Semiparametric Subsampling and Data Condensation for Large-scale Data Analytics

Omar Alhussein*, Paul D. Yoo‡, Sami Muhaidat†, and Jie Liang§
* Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Canada
† Department of Electrical and Computer Engineering, Khalifa University, UAE
‡ CSIS Dept., Birkbeck College, University of London, UK
§ Simon Fraser University, Canada

Email: oalhusse@uwaterloo.ca, paul.d.yoo@ieee.org, sami.muhaidat@kustar.ac.ae, jiel@sfu.ca

***Abstract*—Subsampling is often used to reduce the complexity of large datasets. However, such methods need to ensure that the subsampled data are representative of the original dataset. Here, we introduce a new clustering-based data condensation (subsampling) framework for large datasets. The framework relies on the use of stratified sampling, Voronoi diagrams, and variational Bayes-based Gaussian mixture clustering. We tested the proposed framework on three large imbalanced benchmark datasets, namely cod-RNA, ds1.10, and ds1.100. The efficiency and generality of the proposed framework were assessed by comparing the predictive performance of the reduced datasets with the original datasets over two machine-learning classifiers, namely the random forest, and the radial basis function network. The evaluation metrics included the accuracy, F-measure and reduction percentage. We found that very high reduction percentages can be achieved using our new framework while maintaining satisfactory predictive performance.**

***Index Terms*—Data Condensation, machine learning, subsampling, clustering.**

I. Introduction

In machine learning (ML), datasets can broadly be classified into three types based on the existence of data labels: supervised, semi-supervised, and unsupervised. In supervised learning, fully labeled datasets are used, whereas in semi-supervised learning the datasets may be partially labeled, for example to reduce costs when full data labeling is expensive. In either category, when the dataset is very large, there is a need to reduce the dataset or select a smaller representative dataset, whereby the predictive performance of the condensed dataset $\boldsymbol{C}$ is close to that of the original dataset $\boldsymbol{D}$. In supervised learning, such need is motivated by the fact that most conventional ML algorithms do not scale to large datasets in terms of computational time and space complexity [1]. In addition, large datasets often reduce the classifier's generalization accuracy due to overfitting caused by noisy data, and irrelevant or redundant features. Indeed, as the size of the dataset increases, the learning curve in a classifier's performance reaches a plateau [2]. In semi-supervised learning, requesting labels for the whole dataset is often cumbersome and unnecessary, and it is therefore more appropriate to produce a smaller representative dataset to request for labeling, which calls for the process of data condensation.

Data condensation (reduction) can be achieved by either instance reduction (instance selection) or dimensionality reduction techniques. Instance selection methods may be classified as either wrapper methods or filter methods. In the former, the instance selection criterion is based on the error of a predetermined classifier model, commonly a k-nearest neighbor (k-NN) classifier [3]. For example, the condensed nearest neighbor (CNN) [4] is an incremental method in which the initial step involves the random selection of one instance from each class, which is assigned to $\boldsymbol{C}$. Then each new instance $\boldsymbol{p}$ ($\in$ $\boldsymbol{D}$) is classified incrementally using $\boldsymbol{C}$. If $\boldsymbol{p}$ is misclassified, then it is included in $\boldsymbol{C}$ and so forth. Many extensions and variants of the CNN algorithm have been reported, e.g. the reduced nearest neighbor (RNN) [5], selective nearest neighbor (SNN) [6], and generalized condensed nearest neighbor (GCNN) methods [7]. Another useful classifier when combined with instance selection is the support vector machine (SVM) [8], [9]. For example, the SV-kNNC algorithm classifies $\boldsymbol{D}$ using the SVM, and then the resulting support vectors are clustered using the k-means algorithm [9]. Consequently, instances belonging to homogeneous clusters are retained. If the cluster is not homogeneous, then only instances of the majority class are preserved.

The other category of instance selection methods (filter methods) use a selection criterion that is not based on any classification algorithm. Various filter methods have been reported, some of which rely on selecting the so-called border instances. An instance $\boldsymbol{p}$ ($\in$ $\boldsymbol{D}$) is a border instance of some class A if $\boldsymbol{p} \in A$ and if $\boldsymbol{p}$ is the nearest neighbor of an instance belonging to another class that is not in A [10]. Examples of such algorithms include POP and POC-NN [11], [12]. Some other filter methods rely on clustering, in which $\boldsymbol{D}$ is split into a number of clusters and the centroids (medoids) of the clusters are selected for the reduced set $\boldsymbol{C}$ [13]–[18]. Examples include the generalized-modified Chang (GCM) algorithm, which merges clusters containing similar classes and then fetches their centroids [17], and the nearest sub-class classifier (NSB) method, which selects different numbers of centroids per class using the maximum variance cluster algorithm [18]. Wrapper and filter methods have been comprehensively reviewed [2], [10], [19].

In this paper, we introduce a new filter-based data condensation framework for supervised, semi-supervised and unsupervised datasets through the use of stratified random sampling and variational inference machinery. In contrast to the aforementioned filter-based techniques, the proposed framework can be generalized not only to fully labeled datasets but also to semi-labeled and unlabeled datasets. Briefly, the framework first selects some representative instances from $\boldsymbol{D}$, which are used to construct a Voronoi diagram. For every nth Voronoi region, we run the variational Bayes-based Gaussian mixture (VBGM) clustering algorithm to select K_n centroids. The Voronoi diagram allows us to apply the clustering-based data condensation algorithm to each Voronoi region in parallel. We test our framework on three large imbalanced benchmark datasets, namely cod-RNA [20], ds1.10, and ds1.100 [21]. We use two ML classifiers, namely on the random forest (RF), and the radial basis function network (RBFN).

The rest of this paper is organized as follows. Section II introduces the three different stages of the data condensation framework. Section III provides a short summary of the proposed framework. Section IV describes our comparative predictive performance tests on the original and condensed benchmark datasets using different ML classifiers. Section V presents our conclusions.

II. Data Condensation Algorithm

In supervised learning, one is given a training set $\boldsymbol{B} \in \mathbb{R}^{N\times M}$, where N and M correspond to the number of instances (tuples) and the number of features (attributes), respectively. Each instance $\{\boldsymbol{b}_n\}_{n=1}^{N}$ $(\in \mathbb{R}^{1\times M})$ is mapped into some target value $\{t_n\}_{n=1}^{N}$ $(\in \mathbb{R})$. The dataset $\boldsymbol{D}$ is a concatenation of $\boldsymbol{B}$ and $\boldsymbol{t}$.

The proposed data condensation framework which extends the works of Zaknich and Yu et al. [22], [23] consists of the following main stages.

- **Sampling**: Using stratified sampling, the original dataset $\boldsymbol{D}$ is sampled to produce a representative set $\boldsymbol{R} \in \mathbb{R}^{L\times M}$, where $L < N$.
- **Voronoi diagram**: A Voronoi diagram $\mathcal{V}(\boldsymbol{R})$ is constructed based on the representative set $\boldsymbol{R}$.
- **Clustering**: The original dataset $\boldsymbol{D}$ is superimposed over the constructed Voronoi diagram $\mathcal{V}(\boldsymbol{R})$, where for each nth Voronoi region $VR(\boldsymbol{r}_n, \boldsymbol{R}) \in \mathcal{V}(\boldsymbol{R})$ centered by an instance $\boldsymbol{r}_n \in \boldsymbol{R}$, K_n centroids are fetched out using the variational inference machinery, specifically via the use of the bariational Bayes-based Gaussian mixture (VBGM) algorithm. The centroids of all Voronoi regions $\{VR(\boldsymbol{r}_n, \boldsymbol{R})\}_{n=1}^{L}$ form the new reduced dataset, $\boldsymbol{C}$.

The following sections discuss each stage in greater detail.

A. Sampling

In the context of ML, sampling is often used as a standalone tool to reduce the size of large datasets. However, it is cumbersome to characterize the loss-or rarely the gain- in the subsequent performance predictive measures because the problem is data dependent [24]. Thus, the choice of the sampling technique and its corresponding sample size is a crucial and sensitive step.

Sampling strategies can be classified as dynamic or static [25]. Dynamic sampling requires some knowledge of the classification algorithm. As such, the classifier is used directly to determine whether the sample is representative of the whole or not. This is reminiscent of the wrapper condensation approach explained above. In contrast, the classification algorithm in static sampling is assumed to be unknown, and certain fixed statistics-based criteria must be used to determine whether or not the sample is representative of the original dataset.

Here, we prefer static sampling because it allows us to make no assumptions about the subsequent machine-learning algorithm, thus offering more versatility. For the sampling technique, we adopt the stratified sampling approach. In simple random sampling (SRS), a sample $\boldsymbol{R}$ is selected uniformly from $\boldsymbol{D}$, with replacement following a binomial distribution. Equivalent weights are given to all samples in the dataset so that any sample is chosen with equal probability regardless of whether or not it was previously sampled [26]. SRS minimizes bias and simplifies further analysis, but if the dataset is imbalanced then stratified sampling is better. In this context, $\boldsymbol{D}$ is divided into set of strata according to the different target values in $\boldsymbol{t}$, and SRS is performed on each stratum independently. Typically, stratified sampling can be implemented using four known approaches, namely proportional sampling, equal size sampling, Neyman's allocation, and optimal allocation. Here we adopt proportional sampling, in which the proportion sampled from each stratum is equal in the sample as it is in the original dataset. For example, if $\boldsymbol{D}$ is highly imbalanced, then a higher sampling percentage would be allocated to the small target stratum.

Upon sampling from $\boldsymbol{D}$ to produce $\boldsymbol{R}$, we need to assess whether or not the sample is sufficiently representative. There are two types of statistical tests for this purpose: parametric and non-parametric1. Parametric tests make assumptions about the statistical distribution of the original dataset. For example, the well-known t-tests and z-tests assume assume that the underlying data are normally distributed. In contrast, non-parametric tests assume no underlying statistical structure. In order to maintain generality in our framework, we use non-parametric analysis and rely on the relative entropy (Kullback-Leibler divergence) as a measure of goodness of fit between any mth feature $\boldsymbol{r}_m$ $(\in \mathbb{R}^{L\times 1})$ and its corresponding mth feature in the original dataset $\boldsymbol{d}_m$ $(\in \mathbb{R}^{N\times 1})$. The relative entropy between some two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ is given by

$$D_{KL}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{E} \Pr\{\boldsymbol{x}[i]\} \ln \frac{\Pr\{\boldsymbol{x}[i]\}}{\Pr\{\boldsymbol{y}[i]\}},$$

where E is the length of the vectors $\boldsymbol{x}$, $\boldsymbol{y}$, and $\Pr\{\boldsymbol{x}[i]\}$ is the probability of the $\boldsymbol{x}[i]$. In our case, $\boldsymbol{r}_m$ and $\boldsymbol{d}_m$ are not of the same length. Therefore, we fairly discretize both vectors to E levels.

In this paper, we perform this test using the most significant features. The feature subset selection is a well-studied problem in the open literature, whereby a feature selection algorithm consists of a search technique in conjunction with an evaluation metric. Here we rely on a simple filter-based feature selection method, whereby the evaluation metric is the information gain with respect to the class as follows,

$$IG = H(class) - H(class|Attribute),$$

where $H(\boldsymbol{x})$ is the entropy of $\boldsymbol{x}$, given by $H(\boldsymbol{x}) = -\sum_{i=1}^{E} \Pr(\boldsymbol{x}[i]) \ln \Pr(\boldsymbol{x}[i])$.

B. Voronoi Diagram

Based on the subsampled dataset $\boldsymbol{R}$, a Voronoi diagram (Dirichlet tessellation) $\mathcal{V}(\boldsymbol{R})$ is constructed by partitioning the dataset space into L Voronoi regions $VR(\boldsymbol{r}_n, \boldsymbol{R}) \in \mathcal{V}(\boldsymbol{R})$ whose centers $\boldsymbol{r}_n$ are instances in $\boldsymbol{R}$. A brief rigorous definition of the Voronoi diagram is provided below [27]. If $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^M$, then the bisector of $\boldsymbol{x}$ and $\boldsymbol{y}$,

$$B(\boldsymbol{x}, \boldsymbol{y}) = \{\boldsymbol{s} \in \mathbb{R}^M | \, ||\boldsymbol{x} - \boldsymbol{s}||_p = ||\boldsymbol{y} - \boldsymbol{s}||_p\} \tag{1}$$

is the perpendicular line through the center of the line segment $\overline{\boldsymbol{xy}}$, where $\overline{A}$ denotes the closure of some set A [28], and $||.||_p$ corresponds to the p-norm distance, defined as $||x||_p = (\sum_{i=1}^{M} x_i^p)^{1/p}$. The bisector separates the half plane $D(\boldsymbol{x}, \boldsymbol{y})$ defined as:

$$D(\boldsymbol{x}, \boldsymbol{y}) = \{\boldsymbol{s} \in \mathbb{R}^M | \, ||\boldsymbol{x} - \boldsymbol{s}||_p < ||\boldsymbol{y} - \boldsymbol{s}||_p\}. \tag{2}$$

The corresponding Voronoi region of $\boldsymbol{x}$ with respect to $\boldsymbol{R}$ is thus written as:

$$VR(\boldsymbol{x}, \boldsymbol{R}) = \bigcap_{\boldsymbol{y} \in \boldsymbol{R}, \boldsymbol{y} \neq \boldsymbol{x}} D(\boldsymbol{x}, \boldsymbol{y}). \tag{3}$$

The Voronoi diagram $\mathcal{V}(\boldsymbol{R})$ is the collection of all Voronoi regions. The naive method for computing a Voronoi diagram is by the brute force approach, using the following basic definition [29]:

$$VR(\boldsymbol{r}_n, \boldsymbol{R}) = \{\boldsymbol{x} \in R^M | \forall j \neq n, \, ||\boldsymbol{r}_n - \boldsymbol{x}||_p < ||\boldsymbol{r}_j - x||_p\}. \tag{4}$$

Several more efficient algorithms exist to compute the Voronoi diagram, such as the incremental construction, divide & conquer, and plane sweep methods. According to Theorems 3.3 and 3.4 in reference [27], the divide & conquer algorithm and the plane sweep constructs a Voronoi diagram of n points within time $O(n \log n)$ and linear space, in the worst case, where both bounds are optimal.

Here, we use the Euclidean distance measure (i.e., $p = 2$). However, it is worth mentioning that variety of other distance metrics can be chosen depending on the nature of the problem and the data. For example, for continuously-valued attributes, suitable distance metrics include the Minkowski, Mahalanobis, Chebychev, Camberra, Quadratic, Correlation, Chi-square distances, as previously reviewed [19]. If the data are nominal, then the overlap metric or the value difference metric (VDM) can be used [19]. In addition, cheap distance metrics can be used for highly-dimensional datasets. For example, only the most significant features can be chosen for the distance metric, which in turn drastically reduces the time latency when computing the distances. Moreover, the construction of an approximate Voronoi diagram can be achieved using kd-trees [30], [31]. This stage is critical because it reduces the complexity of the next clustering stage, where we assume that data points which are far apart do not have an effect on each other. Importantly, our framework allows parallel condensation to be performed on the different Voronoi regions.

C. Clustering

After constructing the Voronoi diagram, $\mathcal{V}(\boldsymbol{R})$, we overlay it with the original dataset $\boldsymbol{D}$. Now our goal is to fetch representative centroids from each Voronoi region, for which the collection of all centroids comprises the reduced dataset $\boldsymbol{C}$. The problem is now reduced to the two following queries: (1) Which clustering algorithm do we choose? (2) How many centroids per Voronoi region should be fetched?

Since we assume this data condensation framework is applicable to all types of datasets, namely labeled, unlabeled, and semi-labeled, then utilizing the labels of the instances for the clustering algorithm is not assumed. One of the most common clustering preferences is the k-means clustering, which minimizes the within-cluster sum of squares measure. This optimization problem is generally solved by two iterative phases via the use of the expectation-maximization (EM) algorithm [32]. The k-means algorithm performs the hard assignment of points to clusters, which might not be an appropriate idea for the points that lie midway between the cluster centers and the decision boundary [33]. A more accurate approach is to adopt the soft clustering alternative of the k-means, which is realized by assuming that the set of data points in each Voronoi region ($VR(\boldsymbol{r}_n, \boldsymbol{R})$) is distributed according to some probabilistic (generative) model, such as the GM.

Variational Bayesian (VB) or variation inference is a Bayesian treatment technique that eliminates many challenges that arise when working with the maximum likelihood and maximum log-likelihood approach. First, maximizing the log-likelihood function through the EM algorithm is an ill-posed problem because of some singularities that may occur, especially in highly non-parametric datasets such as which exist in ML literature. Second, the EM algorithm is susceptible to overfitting due to the inherent use of the maximum likelihood approach [33]. Third, The vanilla EM algorithm does not provide a way of determining the optimal number of clusters, K_n. Here, we adopt the variational inference machinery based on the GM model [33, Section (10.2)]. The general technique can be generalized to other exponential distributions; interested readers are referred to [34], [35].

In this paper, for each Voronoi region, the VB algorithm is initialized with $\sqrt{\frac{U_n}{2}}$ clusters, where U_n is the number of datapoints in the nth Voronoi region. In the VB algorithm [33, Section (10.2)], the hyper parameter, α_0, reflects our confidence of the assigned initial conditions. The larger the value of α_0, the more influence the prior has on the posterior distribution. Setting $\alpha_0 < 1$ induces sparsity, if any, in the corresponding mixing coefficients $\boldsymbol{\pi}$. The VB algorithm therefore offers a complexity-free method to determine the effective number of centroids in a Voronoi region. In addition, the VB algorithm is neither susceptible to overfitting nor is vulnerable to potential singularities, in contrast to the likelihood approach.

III. Summary of the subsampling Framework

The data condensation framework consists of three main stages. The following pseudocode in Algorithm 1 illustrates the proposed framework.

Algorithm 1: The proposed subsampling framework

```
Procedure ReduceDataset($\boldsymbol{D}$, $S$);
Output: $\boldsymbol{C}$
$\boldsymbol{R} \leftarrow \mathrm{StratifiedSampling}(\boldsymbol{D}, S)$;            ▷ Stage 1
$\mathcal{V}(\boldsymbol{R}) \leftarrow \mathrm{ConstructVoronoi}(\boldsymbol{R})$;          ▷ Stage 2
Overlay $\boldsymbol{D}$ onto $\mathcal{V}(\boldsymbol{R})$;
                                                   ▷ Stage 3
while $n \leq S \times N$ do
    $\boldsymbol{C} \leftarrow \boldsymbol{C} +$ VBGM($VR(\boldsymbol{r}_n, \boldsymbol{R})$,$\sqrt{\frac{U_n}{2}}$);
end
```

In the pseudocode, the framework requires the Dataset $\boldsymbol{D}$ and an initial (rough) sampling percentage S has a value between 0 and 1. During the first stage, stratified random sampling is called to select a representative set $\boldsymbol{R}$ from $\boldsymbol{D}$, where the number of selected instances corresponds to the specified percentage S. Due to the imbalance in the benchmark datasets, all instances from the minority class are selected. During stage 2, based on $\boldsymbol{R}$, a Voronoi diagram $\mathcal{V}(\boldsymbol{R})$ is constructed by partitioning the dataset space into $L = S \times N$ Voronoi regions $V(\boldsymbol{r}_n, \boldsymbol{R}) \in \mathcal{V}(\boldsymbol{R})$. Finally during stage 3, the original instances from $\boldsymbol{D}$ are overlaid back onto $\mathcal{V}(\boldsymbol{R})$ and the VBGM clustering algorithm is used to fetch at most $\sqrt{\frac{U_n}{2}}$ centroids from each nth Voronoi region. We set α_0 to 0.01, which usually results in the selection of fewer than $\sqrt{\frac{U_n}{2}}$ centroids. Therefore, although the sampling percentage is specified at the beginning of the algorithm, the final sampling percentage is most slightly lower, where the highest worst case condensation percentage is $\frac{\sum_{n=1}^{L} K_n}{N}$, where $K_n \leq \sqrt{\frac{U_n}{2}}$.

IV. Framework Validation and Testing

We tested the proposed data condensation framework by reducing three large benchmark datasets and studying their predictive performance of the framework against the RF and RBFN classifiers, using 10-fold cross validation. The predictive performance measures included the following: accuracy, F-measure, and training time. We found that the framework outperformed the existing stratified sampling technique considerably at low sampling percentages, and minimized the variability of the predictive performance measures.

Dataset	# of Instances (N)	Dimensionality (M)	Class (0:1)
cod-RNA	22,660	8	22,000:660
ds1.10	26,733	10	25,929:804
ds1.100	26,733	100	25,929:804

TABLE I
Summary of benchmark datasets.

A. Benchmark Datasets

The first two datasets (ds1.10 and ds1.100) were downloaded from their online source [21]. These two datasets are imbalanced dense datasets of 10 and 100 dimensions, respectively. They were obtained by applying principal component analysis (PCA) to a sparse dataset (ds1) which is provided by the National Cancer Foundation [36]. Each row in the original ds1 dataset comprises a chemical or biological experimentation in which the result is binary. The third benchmark dataset (cod-RNA) was developed for detecting non-coding RNAs [20], [37]. Table I summarizes the properties of the three benchmark datasets.

B. Evaluation and Analysis

We study the relation between the sampling percentage and various evaluation measures including the accuracy (5), F-measure (6) and training time. Each feature vector may be assigned to either *positive* or *negative* class. A positive instance is counted as true-positive (TP) if it is correctly classified, and false-positive (FP) otherwise. A negative instance is counted as true-negative (TN) if it is correctly classified, and false-negative (FN) otherwise. The accuracy is given by:

We investigated the relationship between the sampling percentage and various evaluation measures including the accuracy (5), F-measure (6) and training time. Each feature vector was assigned to either positive or negative. A positive instance is counted as true-positive (TP) if it is correctly classified, and as false-positive (FP) otherwise. The accuracy is given by

$$Acc = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}. \tag{5}$$

Another evaluation metric that is particularly useful for imbalanced data is the F-measure, which is the harmonic mean of the precision and recall, expressed as:

$$F = 2\frac{\mathrm{precision.recall}}{\mathrm{precision} + \mathrm{recall}}, \tag{6}$$

where $\mathrm{precision} = \frac{TP}{TP+\mathrm{FP}}$, and $\mathrm{recall} = \frac{\mathrm{TP}}{\mathrm{TP+FN}}$. We also measured the variances of the accuracy, i.e. σ^2_{Acc}, and the F-measure, σ^2_{F1}, to ensure the generality of our approach. We also compared the performance of the proposed framework against the naive stratified sampling strategy, whereby the condensed dataset is simply the output of stage 2. The average accuracy and F-measure of the RF classifier versus the sampling percentage for the cod-RNA and ds1.10 ae shown in Fig. 1 and Fig. 2, respectively.

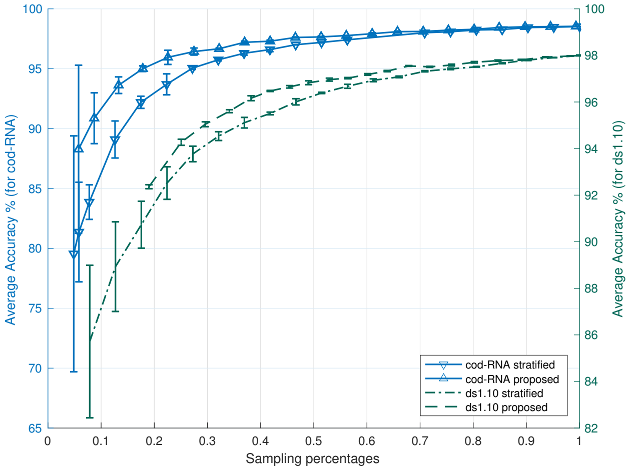

Fig. 1. Average accuracy of the RF classifier with 10-fold cross validation for the cod-RNA and ds1.10 benchmark datasets with the proposed method and the stratified sampling.

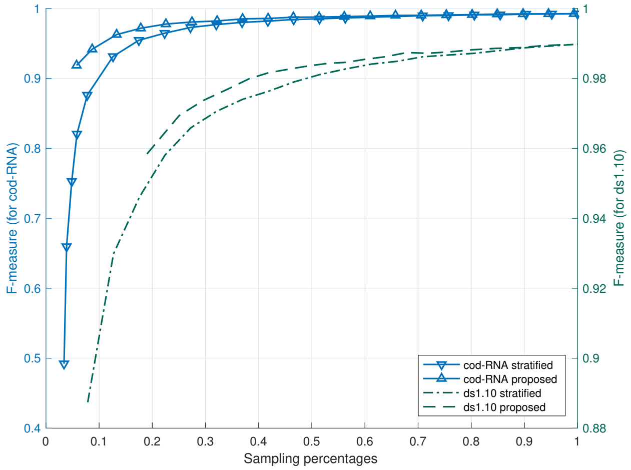

Fig. 2. Average F-measure of the RF classifier with 10-fold cross validation for the cod-RNA and ds1.10 benchmark datasets with the proposed method and the stratified sampling.

Our proposed condensation framework shifted the average accuracy curve by a considerable amount for the two datasets while reducing the variances of the measures (Fig. 1). For example, if we sample only 10% of the cod-RNA dataset using stratified sampling alone, the average accuracy was $Acc = 84\%$ with a variance of $\sigma^2_{Acc} = 1.5$, whereas using our condensation framework, the accuracy increased to $Acc = 94\%$ with a variance reduced to $\sigma^2_{Acc} = 0.4$. Furthermore, the predictive performance for only 10% of the dataset was very close to maximum accuracy ($Acc_{max} = 98\%$). Similar improvements were observed for the F-measure (in Fig. 2). A consistent improvement was observed for the ds1.100 dataset, but it was not shown for the sake of clarity.

Next, we test our framework with the RBFN classifier.

Dataset	Sampling	$F1$	Acc	σ^2_{F1}	σ^2_{Acc}
cod-RNA	-	0.9853	97.114	1.71E-07	0.0035
ds1.10	-	0.9853	97.587	5.57E-07	0.0460
ds1.100	-	0.9876	97.849	1.19E-06	0.0426
cod-RNA	Stratified	0.9339	88.788	2.91E-05	0.4195
ds1.10	Stratified	0.9270	87.748	5.47E-05	2.2967
ds1.100	Stratified	0.9430	90.305	0.000110	1.7797
cod-RNA	Proposed	0.94431	90.327	5.01E-05	0.3715
ds1.10	Proposed	0.94788	90.726	1.81E-05	0.4149
ds1.100	Proposed	0.96279	93.268	5.54E-05	1.0440

TABLE II
PREDICTIVE PERFORMANCE OF THE RBFN CLASSIFIER WITH SAMPLING PERCENTAGES OF 100% AND 17% WITH THE STRATIFIED SAMPLING AND THE PROPOSED METHOD.

Likewise, our framework shows better performance than the stratified sampling for the three datasets. Table II provides a summary of the obtained predictive performance measures for the proposed subsampling framework and the naive subsampling when the sampling percentage is 100% and 17%.

V. Conclusion

We have developed a clustering-based data condensation framework which relies on the use of stratified sampling, Voronoi diagrams, and the variational inference machinery for clustering. Three large imbalanced benchmark datasets were tested, and we confirmed that the proposed data condensation framework has the ability to improve the predictive performance of machine learning classifiers at low sampling percentages. The proposed framework is scalable to large datasets as it allows for parallel clustering mechanisms. The proposed framework stimulates several interesting questions for further research. One can further investigate the effectiveness of each stage, and how they can be improved in the context of problem-dependent or -independent frameworks. Moreover, the proposed mechanism decouples the different stages for efficiency. However, it can yield enhanced predictive performance when the different stages are jointly considered.

References

[1] O. Y. Al-Jarrah, P. D. Yoo, S. Muhaidat, G. K. Karagiannidis, and K. Taha, "Efficient machine learning for big data: A review," *Big Data Research*, vol. 2, no. 3, pp. 87 – 93, 2015.

[2] R. L. Figueroa, Q. Zeng-Treitler, S. Kandula, and L. H. Ngo, "Predicting sample size required for classification performance," *BMC Medical Informat. and Decision Making*, vol. 12, no. 1, p. 8, 2012.

[3] S. H. Mirisaee, A. Douzal, and A. Termier, "Selecting representative instances from datasets," in *IEEE Int. Conf. DSAA*, Oct 2015, pp. 1–10.

[4] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Tran. Inf. Theory*, vol. 13, no. 1, pp. 21–27, jan 1967.

[5] G. Gates, "The reduced nearest neighbor rule (corresp.)," *IEEE Trans. Inf. Theory*, vol. 18, no. 3, pp. 431–433.

[6] G. Ritter, H. Woodruff, S. Lowry, and T. Isenhour, "An algorithm for a selective nearest neighbor decision rule (corresp.)," *IEEE Tran. Inf. Theory*, vol. 21, no. 6, pp. 665–669, Nov 1975.

[7] C.-H. Chou, B.-H. Kuo, and F. Chang, "The generalized condensed nearest neighbor rule as a data reduction method," in *Proc. ICPR*, Aug 2006, pp. 556–559.

[8] Y. Li, Z. Hu, Y. Cai, and W. Zhang, "Support vector based prototype selection method for nearest neighbor rules," in *Advances in Natural Computation*. Berlin, Heidelberg: Springer, 2005, pp. 528–535.

[9] Q. Yang and G. Webb, Eds., *PRICAI 2006: Trends in artificial intelligence*, ser. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2006, vol. 4099.

[10] J. A. Olvera-López, J. A. Carrasco-Ochoa, J. F. Martínez-Trinidad, and J. Kittler, "A review of instance selection methods," *Artificial Intelligence Review*, vol. 34, no. 2, pp. 133–143, 2010.

[11] J. Riquelme, J. Aguilar-Ruiz, and M. Toro, "Finding representative patterns with ordered projections," *Pattern Recognition*, 2003.

[12] T. Raicharoen and C. Lursinsap, "A divide-and-conquer approach to the pairwise opposite class-nearest neighbor (POC-NN) algorithm," *Pattern Recognition Rett.*, 2005.

[13] H. Liu and H. Motoda, "On issues of instance selection," *J. Data Mining and Knowledge Discovery*, 2002.

[14] J. C. Bezdek and L. I. Kuncheva, "Nearest prototype classifier designs: An experimental study," *J. Int. Intell. Syst.*, vol. 16, no. 12, pp. 1445–1473, 2001.

[15] Y. Caises, A. González, E. Leyva, and R. Pérez, "SCIS: Combining instance selection methods to increase their effectiveness over a wide range of domains," in *Proc. Intell. Data Eng. Automated Learning*, E. Corchado and H. Yin, Eds., Berlin, Heidelberg, 2009, pp. 17–24.

[16] B. Spillmann, M. Neuhaus, H. Bunke, E. Pękalska, and R. P. W. Duin, "Transforming strings to vector spaces using prototype selection," in *Structural, Syntactic, and Statistical Pattern Recognition*. Berlin, Heidelberg: Springer, 2006, pp. 287–296.

[17] R. A. Mollineda, F. J. Ferri, and E. Vidal, "An efficient prototype merging strategy for the condensed 1-nn rule through class-conditional hierarchical clustering," *Pattern Recognition*, vol. 35, pp. 2771–2782, 2002.

[18] C. J. Veenman and M. J. T. Reinders, "The nearest subclass classifier: a compromise between the nearest mean and nearest neighbor classifier," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 9, pp. 1417–1429, Sep 2005.

[19] D. R. Wilson and T. R. Martinez, "Reduction techniques for instance-based learning algorithms," *Machine learning*, vol. 38, no. 3, pp. 257–286, 2000.

[20] M. Maalouf and M. Siddiqi, "Weighted logistic regression for large-scale imbalanced and rare events data," *Knowledge-Based Systems*, vol. 59, pp. 142–148, 2014.

[21] P. Komarek, "Datasets." [Online]. Available: http://komarix.org/ac/ds/

[22] T. Yu, T. Jan, S. Simoff, and J. Debenham, "A hierarchical VQSVM for imbalanced data sets InNeural Networks," in *Proc. Joint Conf. Neural Netw.*, 2007, pp. 518–523.

[23] A. Zaknich, "Introduction to the modified probabilistic neural network for general signal processing applications," *IEEE Tran. Signal Pross.*, vol. 46, pp. 1980–1990, Jul 1998.

[24] S. Parthasarathy, "Efficient progressive sampling for association rules," in *IEEE Proc. Data Mining*, Dec 2002, pp. 354–361.

[25] G. H. John and P. Langley, "Static versus dynamic sampling for data mining," in *Proc. Int. Conf. Knowledge Discovery and Data Mining*. AAAI Press, 1996, pp. 367–370.

[26] D. Braha, *Data mining for design and manufacturing: Methods and applications*. Norwell, MA, USA: Kluwer Academic Publishers, 2001.

[27] F. Aurenhammer and R. Klein, "Voronoi diagrams," in *Handbook of Computational Geometry*. Amsterdam: North-Holland, 2000, pp. 201–290.

[28] D. G. Luenberger, *Optimization by vector space methods*. John Wiley & Sons, 1969.

[29] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars, *Computational geometry: Algorithms and applications*, 3rd ed. Santa Clara, CA, USA: Springer-Verlag, 2008.

[30] S. Arya and T. Malamatos, "Linear-size approximate Voronoi diagrams," in *Proc. ACM-SIAM Symp. Discrete Algorithms*, Philadelphia, PA, USA, 2002, pp. 147–155.

[31] S. Arya, T. Malamatos, and D. M. Mount, "Space-efficient approximate Voronoi diagrams," in *Proc. ACM Symp. Theory Comput.*, 2002, pp. 721–730.

[32] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Royal Statistical Soc.*, vol. 39, no. 1, pp. 1–38, 1977.

[33] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[34] D. G. Tzikas, C. L. Likas, and N. P. Galatsanos, "The variational approximation for Bayesian inference," *IEEE Signal Process. Mag.*, vol. 25, no. 6, pp. 131–146, 2008.

[35] M. J. Beal, "Variational algorithms for approximate Bayesian inference," Ph.D. dissertation, University of London, 2003.

[36] P. Komarek, "Logistic regression for data mining and high-dimensional classification," Ph.D. dissertation, Pittsburgh, PA, USA, 2004.

[37] A. V. Uzilov, J. M. Keegan, and D. H. Mathews, "Detection of non-coding RNAs on the basis of predicted secondary structure formation free energy change," *BMC bioinformatics*, vol. 7, 2006.