Weighted k-Means Coreset for Active Learning

Pushkar Nimkar

January 2022

1 Introduction

In modern supervised learning systems, one often struggles with the problem of finding good quality labeled data. It is impractical to label every instance in the data set, thus, it becomes necessary to select a subset of data to label. A common practice is to randomly sample the data. While random sampling is a good starting point, it is often affected by the input distribution. A grid-based sampling approach can be effective, but it is difficult to scale with input dimension. The problem becomes even more interesting with class imbalance. Another issue often faced in classification problems is that, all target classes are not known prior to sampling (open-set annotation). For instance, consider a face recognition system that can access numerous images but not every person in the input set is known to the system.

Active sampling is a technique in which the learning algorithm "actively" involves in sampling the data for annotation to maximize the performance on a supervised learning task. It assumes that the learner can ask queries to an "oracle" to reveal the label of points at some cost. Practically, oracle can be a human expert, an expensive biological experiment, or an user accepting or rejecting the recommended content. [Settles, 2012] presents an overview of different scenarios of active sampling. Broadly, the methods can be divided into stream-based methods and pool-based methods. In stream-based methods, the active learner needs to decide on every streaming input whether to query the label or not. The learner has to make this choice independently for each streaming input. In pool-based methods, the learner can repeatedly access a pool of unlabeled data and draws samples from the pool to maximize the performance on supervised learning task. [Settles, 2012] also lists four objective functions from earlier literature that the learner can optimize to maximize the performance on supervised learning task at a given budget of queries. The objectives include: minimizing the uncertainty, minimizing the size of hypothesis space, minimizing the expected error/variance, and maximizing the "informativeness" weighted by information density.

Quoting from [Huggins et al., 2016], an important insight is that data is most often redundant. For instance, a face recognition system installed for surveillance can capture multiple images of staff members, but images of guests are relatively rare. Hence, methods that focus on selecting representative points from a set of densely populated data points are important. At the same time, practical data sets often have complex class boundaries. A sampling method that entirely focuses on representation may not collect sufficient samples near the class boundaries to accommodate the complexity. Hence, there is a trade-off between representation and uncertainty. In this project, we propose an active sampling method that uses the coreset of weighted k-Means clustering for active learning. The weights are derived from classifier uncertainty.

Section 2 presents a brief literature overview of other relevant algorithms. The We shall restrict ourselves to a pool-based active learning scenario for classification in this project. Section 3 presents the proposed algorithm and describes the results. We present the limitation of this approach in section 4 and make some concluding remarks.

2 Literature Survey

[Settles, 2012] provides a valuable compilation of taxonomy and broad overview of classical active learning literature. [Hino, 2020] explores more recent trends in the active learning techniques. [Har-Peled et al., 2020] utilizes active learning to learn a low-dimensional convex-body from a point set by minimizing the membership queries.

Several attempts are made in the machine learning community to construct a coreset to upper bound the training loss for supervised learning algorithms such as [Huggins et al., 2016]. Similarly, [Chen, 2009] suggest a coreset algorithm to upper bound the k-Means cost function for weighted k-Means clustering. [Bachem et al., 2017] extends the algorithm from [Chen, 2009]. The weighted k-Means implementation used in this project is the one developed by [Bachem et al., 2017]. [Agarwal et al., 2005] provide a widely known survey on coreset methods. On a different note, [Shim et al., 2021] utilize the coreset selection approach to reduce the sample size of a larger data set to optimize Neural Architecture Selection (NAS).

[Sener and Savarese, 2018] provide an interesting geometric approach to active learning. They formulate active learning as a set cover problem. Assume a set of n unlabeled data points in $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$. A subset $\mathbf{s}$ of data points chosen such that the set of spheres in $\mathcal{X}$ with radius δ centered at points in $\mathbf{s}$ cover all n unlabeled points. At each iteration i, a batch of size b is selected from $\{\mathbf{x}_j\}_{j\in[n]\setminus\mathbf{s}}$ so as to minimize δ corresponding to $\mathbf{s} \cup \mathbf{s}_{i+1}$. This is the standard facility location problem. As the orignial problem is NP-hard, they also provide a greedy approximation bounded by $2 - OPT$.

Clustering methods are widely used for active sampling. Chapter 5 of [Settles, 2012] discusses this in depth. [Dasgupta and Hsu, 2008] use hierarchical clustering for active sampling. In their algorithm, learner queries from a tree node till a certain upper bound is met about "purity" (node is considered pure if it contains points from a single target class) of the node. If a node is impure, it is further split and samples are drawn from its children.

3 Proposed Algorithm

The intuition behind the proposed method is that if we cluster the data weighted by uncertainty scores, we expect to see more densely packed clusters in the regions of maximum uncertainty than otherwise. However, in an attempt to minimize the clustering objective function, the clustering method is expected to pack more clusters in densely populated regions while covering the space.

Algorithm 1 states the proposed algorithm. We assume that the classification model θ gives a probability distribution over the set of know classes $\theta : x \to \Delta_y$. The uncertainty for each x is calculated as statistical entropy of Δ_y. We use the algorithm from [Bachem et al., 2017] to construct coreset for k-Means clustering as described above.

3.1 Dataset for Testing

Although active sampling is a problem of practical use and often used with fairly high dimensional inputs such as image or video embeddings, we are using a simple synthetic two-dimensional data set here to illustrate the features of this algorithm. Figure 1 shows the data set used for the experiments.

The data set contains two majority classes: orange and blue of 1000 samples each, and four minority classes: green, red, brown, and violet of 30 samples each. Although the classes significantly differ in positions, the boundaries are not well-separated. Thus, the classifier would always retain some uncertainty.

Figure 3 illustrates the set of samples selected for one batch on synthetic data. We note that the sampled points (coreset for clustering) are influenced by samples with high uncertainty as well high density regions.

Algorithm 1 Weighted k-Means coreset sampling

```
$\mathcal{L}$ = Pool of labeled instances $\{\langle x, y\rangle^{(l)}\}_{l=1}^{L} = \emptyset$
$\mathcal{U}$ = Pool of unlabeled instances $\{x^{(u)}\}_{u=1}^{U}$
$\theta$ = Current classification model
$(k, \epsilon)$ = Clustering coreset parameters.        ▷ $k$ is the number of $k$-Means centers
for each batch do
    if $\theta$ exists then
        $w \leftarrow$ uncertainty$(\theta(\mathcal{U}))$        ▷ Calculate the uncertainty
    else
        $w \leftarrow \langle 1^{(u)}, ...\rangle_{u=1}^{U}$
    end if
    Calculate $(k, \epsilon)$ $k$-Means coreset $(R, \rho)$ of $(\mathcal{U}, w)$
    $Q = \{(x^*, y^*) | \forall x^* \in R, y^* = y(x^*)\}$        ▷ Query labels from the oracle
    $\mathcal{U} \leftarrow \mathcal{U} \setminus Q$, $\mathcal{L} \leftarrow \mathcal{L} \cup Q$
    Train classifier $\theta$ using $\mathcal{L}$
end for
```

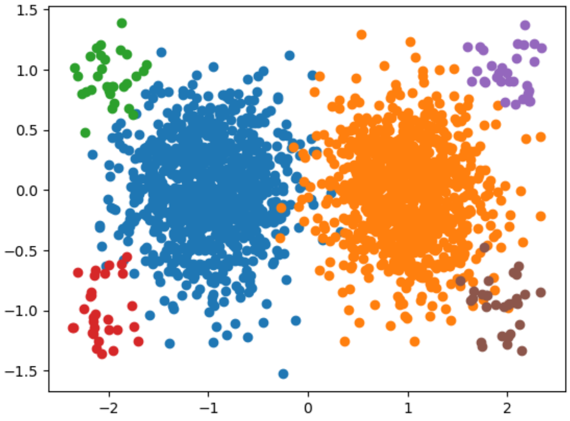
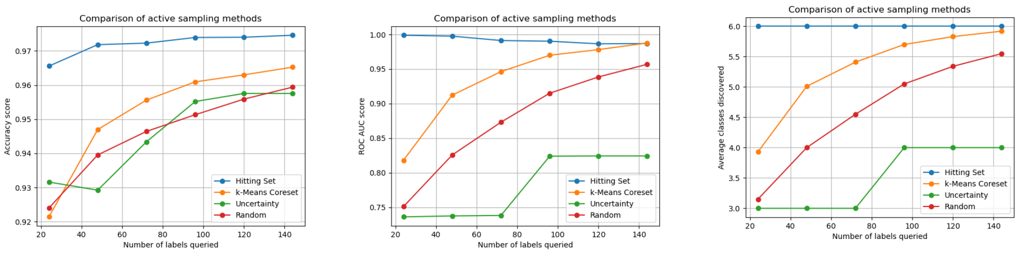


Figure 1: “Double mickey” synthetic data set



(a) Accuracy scores

(b) ROC-AUC scores

(c) Classes discovered

Figure 2: Comparing the performance of described active sampling algorithms

3.2 Results

In this section, we shall compare the above mentioned algorithm with three other popular methods on the synthetic dataset described in 3.1. The first method is from [Sener and Savarese, 2018] described above. The second method is standard uncertainty sampling. The third method is random sampling. Figure 2 shows various classification performance metrics for the active learning algorithms. We use the standard method for calculating accuracy score. The ROC-AUC scores are calculated by averaging the scores for one-vs-rest classification for all classes. The third metric, “Discovered classes” describes the total number of classes the learner has discovered for a given sample size. This gives an insight about open-set behavior of the algorithm. This is a proxy for representation metric discussed above. All performance metrics are averaged over 100 iterations. We used a multi-class logistic regression classifier for evaluating all the algorithms. Figure 4 shows samples drawn by each sampling method across all the batches.

We can observe that the method proposed by [Sener and Savarese, 2018] outperforms other methods in this data set in all metrics. The proposed algorithm performs better than random sampling and uncertainty sampling on all three metrics. We note that random sampling takes longer time to discover all classes due to class imbalance. Uncertainty sampling on the other hand can not discover all classes as it is draws all samples from region it has identified as most uncertain based on initial samples. We can clearly see this problem in figure 4. In figure 4, we can also see that the [Sener and Savarese, 2018] method is maximizing the coverage and samples are distributed almost uniformly. This can be problematic if the decision boundary is intricate.

3.3 Limitations & Future Scope

It would be interesting to run similar experiments on data set with more intricate boundaries to observe if the proposed method can utilize the uncertainty weighing more effectively to discover the decision boundary. This also requires repeating the experiments with a hypothesis class of higher complexity.

The algorithm described here is intuitive and performs better than baseline algorithms. However, it lacks rigorous theoretical foundation. I shall work towards developing a better theoretical understanding of this algorithm or identify if there are any serious limitations to this approach.

4 Conclusion

We proposed an active sampling algorithm using weighted k-Means clustering coreset. We measured the performance of this method on a dummy data set and compared it against three well-known algorithms. We noted that the above described method tries to balance between uncertainty and representation metrics from [Settles, 2012].

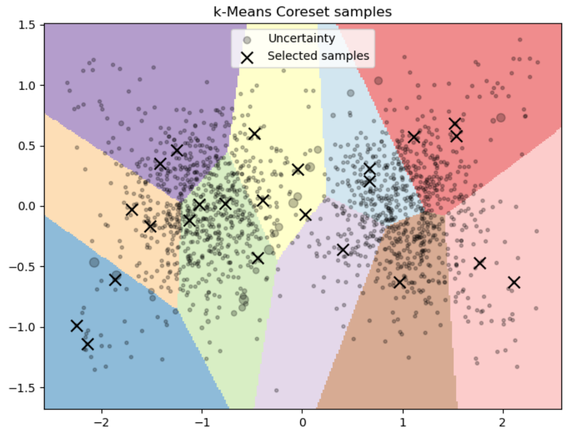
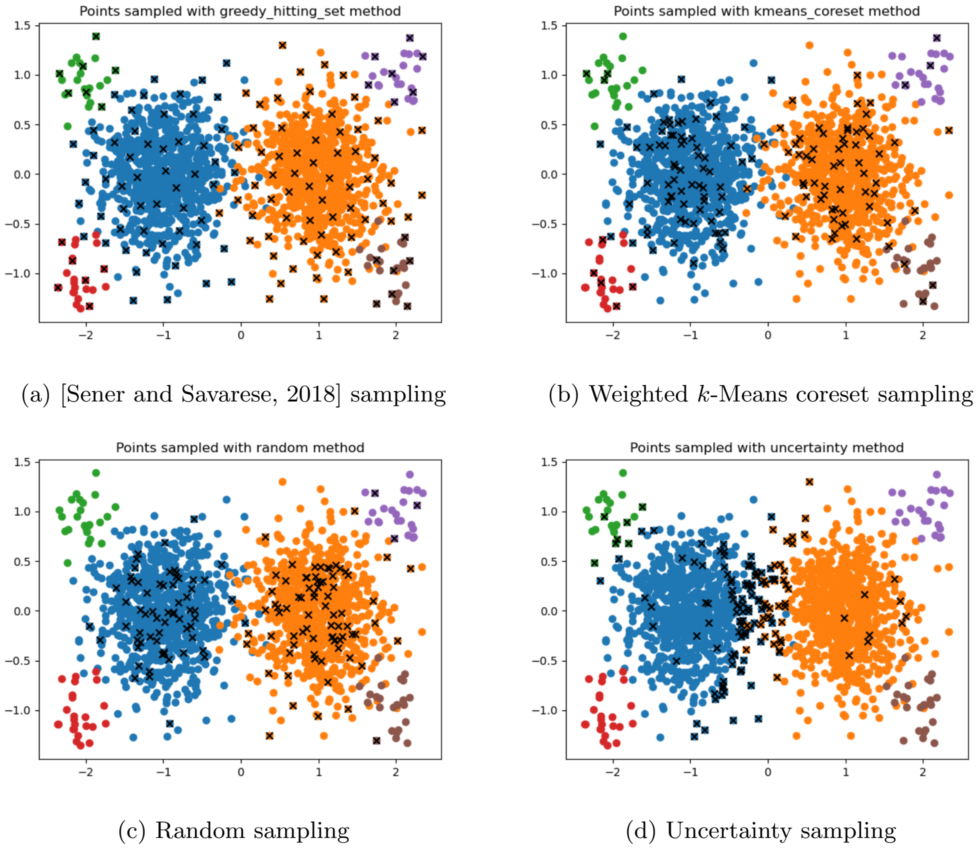


Figure 3: Data selected in a single batch by proposed algorithm



(a) [Sener and Savarese, 2018] sampling

(b) Weighted *k*-Means coreset sampling

(c) Random sampling

(d) Uncertainty sampling

Figure 4: Full set of samples drawn by each method

References

[Agarwal et al., 2005] Agarwal, P. K., Har-Peled, S., and Varadarajan, K. R. (2005). Geometric approximation via coresets. In *COMBINATORIAL AND COMPUTATIONAL GEOMETRY, MSRI*, pages 1–30. University Press.

[Bachem et al., 2017] Bachem, O., Lucic, M., and Krause, A. (2017). Practical coreset constructions for machine learning.

[Chen, 2009] Chen, K. (2009). On coresets for k-median and k-means clustering in metric and euclidean spaces and their applications. *SIAM Journal on Computing*, 39(3):923–947.

[Dasgupta and Hsu, 2008] Dasgupta, S. and Hsu, D. (2008). Hierarchical sampling for active learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 208–215, New York, NY, USA. Association for Computing Machinery.

[Har-Peled et al., 2020] Har-Peled, S., Jones, M., and Rahul, S. (2020). Active learning a convex body in low dimensions. In *ICALP*.

[Hino, 2020] Hino, H. (2020). Active learning: Problem settings and recent developments.

[Huggins et al., 2016] Huggins, J., Campbell, T., and Broderick, T. (2016). Coresets for scalable bayesian logistic regression. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

[Sener and Savarese, 2018] Sener, O. and Savarese, S. (2018). Active learning for convolutional neural networks: A core-set approach.

[Settles, 2012] Settles, B. (2012). Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114.

[Shim et al., 2021] Shim, J., Kong, K., and Kang, S. (2021). Core-set sampling for efficient neural architecture search. *CoRR*, abs/2107.06869.