Principal Sample Analysis for Data Reduction

Benyamin Ghojogh
Department of Electrical and Computer Engineering
Machine Learning Lab, University of Waterloo
Waterloo, ON, Canada
bghojogh@uwaterloo.ca

Mark Crowley
Department of Electrical and Computer Engineering
Machine Learning Lab, University of Waterloo
Waterloo, ON, Canada
mcrowley@uwaterloo.ca

***Abstract*—Data reduction is an essential technique used for purifying data, training discriminative models more efficiently, encouraging generalizability, and for using less storage space for memory-limited systems. The literature on data reduction focuses mostly on dimensionality reduction, however, data sample reduction (i.e. removal of data points from a dataset) has its own benefits and is no less important given growing sizes of datasets and the growing need for usable data analysis methods on the network edge. This paper proposes a new data sample reduction method, Principal Sample Analysis (PSA), which reduces the number (population) of data samples as a preprocessing step for classification. PSA ranks the samples of each class considering how well they represent it and enables better discriminative learning by using the sparsity and similarity of samples at the same time. Data sample reduction then occurs by cutting off the lowest ranked samples. The PSA method can work alongside any other data reduction/expansion and classification method. Experiments are carried out on three datasets (WDBC, AT&T, and MNIST) with contrasting characteristics and show the state-of-the-art effectiveness of the proposed method.**

***Index Terms*—Data reduction, preprocessing, principal sample analysis (PSA), storage efficiency, data ranking**

I. Introduction

Despite increasing resources to store, train, and process big datasets, it is often still desirable, or even necessary, to reduce data. There are several motivations for data reduction, such as (I) finding purer patterns or samples needed for classification and pattern recognition, (II) excluding dummy attributes or samples which do not carry much information from data, and (III) reducing data for the sake of better space and storage efficiency. There exist various **dimensionality reduction** methods, such as random projection, Fisher Linear Discriminant Analysis (FLDA) [1], [2], Principal Component Analysis (PCA) [3], Principal Factor Analysis (PFA), Independent Component Analysis (ICA), Multi-Dimensional Scaling (MDS) [4], [5], Isomap [6], and Locally Linear Embedding (LLE) [7].

Data reduction is mostly synonymous in the literature with dimensionality reduction. However, **data sample reduction**, which reduces the number of samples in a dataset, is an orthogonal but equally important technique to consider. This aspect of data reduction is somewhat under-represented in the literature except for some sampling methods such as Simple Random Sampling (SRS), Sorting by Distance from Mean (SDM), Stratified Sampling [8], [9], and Separate Sampling [10]. Most of these methods were initially developed for the application of survey statistics [11] which often have the goal of learning a parameter of interest about the population [12]. **Our goal in this work is to provide a reduced sample of the data points which enables better representation and discrimination of classes.** There exist several motivations for our work:

- There usually exists some dummy information in the data which is not completely useful (or is sometimes even destructive) for discrimination or representation of data. In other words, the data samples do not contribute equal amounts of information to learning a discriminative model and thus could be sorted by this information if it can be quantified.
- In some cases, some part of data results in better discrimination of classes than others.
- In some applications in edge computing, low-battery embedded systems or space exploration, there is limited possibility for storage or energy. In these domains it is useful to store or transmit a portion of data which is its best representation.

TABLE I: The landscape of data reduction/expansion.

		# samples		
		increase	**decrease**	**no change**
# dimensions	**increase**	Data Augmentation + kernel trick	PSA, SRS, ... + kernel trick	kernel trick
	decrease	Data Augmentation + PCA, Isomap, ...	PSA, SRS, ... + PCA, Isomap, ...	PCA, Isomap, FLDA, ...
	no change	Data Augmentation	**PSA**, SRS, SDM, stratified sampling, ...	Original

Our proposed algorithm, **Principal Sample Analysis (PSA)**, is a method for reducing the number of samples for the goal of purifying the training set for classification, ranking the samples, and better storage efficiency. One can imagine the space of all data reduction approaches by considering methods that reduce the data dimensionality or data population size. Table I summarizes this landscape as we see it with nine different categories based on the fact that either the dimensionality or population size of the data can be reduced or increased. Both dimensionality reduction (e.g., PCA) and expansion (e.g., kernel trick) have many use cases [13]. Data expansion in terms of population is referred to as **data augmentation** in the literature [14], [15] and is essential to scaling some modern deep learning methods. The PSA method stands in the category

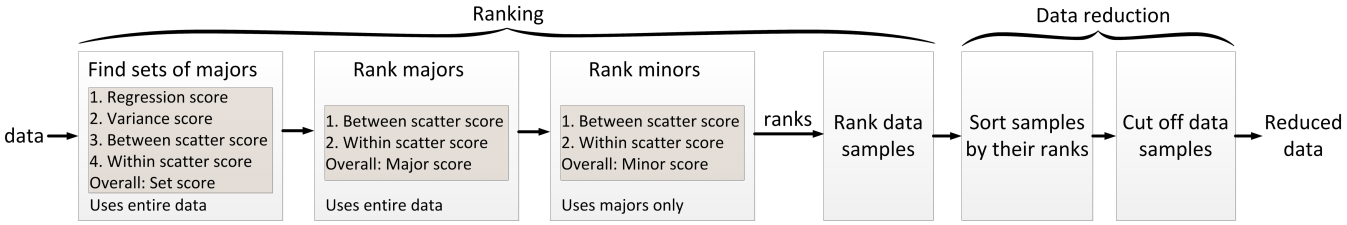


Fig. 1: Overall structure the proposed data reduction method

of reduction in the number (population) of samples while the dimension of data does not change. Note that PSA can be used with any other data reduction and expansion method, such as PCA, as well as any classification algorithm.

In the proposed method, the samples of every class are ranked from best to lowest with respect to representation of class and discrimination of classes. The samples of every class are partitioned into a **major** or **minor** set for each class. A high-level summary of the PSA algorithm is illustrated in Fig. 1. The set of major samples are found primarily using **set scores** obtained by regression, variance, between-class-scatter, and within-class-scatter scores. Thereafter, the samples of the found major set are ranked by their **major scores** obtained by between-class and within-class scatter scores. In every class, all the samples which are not in a major set are minor samples. The minor samples are then ranked based on their **minor scores** considering the major samples of classes. After finding the ranks of samples, the samples of classes are sorted by the obtained ranks and the lowest ranked samples are cut off to reduce the data. The remaining top ranked samples we call **principal samples**. The relative size of the major vs. minor sets is a hyperparameter of the algorithm.

The remainder of this paper is as follows. Section II introduces the PSA algorithm by explaining its stages in detail. A two-dimensional visualization of the performance of PSA is illustrated in Section III for a better interpretation of how it works. Section III introduces the utilized datasets and reports the experimental results which show PSA's superior performance against other sampling methods. Finally, Section IV concludes the paper and discusses future directions.

II. Methodology

The PSA algorithm, works as a preprocessing step before classification, regression, or data mining. In a classification problem, there exist C classes indexed by $j = \{1, \dots, C\}$. The N training samples are denoted by X, and the N^j training samples from the j^{th} class are denoted by X^j. In this paper, the superscript determines the class of sample unless mentioned otherwise. The PSA algorithm consists of four stages: preprocessing, finding sets of major samples, ranking major samples, and ranking minor samples.

A. Preprocessing

The PSA algorithm, while being a preprocessing method itself, might require preprocessing as the first step based on the conditions of data. As is explained in the next section, PSA applies regression on the samples of every class. However, this requires at least $(D-1)$ samples in every class to regress data on a $(D-1)$-dimensional space of samples of the class. Therefore, the number of samples of every class must be at least $(D-1)$. If this condition does not exist for a class, either the number of samples of the class should be increased or the dimensionality of data should be decreased to fulfill the condition. There are several possible solutions: (I) linear (e.g., PCA) or non-linear (e.g., Isomap) dimensionality reduction, (II) data augmentation, or (III) adding Gaussian noisy samples to data. The first approach reduces dimensionality and the second and third ones increase the number of samples. It is noteworthy that in contrast to data augmentation, which can be done differently for different classes, dimensionality reduction must be applied to all classes even if there exist a sufficient number of samples in one of the classes.

B. Finding Sets of Major Samples

1) Regression Score: A good representative set of samples should contain samples which can predict all the data points in the class with minimum error. Inspired by RANdom SAmple Consensus (RANSAC) [16], N_M^j samples of the j^{th} class, denoted by X_M^j, are randomly selected from the samples X^j for several iterations I_{RANSAC}. Note that because of preprocessing, the number of samples of class (N^j) is at least D. If it is equal to D, only one iteration of RANSAC suffices and thus all samples of the class fall in the best set (in other words, $N_M^j = N^j$).

As in RANSAC, a regression method such as linear regression [17] is applied to the selected samples for several iterations, but with a difference. In our problem, there does not exist any label as required by regression. To overcome this challenge, regression is performed $(D-1)$ times where each one of the dimensions is considered once as the label for regression and the rest of dimensions form the observations for regression. The reason of having $(D-1)$ iterations is that the information of the D^{th} iteration exists in the previous iterations and is thus redundant. Suppose the vector containing just the dimension d of samples X_M^j is denoted by $\mathcal{X}_M^j[d]$, and $\mathcal{X}_M^j[-d]$ denotes the $(N_M^j \times D)$-size matrix $\left[\mathbf{1}_{N_M^j \times 1}, \mathcal{X}_M^j[1], \dots, \mathcal{X}_M^j[i], \dots, \mathcal{X}_M^j[D]\right]$ of the remaining dimensions where $i \in \{1, \dots, D\} \backslash \{d\}$. Similar notations $\mathcal{X}^j[d]$ and $\mathcal{X}^j[-d]$ exist for all the samples of j^{th} class X^j. In every iteration of RANSAC, the regression is performed on all N^j

samples of the class and also on the N_M^j major samples. The regression coefficients are calculated as,

All samples:

$$\beta_a^j = \Big((\mathcal{X}^j[-d])^T \mathcal{X}^j[-d]\Big)^{-1} (\mathcal{X}^j[-d])^T \mathcal{X}^j[d], \quad (1)$$

Major samples:

$$\beta_M^j = \Big((\mathcal{X}_M^j[-d])^T \mathcal{X}_M^j[-d]\Big)^{-1} (\mathcal{X}_M^j[-d])^T \mathcal{X}_M^j[d]. \quad (2)$$

Note that for $\Big((\mathcal{X}_M^j[-d])^T \mathcal{X}_M^j[-d]\Big)$ to be full-rank and invertable, at least $(D-1)$ samples are required as mentioned before. Moreover, if the selected samples form a good representative of the all the data for class j, the two vectors β_a^j and β_M^j are closely parallel and have cosine close to 1. Therefore, the regression score $s_M^{j,R}$ of major samples is calculated as,

$$s_M^{j,R} = \cos(\beta_M^j, \beta_a^j) = \frac{(\beta_M^j)^T \beta_a^j}{||\beta_M^j||_2 ||\beta_a^j||_2}, \quad (3)$$

where $||.||_2$ denotes l_2-norm. To have better intuition of why the two vectors β_a^j and β_M^j should be parallel, note that the β vector of regression can be interpreted as a normal vector to the hyperplane fitted to the data by the regression. In the following, we briefly mention the proof of this interpretation.

Proof. Suppose there exist N D-dimensional observations $X = [x_1, \ldots, x_N]$ and N labels $Y = [y_1, \ldots, y_N]^T$. Considering $\mathcal{X} = [\mathbf{1}, X^T]$, the regression formulas are [17],

$$\beta = (\mathcal{X}^T \mathcal{X})^{-1} \mathcal{X}^T Y, \quad (4)$$

where $\beta = [\beta_0, \beta_1, \ldots, \beta_D]^T = [\beta_0, \beta_*^T]^T$. Suppose two data samples $[1, x_1^T]^T$ and $[1, x_2^T]^T$ are on the hyperplane with equation $\beta_*^T x + \beta_0 = 0$. Therefore, $\beta_*^T x_1 + \beta_0 = \beta_*^T x_2 + \beta_0$ and $\beta_*^T (x_1 - x_2) = 0$ which means $\beta_* \perp (x_1 - x_2)$. Noting that the valuable information of intercept exists in β_0, the whole vector β can be interpreted as the normal vector. □

Therefore, when β_a^j and β_M^j are closely parallel, it means that the major samples are representing a hyperplane almost parallel to the hyperplane represented by all samples of class.

2) Variance Score: The major samples are supposed not to be close to each other because very nearby samples share similar information and therefore are redundant. On the other hand, sampling from different sparse points in the space of data gives a good estimation of the distribution of data. To make the major samples far from each other, the scatter of major samples of class j, denoted by $S_M^{j,\nu}$, is calculated as,

$$S_M^{j,\nu} = \sum_{i=1}^{N_M^j} (x_{M,i}^j - \overline{x}_M^j)(x_{M,i}^j - \overline{x}_M^j)^T, \quad (5)$$

where $x_{M,i}^j$ is the i^{th} sample in the set X_M^j, and $\overline{x}_M^j = (1/N_M^j) \sum_{i=1}^{N_M^j} x_{M,i}^j$ is the mean of samples in set X_M^j. As the eigenspace of the scatter matrix carries information about the variance of the data [17], [18], the variance score can be,

$$s_M^{j,\nu} = \sum \mathbf{diag}(\Lambda(S_M^{j,\nu})) \propto \mathrm{Var}(x_M^j), \quad (6)$$

where $\Lambda(S)$ is the matrix whose diameter contains eigenvalues of matrix S. Note that $\sum \mathbf{diag}(\Lambda(S))$ is equal to $\mathbf{trace}(S)$. The larger the variance of major samples, the larger this score is.

3) Between Scatter Score: The major samples are supposed to be farther from the other classes for the sake of more discrimination. Therefore, the between scatter [1], [17] of major samples of the j^{th} class from the other classes is calculated,

$$S_M^{j,B} = \sum_{c=1, c\neq j}^{C} \sum_{k=1}^{N^c} w_k^c (\overline{x}_M^j - x_k^c)(\overline{x}_M^j - x_k^c)^T, \quad (7)$$

where C is the number of classes, x_k^c is the k^{th} sample of class c, $\overline{x}_M^j$ is the mean of major samples in class j, and w_k^c is the weight associated with x_k^c, calculated as,

$$w_k^c = \frac{1}{2}\big(1 + \cos(x_k^c, \overline{x}^c)\big) = \frac{1}{2}\big(1 + \frac{x_k^c . \overline{x}^c}{||x_k^c||_2 ||\overline{x}^c||_2}\big), \quad (8)$$

where $\overline{x}^c$ is the mean of class c. This weighting, which is in the range $[0, 1]$, gives more weight to the samples close (parallel) to the mean of its own class, and gives less weight to the samples very far from (not parallel to) the mean of its class. Finally, the between scatter score is calculated as,

$$s_M^{j,B} = \sum \mathbf{diag}(\Lambda(S_M^{j,B})). \quad (9)$$

4) Within Scatter Score: Although it is better for the major samples to be sparse, as explained in variance score, they should also be close to each other to *represent the core of the class*. In other words, a trade-off exists between being sparse and compact. The within scatter [1], [17] of the major samples of the j^{th} class is,

$$S_M^{j,W} = \sum_{i=1}^{N_M^j} \sum_{k=1, k\neq i}^{N_M^j} w_{M,k}^j (x_{M,i}^j - x_{M,k}^j)(x_{M,i}^j - x_{M,k}^j)^T, \quad (10)$$

where weight $w_{M,k}^j$ is the same as equation (8) if substituting x_k^c and $\overline{x}^c$ with $x_{M,k}^j$ and $\overline{x}_M^j$ respectively. The within scatter score is,

$$s_M^{j,W} = \sum \mathbf{diag}(\Lambda(S_M^{j,W})). \quad (11)$$

5) Ranking Sets: The score of a set of major samples (set score) is finally calculated as,

$$s_M^j = s_M^{j,R} \times s_M^{j,\nu} \times s_M^{j,B} \times (1/s_M^{j,W}), \quad (12)$$

because a better set should have larger regression score, larger variance score, larger between scatter score, and smaller within scatter score. The algorithm for finding sets of major samples is shown in Algorithm 1. This algorithm is performed for every class j. As mentioned before, sets are randomly sampled from training samples of the class for several iterations (see lines 7 and 9 of Algorithm 1). In every iteration, the set score of selected samples is found and finally the set of samples having the best set score is returned.

Algorithm 1 Finding sets of major samples in PSA

Inputs: X, j
$X^j \leftarrow$ samples of class j in X
if $N^j = D$ **then**
  $I_{\text{RANSAC}} \leftarrow 1$, $X_M^j \leftarrow X^j$
  return X_M^j
$s_{\text{bestSet}} \leftarrow -\infty$
for I_{RANSAC} times **do**
  $X_M^j \leftarrow$ Randomly take N_M^j samples from X^j
  for d from 1 to $D-1$ **do**
    Compute β_a^j and β_M^j
    Compute scores $s_M^{j,R}, s_M^{j,\nu}, s_M^{j,B}, s_M^{j,W}$
    $s_M^j \leftarrow s_M^{j,R} \times s_M^{j,\nu} \times s_M^{j,B} \times \frac{1}{s_M^{j,W}}$
    if $s_M^j > s_{\text{bestSet}}$ **then**
      $s_{\text{bestSet}} \leftarrow s_M^j$
      bestSetSamples $\leftarrow X_M^j$
return bestSetSamples

C. Ranking Major Samples

1) Between Scatter Score: The major samples are supposed to be far from the other classes. The between scatter of a major sample $x_{M,i}^j$ in class j from the samples of other classes is computed as,

$$S_{M,i}^{j,B} = \sum_{c=1,c\neq j}^{C} \sum_{k=1}^{N^c} w_k^c (x_{M,i}^j - x_k^c)(x_{M,i}^j - x_k^c)^T, \quad (13)$$

where weight w_k^c is the same as equation (8). Note that equation (13) tries to capture the distance of major samples in a major set from the samples of other classes, while equation (7) captures the distance of the mean of a major set from the samples of other classes. In other words, equations (13) and (7) are for evaluating the individual major samples and the entire set of major samples, respectively.

The between scatter score of a sample in the set is then found as,

$$s_{M,i}^{j,B} = \sum \mathbf{diag}(\Lambda(S_{M,i}^{j,B})). \quad (14)$$

2) Within Scatter Score: It is better for the major samples to be close to the samples of their own class. The within scatter of a major sample $x_{M,i}^j$ in class j from the samples of its own class is found as,

$$S_{M,i}^{j,W} = \sum_{k=1}^{N^j} w_k^j (x_{M,i}^j - x_k^j)(x_{M,i}^j - x_k^j)^T, \quad (15)$$

where weight w_k^j is the same as equation (8) if substituting x_k^c and $\overline{x}^c$ with x_k^j and $\overline{x}^j$ respectively. The within scatter score of a sample in the set is,

$$s_{M,i}^{j,W} = \sum \mathbf{diag}(\Lambda(S_{M,i}^{j,W})). \quad (16)$$

Algorithm 2 Ranking major samples in PSA

Inputs: X, X_M, j
for i from 1 to N_M^j **do**
  $x_{M,i}^j \leftarrow$ sample i in X_M^j
  Compute scores $s_{M,i}^{j,B}, s_{M,i}^{j,W}$
  $s_{M,i}^j \leftarrow s_{M,i}^{j,B} \times \frac{1}{s_{M,i}^{j,W}}$
ranks $\leftarrow$ **Sort** in descending order according to $s_{M,i}^j$
return ranks

3) Ranking: The algorithm for ranking the major samples is shown in Algorithm 2. The score of a major sample is,

$$s_{M,i}^j = s_{M,i}^{j,B} \times (1/s_{M,i}^{j,W}), \quad (17)$$

because the better sample in the major set is farther from the samples of other classes and is closer to samples of its own class. This score is found for every sample in the best major set of the class and the major samples are ranked by these scores.

D. Ranking Minor Samples

1) Between Scatter Score: It is better for the minor samples to be far from the major samples of other classes. Here, the major samples of other classes are assumed to be proper and purer representatives of their classes, and thus the found major samples are used rather than whole samples of other classes in calculations for ranking minor samples. The between scatter score of a minor sample $x_{m,i}^j$ in class j from the major samples of other classes is found as,

$$S_{m,i}^{j,B} = \sum_{c=1,c\neq j}^{C} \sum_{k=1}^{N_M^c} w_{M,k}^c (x_{m,i}^j - \mathbf{x}_{M,k}^c)(x_{m,i}^j - \mathbf{x}_{M,k}^c)^T, \quad (18)$$

where $\mathbf{x}_{M,k}^c$ is the indexed sample in the sorted major samples according their score (i.e., $\mathbf{x}_{M,1}^c$ has the best rank in set of majors). The weight $w_{M,k}^c$ is,

$$w_{M,k}^c = \frac{N_M^c - k + 1}{N_M^c(N_M^c + 1)/2}, \quad (19)$$

which gives larger weight to better ranked major samples because they are more important to their class. These weights are in the range [0,1] and are summed to one. The between scatter score of a minor sample is,

$$s_{m,i}^{j,B} = \sum \mathbf{diag}(\Lambda(S_{m,i}^{j,B})). \quad (20)$$

2) Within Scatter Score: The minor samples should also be close to the major samples of their own class. Again, here the major samples are taken as good representatives of the class. The within scatter of a minor sample from the major samples of its class is calculated as,

$$S_{m,i}^{j,W} = \sum_{k=1}^{N_M^j} w_{M,k}^j (x_{m,i}^j - \mathbf{x}_{M,k}^j)(x_{m,i}^j - \mathbf{x}_{M,k}^j)^T, \quad (21)$$

Algorithm 3 Ranking minor samples in PSA

Inputs: X, X_M, j, ranks
for i from 1 to N^j **do**
    $x^j_{m,i} \leftarrow$ sample i in $X^j \backslash X^j_M$
    Compute scores $s^{j,B}_{m,i}, s^{j,W}_{m,i}$
    $s^j_{m,i} \leftarrow s^{j,B}_{m,i} \times \frac{1}{s^{j,W}_{m,i}}$
Update ranks $\leftarrow$ **Sort** in descending order according to $s^j_{m,i}$ and ranks
return ranks

where weight w^j_k is the same as equation (19) if substituting N^c_M with N^j_M. The within scatter score of a minor sample is,

$$s^{j,W}_{m,i} = \sum \mathbf{diag}(\Lambda(S^{j,W}_{m,i})). \tag{22}$$

3) Ranking: The algorithm for ranking the minor samples is shown in Algorithm 3. The score of a minor sample is,

$$s^j_{m,i} = s^{j,B}_{m,i} \times (1/s^{j,W}_{m,i}), \tag{23}$$

because a better minor sample is farther from the major samples of other classes and is closer to major samples of its own class. This score is found for every minor sample of class and the minor samples are ranked by the scores. Finally, in every class, the ranks of minor samples are concatenated after the ranks of major samples to have the ranks of all samples in the class.

E. Overall Algorithm

The overall proposed PSA algorithm is shown in Algorithm 4 in which, after a possible preprocessing stage, the best sets of major samples in every class are found primarily. Thereafter, the major samples are ranked in every class. Finally, the minor samples are also ranked and the overall ranks are found by concatenating the minor samples after the major samples. However, note that if, in the preprocessing step, the noisy samples were added to the training set of a class, the ranks of those samples should be omitted and the ranks of original samples updated accordingly. Finally, for every class, the samples are sorted based on their ranks and the $\mathcal{N}$ best samples are selected as the principal samples. Note that $\mathcal{N}$ is a hyperparameter to be specified and could in principle have different values for different classes.

III. Experimental Results

A. Two-Dimensional Visualization

For better understanding the flow and performance of the PSA algorithm, we show the results of a simple two-dimensional case with samples illustrated as well as their ranks in Fig. 2. As shown in Fig. 2a, three classes are created with different characteristics to show the effectiveness of PSA on different conditions. The first class, with population of 30, has different variances in its dimensions, while the second class has 50 samples and its variances are equal in both dimensions. The third class, on the other hand, has 40 samples and is denser

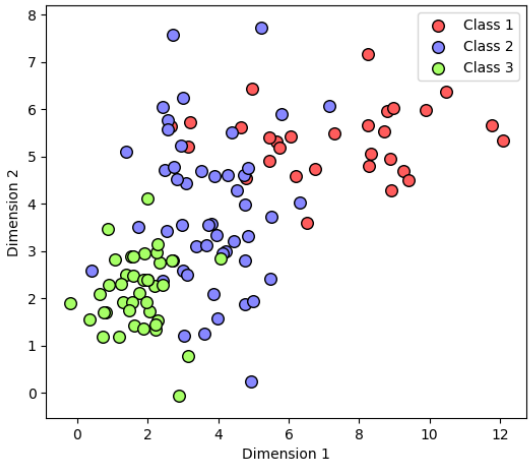
Algorithm 4 PSA

Inputs: X, $\mathcal{N}$
$X \leftarrow$ Preprocessing(X)
for every class j from 1 to C **do**
    $X^j_M \leftarrow$ Find_sets_of_major_samples(X, j)
for every class j from 1 to C **do**
    ranks $\leftarrow$ Rank_major_samples(X, X^j_M, j)
    ranks $\leftarrow$ Rank_minor_samples(X, X_M, j, ranks)
    if noisy samples were added in preprocessing **then**
        ranks $\leftarrow$ Remove the ranks of added noisy samples
    Sort the samples of class based on ranks
    principal samples $\leftarrow$ Select the $\mathcal{N}$ best samples
return ranks and principal samples for every class

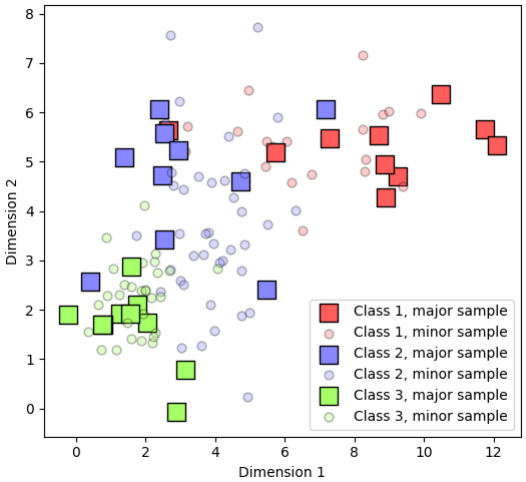
than the other two classes. Figure 2b illustrates the found sets of major samples in the three classes. As expected, the major samples are diverse enough to represent the classes while they are considering the regions of classes. The ranked major samples are shown in Fig. 2c where the larger marker means better rank. The best ranks are determined by considering both similarities and diversities. Figure 2d shows the ranks of minor samples. As can be seen in this figure, the minor samples are ranked according to their distance from major samples of their own and other classes. The final ranked samples obtained by PSA are depicted in Fig. 2e. For the sake of comparison, the samples are also ranked by SDM method. In SDM, the samples of a class are ranked by their distance from the mean of the class in ascending order. The ranks obtained by SDM are shown in Fig. 2f. Comparing figures 2e and 2f demonstrates that SDM concentrates on the mean of each class, while PSA takes into account the representation of classes, discrimination of classes, similarities, and diversities at the same time. As it will be reported in the experiments, PSA strongly outperforms SDM and the intuitive reason can be seen in these two figures.

B. Utilized Datasets

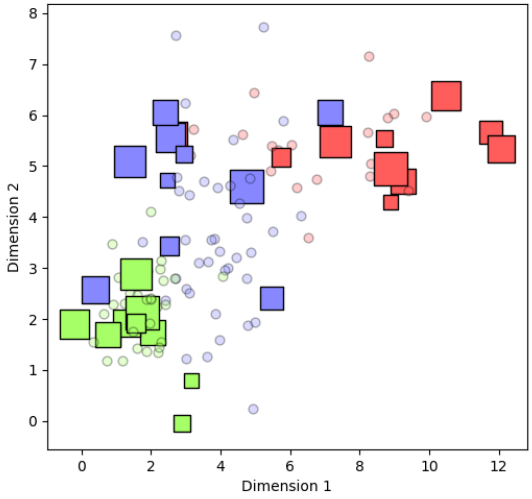
In order to show the effectiveness of the proposed method for data reduction, PSA is applied on three datasets having different characteristics, the Wisconsin Diagnostic Breast Cancer (WDBC) dataset [19], [20], the AT&T face dataset [21], [22], and the MNIST dataset [23], [24]. The WDBC dataset includes 569 samples with 32 dimensions and two classes M and B standing for malignant and benign, respectively. The AT&T face dataset includes facial images of 40 persons each having 10 samples with different poses and expressions. MNIST is a dataset of ten handwritten digits with 60,000 training and 10,000 testing samples. The MNIST dataset is a representative of a small number of classes each having a large number of samples with large dimensionality. The AT&T dataset is representative of datasets having a large number of classes each having a small number of samples with large dimensionality. The WDBC dataset, on the other hand, includes a small number of classes and a small number of dimensions because of not being an image dataset.



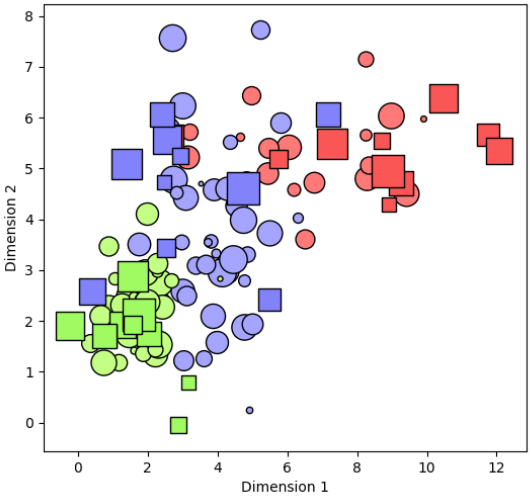
(a) 2D samples of three classes



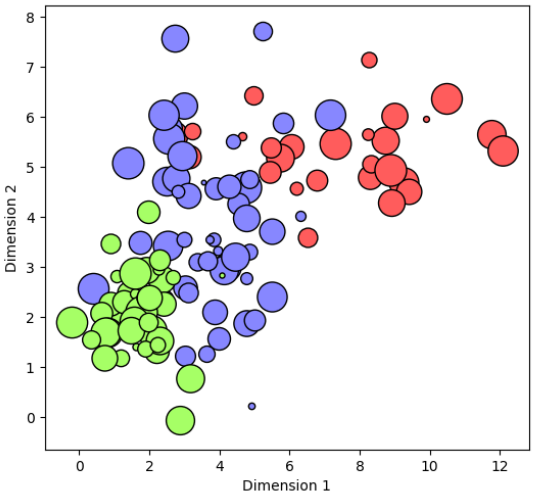
(b) Finding sets of major samples in the classes



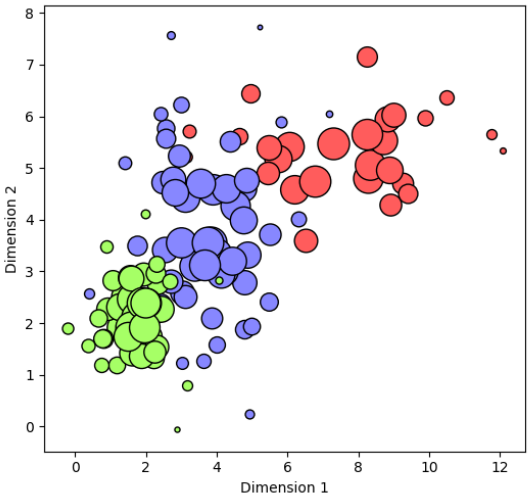
(c) Ranking major samples in each class



(d) Ranking minor samples in each class



(e) Ranked samples by PSA



(f) Ranked samples by SDM

Fig. 2: Two-dimensional visualization of PSA performance and its visual comparison to SDM.

TABLE II: Results on WDBC dataset

	Portion	SVM	LDA	QDA	RF	LR	NB
PSA	20%	**92.98%**	90.76%	**62.57%**	**92.63%**	92.45%	93.80%
	60%	94.56%	**95.49%**	94.61%	**93.21%**	**94.79%**	94.03%
	90%	**95.26%**	95.61%	95.08%	**93.15%**	94.79%	94.03%
SRS	20%	92.78%	**91.13%**	**62.57%**	92.62%	**92.79%**	**94.00%**
	60%	**94.80%**	94.84%	**94.73%**	92.79%	94.71%	**94.08%**
	90%	95.15%	95.33%	**95.19%**	92.98%	**94.90%**	**94.13%**
SDM	20%	90.23%	89.41%	**62.57%**	90.11%	90.00%	88.07%
	60%	89.18%	93.39%	90.58%	86.54%	88.71%	93.09%
	90%	93.91%	**96.08%**	93.80%	92.16%	94.32%	94.09%
Entire Data	100%	95.20%	95.55%	95.20%	92.80%	94.91%	94.26%

TABLE III: Results on AT&T dataset

	Portion	SVM	LDA	QDA	RF	LR	NB
PSA	50%	**87.91%**	**88.16%**	16.50%	**21.83%**	77.66%	**61.00%**
	90%	95.50%	90.33%	23.08%	**34.66%**	**85.83%**	**86.08%**
SRS	50%	87.37%	87.17%	**17.12%**	19.30%	**78.22%**	59.86%
	90%	**95.66%**	**91.50%**	**23.97%**	31.67%	85.35%	85.45%
SDM	50%	76.00%	79.33%	15.58%	16.91%	70.41%	49.33%
	90%	94.50%	88.91%	22.16%	32.16%	82.83%	79.83%
Entire Data	100%	96.66%	92.33%	23.25%	30.75%	86.50%	87.75%

TABLE IV: Results on MNIST dataset

	Portion	SVM	LDA	QDA	RF	LR	NB
PSA	6%	47.97%	**84.52%**	**94.83%**	**61.14%**	86.24%	**84.86%**
	50&	65.42%	84.49%	**95.01%**	**61.47%**	86.80%	**85.46%**
	90%	71.90%	**84.55&**	94.95%	**61.76%**	**86.85%**	85.32%
SRS	6%	52.70%	84.32%	94.36%	60.07%	**86.33%**	84.83%
	50%	69.91%	**84.55%**	94.98%	60.30%	**86.85%**	85.28%
	90%	72.13%	**84.55%**	**95.01%**	60.44%	**86.85%**	**85.34%**
SDM	6%	**64.46%**	77.97%	74.33%	57.27%	73.95%	68.69%
	50%	**71.08%**	81.31%	87.48%	57.43%	82.73%	78.44%
	90%	**75.21%**	83.95%	93.91%	60.20%	86.71%	84.14%
Entire Data	100%	76.22%	84.51%	95.03%	60.67%	86.78%	85.30%

C. Experiments

We compare PSA with SRS, SDM, and the entire dataset (100% of samples). In SRS, all samples of a class have equal probability of selection and the sampling from instances of a class is without replacement [8], [9]. In all experiments of SRS, sampling is performed 20 times and the average result is considered as the result of that fold in cross validation. In SDM, the samples of a class are ranked by their distance from the mean of the class in ascending order. The selection of samples is done from the best ranked samples which are closest to the mean. Note that for PSA, SRS, and SDM, we experiment with different amounts of data reduction, retaining 6%, 20%, 60% (or 50%), 90%, and 100% of the data before carrying out classification. We set the appropriate portions according to the size of each dataset. Six different classifiers are used for verifying the effectiveness of the proposed method when using any classifier; we use Support Vector Machines (SVM), Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), Random Forest (RF), Logistic Regression (LR), and Gaussian Naive Bayes (NB).

1) Experiments on the WDBC dataset: For the WDBC dataset, data was shuffled and then split into train/test sets using stratified sampling [8] in 10-fold cross validation with 0.3 for size portion of test set. The average accuracy rates of the folds are reported in Table II. In tables, bold indicates the best result for each portion for a given data reduction/classifier combination. As seen in Table II, using PSA while retaining 90% of the data even outperforms using the entire dataset for SVM, LDA, and RF classifiers. This shows that not all the training samples have a constructive impact on training. Also note that for RF and LR classifiers using PSA, the accuracy of portion 60% outperforms portion 90%. This shows that there exists some dummy training samples which can be removed especially for the sake of space efficiency. In most of the cases for any portion and any classifier, PSA strongly outperforms SDM. The reason is that SDM gives more attention to the samples close to mean and thus it cannot capture the disparity and variance of a dataset properly. In contrast, the PSA algorithm considers both similarities and variances at the same time. Compared to SRS, PSA outperforms it in cases SVM (90%), LDA and RF (90% and 60%), and LR (60%). This shows that PSA works better than merely random sampling from a dataset because it tries to value the more important samples in discrimination while SRS looks at all samples with the same eye. Note that PSA is performed once while SRS is performed 20 times so that in some cases SRS outperforms PSA as some bad results are absorbed in averaging.

2) Experiments on the AT&T dataset: For the AT&T dataset, the same cross validation setup was used with PCA used as a preprocessing step due to the high dimensionality of images. The accuracy rates of experiments on this dataset are reported in Table III. PSA outperforms using the entire data for RF classifier (portion 90%). Apart from this case, PSA does not outperform the entire data because in this dataset, the number of samples of every class is very small compared to the number of classes and sampling is mostly expected to reduce the accuracy slightly. In all portions and classifiers, PSA strongly outperforms SDM due to the use of the disparity and variance of data in addition to similarities of samples in every class. Moreover, one strength of PSA over SRS is valuing the samples based on their representation and discrimination. This explains why in most of the cases, which are SVM and LDA (50%), RF and NB (50% and 90%), and LR (90%), PSA strongly outperforms SRS.

3) Experiments on the MNIST dataset: The MNIST dataset has its own standard training/testing sets which were used in this work. Similar to the AT&T dataset, PCA is applied on the image data as a preprocessing step. The accuracy rates of experiments on this dataset are reported in Table IV. As shown in this table, PSA with portion 90% outperforms using the entire data for LDA, RF, LR, and NB classifiers. It is interesting that PSA even outperforms using the entire dataset in portions 6% (using LDA and RF) and 50% (using RF, LR, and NB). In the cases LDA (6%), QDA and NB (6% and 50%), and RF (6%, 50%, and 90%), PSA strongly outperforms SRS. Moreover, in all portions using all classifiers except SVM, PSA strongly outperforms SDM in this dataset. To better compare PSA, SRS, and SDM, see Fig. 3 which

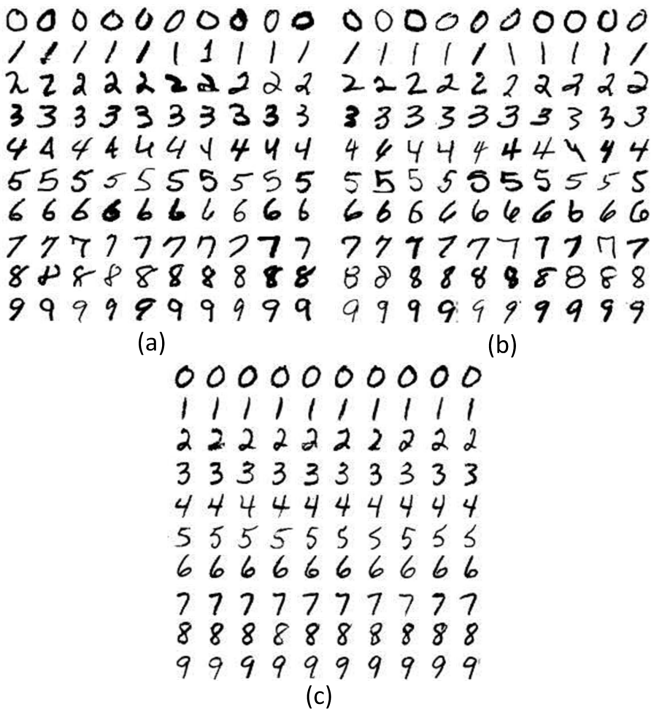


Fig. 3: Top ranked samples of MNIST dataset using (a) PSA, (b) SRS, and (c) SDM methods.

depicts the top ranked samples of every class in MNIST dataset using these three methods. For SRS, the random samples of one of the runs are shown. Comparing PSA and SDM, PSA has captured more diverse samples because of taking both similarities and variances into account, while SDM gives top ranks to the samples close to mean and thus the top ranks are very similar. Comparing PSA and SRS, more diverse but 'better' representative samples of digits are selected by PSA. For PSA, this is specifically seen for digits 1 (different slopes), 2 and 3 (styles), 4 (two different types of writing), 5, 6, 7, and 8 (styles), and 9 (slopes).

IV. Conclusion and Future Work

A new data reduction method, Principal Sample Analysis, was proposed which was shown to outperform other data reduction approaches on a diverse range of data problems. The algorithm is useful for ranking samples of every class based on the discriminative information of data samples. Moreover, it was shown that PSA can outperform other sampling methods such as SRS and SDM in many cases, and this shows the effectiveness of the proposed data reduction method. Interestingly, PSA outperforms even the use of the entire data in many cases which shows that how a reduced dataset can be useful for improved accuracy as well as memory savings. This paper has introduced the concept of PSA but has not focused on computational efficiency of the algorithm. The current runtime is dominated by computation of the regression score where random sampling is done several times while a loop is performed in every iteration of random sampling. For now, high-dimensional data such as images need to be reduced in dimension before feeding to PSA, although the ranked samples would retain the original dimensions finally.

References

[1] M. Welling, "Fisher linear discriminant analysis," Department of Computer Science, University of Toronto, Tech. Rep., 2005.

[2] M. Sugiyama, "Local fisher discriminant analysis for supervised dimensionality reduction," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 905–912.

[3] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley interdisciplinary reviews: computational statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[4] I. K. Fodor, "A survey of dimension reduction techniques," Lawrence Livermore National Lab., CA (US), Tech. Rep., 2002.

[5] B. George, "A study of the effect of random projection and other dimensionality reduction techniques on different classification methods," *Baselius Researcher*, p. 201769, 2017.

[6] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[7] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[8] Y. Tillé, *Sampling algorithms*. Springer, 2006.

[9] D. Barbar'a, W. DuMouchel, C. Faloutsos, P. J. Haas, J. M. Hellerstein, Y. Ioannidis, H. Jagadish, T. Johnson, R. Ng, V. Poosala *et al.*, "The new jersey data reduction report," in *IEEE Data Engineering Bulletin*. Citeseer, 1997.

[10] M. Shahrokh Esfahani and E. R. Dougherty, "Effect of separate sampling on classification accuracy," *Bioinformatics*, vol. 30, no. 2, pp. 242–250, 2013.

[11] T. R. Lunsford and B. R. Lunsford, "The research sample, part i: sampling," *JPO: Journal of Prosthetics and Orthotics*, vol. 7, no. 3, p. 17A, 1995.

[12] P. M. Lance and A. Hattori, *Sampling and evaluation: A guide to sampling for program impact evaluation*. Chapel Hill, North Carolina: MEASURE Evaluation, University of North Carolina, 2016.

[13] D. L. Donoho, "High-dimensional data analysis: The curses and blessings of dimensionality," in *AMS Conference on Math Challenges of the 21st Century*, 2000.

[14] D. A. Van Dyk and X.-L. Meng, "The art of data augmentation," *Journal of Computational and Graphical Statistics*, vol. 10, no. 1, pp. 1–50, 2001.

[15] J. S. Liu and Y. N. Wu, "Parameter expansion for data augmentation," *Journal of the American Statistical Association*, vol. 94, no. 448, pp. 1264–1274, 1999.

[16] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," in *Readings in computer vision*. Elsevier, 1987, pp. 726–740.

[17] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*. Springer series in statistics New York, 2001, vol. 1.

[18] R. Bro and A. K. Smilde, "Principal component analysis," *Analytical Methods*, vol. 6, no. 9, pp. 2812–2831, 2014.

[19] O. L. Mangasarian, W. N. Street, and W. H. Wolberg, "Breast cancer diagnosis and prognosis via linear programming," *Operations Research*, vol. 43, no. 4, pp. 570–577, 1995.

[20] "Machine learning repository, wisconsin diagnostic breast cancer dataset," https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic), accessed: 2018-01-01.

[21] F. S. Samaria and A. C. Harter, "Parameterisation of a stochastic model for human face identification," in *Applications of Computer Vision, 1994., Proceedings of the Second IEEE Workshop on*. IEEE, 1994, pp. 138–142.

[22] "At&t laboratories cambridge, at&t face dataset," http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html, accessed: 2018-01-01.

[23] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[24] Y. LeCun, C. Cortes, and C. J. Burges, "Mnist handwritten digits dataset," http://yann.lecun.com/exdb/mnist/, accessed: 2018-01-01.