Topics in the Frontier of Computer Science for Honor Students

Guide

Prof. Aryeh Kontorovich

Authors

Noam Atia: 311394241

Avi Ferdman: 316420132

opening

In our project we implemented one of the algorithms from the article by Lee-Ad Gottlieb, Aryeh Kontorovich and Pinhas Nisnevitch, Near-optimal sample compression for nearest neighbors.

In this document:

i. We'll give some **context and summarize the relevant parts from the article** related to the algorithm.
ii. **Describe the algorithm** itself.
iii. **Share and analyze the results** of its execution on real data.

1. *Introduction*

Nearest neighbor classification is a machine learning method that aims at labeling previously unseen query objects while distinguishing two or more destination classes. As any classifier, in general, it requires some training data with given labels and, thus, is an instance of supervised learning.

Formally, given sample S of labeled points (X, Y) = $(Xi\ , Yi)_{i\in[n]}$, where Xi is a point in metric space X and $Yi \in \{-1, 1\}$ is its label, and a distance function $\rho: X \times X \to R$.

A new unlabeled point $x \in X$ is classified as its nearest neighbor in S, which is $argmin_{Yi\in\{-1,1\}}\, \rho(x, Xi)$.

A short brief on NN classification method:

	Pros	Cons
1	Expected error is asymptotically bounded by twice the Bayesian error, when the sample size tends to infinity.	Requires storing the entire sample.
2	Simple evaluation on new data.	Requires $\Theta(\|S\|)$ time in high-dimensional metric spaces.
3	Immediate extension to multiclass labels.	Has infinite VC-dimension, implying that it tends to overfit the data.
4	Minimal structural assumptions (it does not assume a Hilbertian or even a Banach space).	

❖ Note: disadvantage no. 3 can be mitigated by taking the majority vote among $k > 1$ nearest neighbors, or by deleting some sample points.

As a result of NN classification disadvantages, Hart to pose **the problem of sample compression.** A data compression in NN classification can deal with the disadvantages we mentioned above. This problem we would like to solve, is called **Nearest Neighbor Condensing problem** (NNC)**.** The goal is to find a minimal $S_* \subset S$ that is **consistent** with S. Consistent to S means: for each $x \in S$, it's nearest neighbor in S_* implies the same label as x.

Formally, we define the NNC as follows:

Given a set $S = S_+ \cup S_-$ of points, and distance metric ρ: S × S → R. We must compute a minimal cardinality subset $S' \subset S$ with the property that for any $p \in S$, the nearest neighbor of p in S' comes from the same subset $\{S_+, S_-\}$ as does p. If p has multiple exact nearest neighbors in S', then they must all be of the same subset.

Unfortunately, $\boldsymbol{NNC \in NP-hard}$**.** Using some **heuristics** papers reached to a runtime of $O(n^2)$. Regarding the approximation, none of the papers provide approximation guarantees. Surprisingly, there are still no approximation algorithms.

The contribution of the authors of the article aims at **closing the existing gap in solutions to the NNC problem.** They present a simple near-optimal approximation algorithm for this problem, where their only structural assumption is that the points lie in some metric space.

More detailed:

i. **Doubling dimension.** For a metric (X, ρ), let λ be the smallest value such that every ball in X of radius r (for any r) can be covered by λ balls of radius $\frac{r}{2}$.

   The doubling dimension of X is: $ddim(X) = \log_2 \lambda$.

ii. $D = \{\, \rho(x,y) \mid x, y \in S \ \wedge\ x, y \text{ are opposite labeled points.} \}$

iii. $diam(S) = SUP_{x,y \in S}\, \rho(x,y)$.

iv. Scaled margin $\gamma < 1$ of a sample S, $\gamma = \frac{\min(D)}{diam(S)}$.

Their algorithm produces a consistent set $S_\gamma \subset S$ of size $\left\lceil \frac{1}{\gamma} \right\rceil^{ddim(S)+1}$. This finding can significantly speed up evaluation on test points and provide simpler generalization bounds than were previously known.

2. *Near-optimal approximation algorithm*

In this chapter we'll describe the algorithm we implemented.

Theorem 1. Given a point set S and its scaled margin $\gamma < 1$, there exists an algorithm that in time $min\{n^2, 2^{O(ddim(S))}\ n\ log\left\lceil\frac{1}{\gamma}\right\rceil\}$ computes a consistent set $S' \subset S$ of size at most $\left\lceil\frac{1}{\gamma}\right\rceil^{ddim(S)+1}$ (as mentioned before).

ε-net of point set S is a subset $S_\varepsilon \subset S$ has two properties:

i. Packing. The minimum interpoint distance in S_ε is at least ε.

ii. Covering. Every point $p \in S$ has a nearest neighbor in S_ε strictly within distance ε.

These properties would help us to find the approximation algorithm for the NNC problem using the following observation:

i. Since the margin of the point set is γ, a γ-net of S is consistent with S. That is, every point $p \in S$ has a neighbor in S_γ strictly within distance γ, and since the margin of S is γ, this neighbor must be of the same label set as p.

ii. By the packing property of doubling spaces, the size of S_γ is at most $\left\lceil\frac{1}{\gamma}\right\rceil^{ddim(S)+1}$.

The solution returned by the algorithm is S_γ, and satisfies the guarantees claimed in Theorem 1. It remains only to compute the net S_γ. A brute-force greedy algorithm can accomplish this in time $O(n^2)$: For every point $p \in S$, we add p to S_γ if the distance from p to all points currently in S_γ is γ or greater, $\rho(p, S_\gamma) \geq \gamma$.

Algorithm 1 Brute-force net construction

Require: S

1: $S_\gamma \leftarrow$ arbitrary point of S

2: for all $p \in S$ do

3: if $\rho(p, S_\gamma) \geq \gamma$ then

4: $S_\gamma = S_\gamma \cup \{p\}$

5: end if

6: end for

3. *Implementation*

i. Create in advanced potential metrics such as Euclidean, Manhattan, Chebyshev, Minkowski, to analyze and compare their compression and prediction.

ii. Get as input a labeled data set, which is set of files. Each file has set of labeled samples. The labels are equals to the file name (which means two samples from same file have same label and two samples from different files have different labels).

iii. Represent the samples we have as a binary vector, each entry in index i is 1 if i exists in the sample, else the entry is 0.

iv. For each potential metric:

1. Find Gamma γ (Minimum distance between two different labeled samples).
2. Divide to test set and train set.
3. Create $\boldsymbol{S_\gamma}$ using Algorithm 1 – Brute-force net construction.
4. Compare the quality of the prediction for both NN and NNC and calculate the compression ratio of NNC.

4. Results & Observations

$\gamma = 0.15$			
	Compression Ratio	**NNC Accuracy**	**NN Accuracy**
Euclidean	656.25	56.45%	99.71%
Manhattan	55.85	99.77%	99.77%
Chebyshev	5250.00	14.40%	99.25%
Minkowski	656.25	58.00%	99.88%
$\gamma = 0.2$			
	Compression Ratio	**NNC Accuracy**	**NN Accuracy**
Euclidean	1312.50	41.54%	99.77%
Manhattan	66.45	99.60%	99.60%
Chebyshev	5250.00	14.85%	99.31%
Minkowski	1312.50	29.65%	99.42%
$\gamma = 0.25$			
	Compression Ratio	**NNC Accuracy**	**NN Accuracy**
Euclidean	2625.00	29.08%	99.82%
Manhattan	80.76	99.71%	99.71%
Chebyshev	5250.00	14.74%	99.25%
Minkowski	1750.00	28.51%	99.94%

i. The decision of the metric distance function is crucial. Different metrics produce different compression ratio and prediction accuracies, and the variance is high.

ii. Manhattan metric can compress the data in $\sim 1 - 2$ orders of magnitude with high quality of prediction accuracy which equals to the prediction of classic NN.

iii. Although the compression in other metrics except Manhattan is much higher, this is useless because the accuracy of their prediction is less than 50% in most cases and much lower than classic NN.

iv. Chebyshev accuracy (which is clearly the worst distance metric function in this case) is almost as a pure guess between the 7 possible labels.

References

[1] Lee-Ad Gottlieb, Aryeh Kontorovich and Pinhas Nisnevitch. Near-optimal sample compression for nearest neighbors.