a sampling scheme which produces "witnesses" for the intervals $[a, b], [b, c]$ and $[c, d]$: If we've found 3 examples, $x_i < x_j < x_k$ with $y_i = 1, y_j = -1$ and $y_k = 1$, then we can use binary search as above to locate the "boundaries" of these intervals. The simplest approach is to just pick examples at random until we've found our witnesses. How many samples do we need? Let T_1 be the number of rounds it takes until we select an example from $[a, b]$, T_2 the number of rounds for $[b, c]$ and T_3 for $[c, d]$. Then the total number of samples needed to identify the witnesses is bounded by $T = T_1 + T_2 + T_3$. Thus the expected number of samples is bounded by $E[T] = E[T_1] + E[T_2] + E[T_3]$ by linearity of expectation. Let us now bound $E[T_1]$ (the other quantities are bounded analogously). Consider picking an example uniformly at random. By our assumptions, the probability that it is contained in $[a, b]$ is at least m/n. Therefore, T_1 is geometrically distributed, with expectation $E[T_1] = n/m$. Thus $E[T] = 3n/m$. After obtaining the witnesses, we perform binary search to locate the interval boundaries, requiring at most $4\lceil \log_2(m) \rceil$ samples. Thus, the expected number of samples needed by the algorithm is $3n/m + 4\lceil \log_2(m) \rceil$.