Data geometric supervised learning: Supplementary material

Anonymous Author 1
Unknown Institution 1

Anonymous Author 2
Unknown Institution 2

Anonymous Author 3
Unknown Institution 3

1 Additional details on mathematical properties

We use the notations $(\Omega, \mathcal{A}, \mathbb{P})$ for the generic σ-algebra pertaining to which our random elements are defined, and ω is the notation for a generic point in the sample space. Expectation extended by the σ-finite measure $\mathbb{P}$ is denoted by $\mathbb{E}$. We consider a generic absolutely continuous random vector $X \in \mathcal{X} \subseteq \mathbb{R}^p$, with median at the origin $\mathbf{0}$ without loss of generality. Random vectors with median not equal to the origin can be location shifted without any significant modification to the results presented below. Random vectors that are not absolutely continuous require some special technical assumptions and additional algebra, which are necessary for mathematical rigor, but are not conceptually challenging though tedious in nature. Below, the norms of all vectors considered is the Euclidean norm, though our results easily extend to many other norms in $\mathbb{R}^p$, for example, any $\mathcal{L}_d$ norm with $d \geq 1$. The inner product is also the standard Euclidean inner product, and again standard variations like weighted inner products may be used with little or no additional effort. All notations are as in the main paper.

Define $\tilde{X}_{\mathbf{u}i} = w_{\mathbf{u}} w_{2j_k} X_{\mathbf{u}i}$, for $i = 1, \ldots, n$. Note that the DCW, in any direction $\mathbf{e}_{\mathbf{u}}$, is a minimizer of the expectation of the following function

$$\Psi_u(q) \quad = \mathbb{I}_{\{||X_{\mathbf{u}\perp i}|| \leq \epsilon\}} \left[|\tilde{X}_{\mathbf{u}i} - q| + ||\mathbf{u}||(\tilde{X}_{\mathbf{u}i} - q) \right].$$

Our results utilize two interesting properties of X and the function $\Psi_u(\cdot)$. First, we establish that $\Psi_u(q)$ is convex in q, and obtain a measurable subgradient function. Second, under fairly standard assumptions on the *population* properties (but not necessarily on the sample properties) of the subgradient at the appropriate quantile value, we establish several mathematical and statistical results. An extremely easy example where population and sample values differ may be seen in the context of a Binomial (n, θ) random variable Z. Note that the expectation of Z/n is θ, which is a smooth function on $(0, 1)$. However, the sample expectation, ie, the same quantile computed under the empirical distribution function, is just Z/n, which is supported only on discretely many values, and is not a smooth function.

Our first result is to establish the convexity of $\Psi_u(\cdot)$.

Lemma 1.1 *The function*

$$\Psi_{\mathbf{u}}(q) \quad = \mathbb{I}_{\{||X_{\mathbf{u}\perp i}|| \leq \epsilon\}} \left[|\tilde{X}_{\mathbf{u}i} - q| - ||\mathbf{u}||(\tilde{X}_{\mathbf{u}i} - q) \right].$$

is convex in q, with a measurable subgradient function given by

$$g(X, q) \quad = \mathbb{I}_{\{||X_{\mathbf{u}\perp i}|| \leq \epsilon\}} \left[\left(2\mathbb{I}_{\{\leq \tilde{X}_{\mathbf{u}i} \leq q\}} - 1 \right) - ||\mathbf{u}|| \right].$$

The proof of this result is quite simple and hence omitted. We assume that $\mathbb{E}\Psi_{\mathbf{u}}(q)$ is finite for all potential choices of q, and has a unique minimizer, which we call $q^*_{\mathbf{u}}$. This only requires that the *population version* $\mathbb{E}\Psi_{\mathbf{u}}(q)$ is strictly convex in a neighborhood of its minimizer, which is not a strong assumption. The sample version does not require uniqueness, but that may be enforced, as is traditionally done, by defining the minimizer to be the infimum over all possible values at which the minimum is reached.

Theorem 1.1 *The sample DCW is a consistent estimator of the population DCW, that is $q^*_{\mathbf{u}} \to q^*_{\mathbf{u}}$ almost surely as sample size $n \to \infty$.*

Theorem 1.2 *Under the additional* **population level** *conditions that $\mathbb{E}g^2(X, q^*_{\mathbf{u}}) < \infty$, and that the function $\mathbb{E}\Psi_{\mathbf{u}}(X, q)$ is twice continuously differentiable at $q^*_{\mathbf{u}}$ with the second derivative H being positive definite, then as $n \to \infty$*

$$n^{1/2}(q_{\mathbf{u}} - q^*_{\mathbf{u}}) = -n^{-1/2} H^{-1} S_n + o_P(1),$$

where $S_n = \sum_{i=1}^n g(X_i, q^*_{\mathbf{u}})$. This implies, in particular, that $n^{1/2}(q^*_{\mathbf{u}} - q^*_{\mathbf{u}})$ is asymptotically Normal,

with asymptotic variance $H^{-1}VH^{-1}$ *where* $V = Var\ g(X, q^*_{\mathbf{u}})$.

The proofs of both Theorem 1.1 and Theorem 1.2 use convexity, and require considerable algebra, hence only an outline of the kind of argument we use is outlined below.

Following [2] and [3], we have the following result:

Lemma 1.2 *Let* $G_n(x,\omega)$ $n = 1, 2, \ldots,$ *be random functions defined on a fixed convex set* $\mathcal{X} \subseteq \mathbb{R}^p$*, that are all convex in* x*, for almost all* ω*. Let* $G(x,\omega)$ *be a random function such that for each fixed* $x \in \mathcal{X}$*,* $G_n(x,\omega) \to G(x,\omega)$ *almost surely. Then for each fixed* $M > 0$*, we have*

$$\sup_{||x|| \leq M} |G_n(x,\omega) - G(x,\omega)| \to 0,$$

almost surely.

The proof of this lemma rests on the fact that pointwise convergence of convex functions on a dense subset implies uniform convergence over compact sets, [4]. As in [3] and [1], we use a countable dense set to establish this result.

Once we have this lemma, the proof of Theorem 1.1 follows from a careful algebraic argument. Then, using Theorem 1.1, we construct the proof of Theorem 1.2 by developing some additional properties relating to the random Bregman divergences constructed using $\Psi_{\mathbf{u}}(X, q)$.

The proof of the theorem on data depth in the original paper is algebraic in nature, and is omitted here.

2 Additional details on the simulation experiment

We created two simulation datasets with two variable $X1$ and $X2$ in each dataset:

1. **Distinct Class Separation:** Figure 1 shows the dataset with distinct class separation. The estimated variance covariance matrix of the two variables in the dataset is shown in table 2: The red points in the elliptical space at the top corresponds to class label 1 and the heart shaped data cluster at the bottom corresponds to data label 0. A total of 15000 data points were generated with 5000 datapoints in class 1 and 10000 datapoints in class 0.

2. **Less Distinct Class Separation:** Figure 4 shows the dataset with less distinct class separation and higher overlap between the two classes. The sample variance covariance matrix of the two variables in the dataset is shown in table 2:

   Also, like dataset 1, the red points in the elliptical space at the top corresponds to class label 1 and the heart shaped data cluster at the bottom corresponds to data label 0. A total of 15000 data points were generated with 5000 datapoints in class 1 and 10000 datapoints in class 0.

We split the data into 80 percent train set and 20 percent test set. We train the classification models with the train dataset in both cases and assess the accuracy of classification on the test 20 percent test dataset. We repeat the process 1000 times to ensure sufficient randomization. Apart from our proposed weighted projection quantile based geometric learning algorithm, we also tried out several other standard classification algorithms in the datasets. Described below are the other standard methods for comparison that we tried out on the two datasets.

- **Logistic Regression:** We used the standard GLM command in R to fit a logistic regression with the train subset. We predicted the class of the test dataset using the fitted model using *predict.glm* method in R.

- **Linear Discriminant Analysis:** We used the *lda* routine in the MASS library of R to fit LDA on the train dataset. We used a prior probability proportional to class representation in the dataset. Since, the sample splitting into train and test were done using stratified sampling for the two classes separately, the proportional representation of the two classes in the original simulated datasets and the train or test subsamples are very similar. The method was run without cross-validation.

- **Quadratic Discriminant Analysis:** We used the *qda* routine in the MASS library of R to fit QDA on the train dataset. We used a prior probability proportional to class representation in the dataset. Since, the sample splitting into train and test were done using stratified sampling for the two classes separately, the proportional representation of the two classes in the original simulated datasets and the train or test subsamples are very similar. The method was run by setting the cross validation parameter to be FALSE.

- **Random Forest:** The R package *randomForest* was used to run random forest classification

algorithm on the datasets. The number of trees for each run of random forest was kept at 500. Sampling for tree building was done with replacement. Rest of the parameters were run with default setting.

- **Neural Network:** The R package *nnet* was used for running neural network on the data. The size of the hidden layer was set at 2, case wise sample weights were set at 1, and entropy fit was used instead of maximum conditional likelihood. Rest of the parameters were set at default values of the package.

- **Support Vector Machine:** The *svm* routine in the R package *e1071* was used to run a SVM fit on the data. *Radial basis kernel* ($= exp(-\gamma * |u - v|^2)$) was used for the SVM fit of type *C-Classification* using a scaling of 1 and a class weight of proportion of observation in each class in the train set.

- **K-nearest neighbor:** The routine *knn* in the R package *class* was used for fitting KNN classification algorithm on the dataset. The parameter k was set at 100 and the minimum voting parameter for definite decision was set at zero to avoid conflicts and NAs in the predicted classification.

The table 2 indicates the accuracy of classification and running time of each of the algorithms on the dataset 1 with distinct class separation. The table 2 indicates the accuracy of classification and running time of each of the algorithms on the dataset 2 with less distinct class separation.

The following simulation graphics shows the graphical output for the two class seperation boundaries generated by various methods in comparison to the proposed method. For each dataset thee graphical plots are included, i.e., the class boundaries with the scatter plot of the data, the class boundaries in comparison to the corresponding iso-depth lines corresponding to the 90^{th} quantiles and the class seperation boundaries achieved by the standard methods in comparison to the proposed method.

Table 1: **Variance-Covariance of dataset 1**

	X1	**X2**
X1	0.1341062120	0.0002883527
X2	0.0002883527	0.0359856218

Table 2: **Variance-Covariance of dataset 2**

	X1	**X2**
X1	0.0938121199	0.0001310929
X2	0.0001310929	0.0526307035

Table 3: **Comparison of the Proposed Classification Algorithm with Standard Classification Algorithms for Dataset 1**

Sl.	Method	Classification Accuracy	Running Time
1	Geometric Learning	0.996	1.46
2	Logistic Linear Model	0.981	0.28
3	LDA	0.969	0.22
4	QDA	0.992	0.27
5	Random Forest (500 trees)	0.998	23.61
6	Neural Network	0.967	4.76
7	SVM	0.984	6.45
8	KNN	0.997	1.53

Table 4: **Comparison of the Proposed Classification Algorithm with Standard Classification Algorithms for Dataset 2**

Sl.	Method	Classification Accuracy	Running Time
1	Geometric Learning	0.976	1.48
2	Logistic Linear Model	0.901	0.31
3	LDA	0.881	0.20
4	QDA	0.974	0.21
5	Random Forest (500 trees)	0.976	31.77
6	Neural Network	0.962	3.56
7	SVM	0.974	8.54
8	KNN	0.989	1.62

References

[1] A. Bose. Bahadur representation of m_m estimates. *Ann. of Statist.*, 26-2:771–777, 1998.

[2] S.J. Haberman. Concavity and estimation. *Ann. of Statist.*, 17:1631–1661, 1989.

[3] W. Niemiro. Asymptotics for m-estimators defined by convex minimization. *Ann. of Statist.*, 20:1514–1533, 1992.

[4] R.T. Rockafeller. *Convex Analysis.* Princeton Univ. Press, Princeton, NJ, 1970.

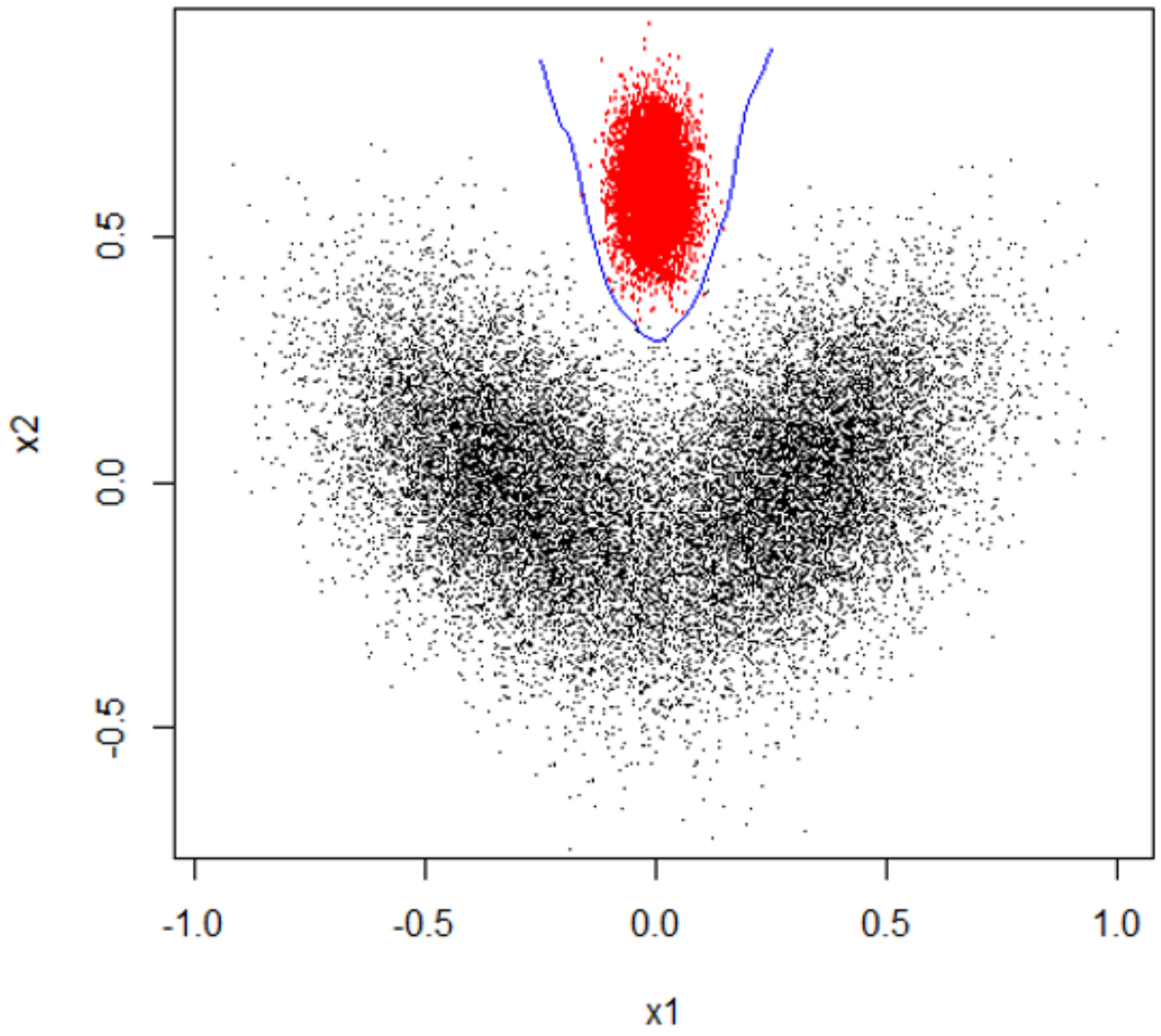

Figure 1: Scatterplot of Simulated Dataset 1 with Class Separator by the Proposed Geometric Classification Method

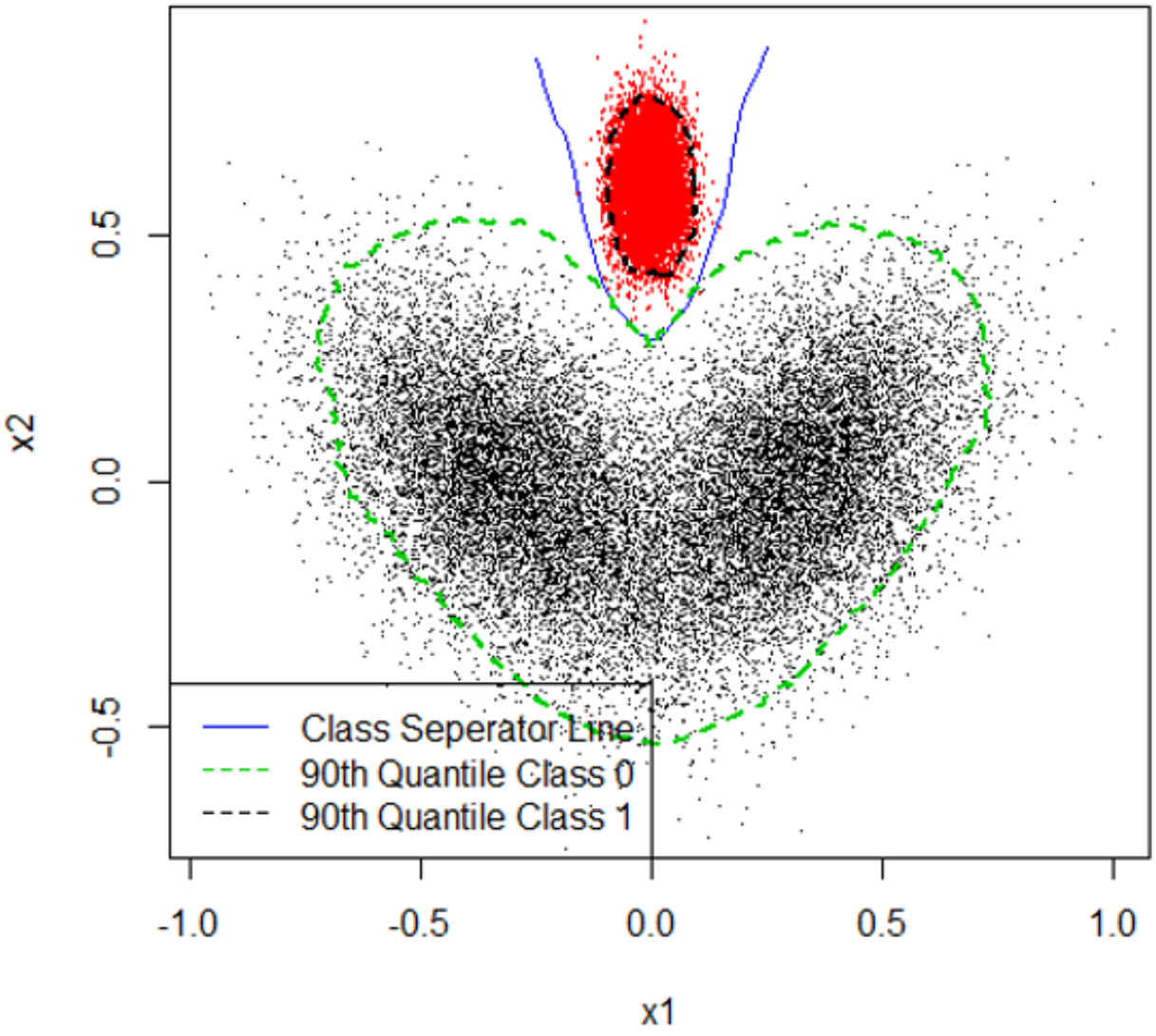

Figure 2: Scatterplot of Simulated Dataset 1 with Class Separator by the Proposed Geometric Classification Method and the Iso-depth Contours

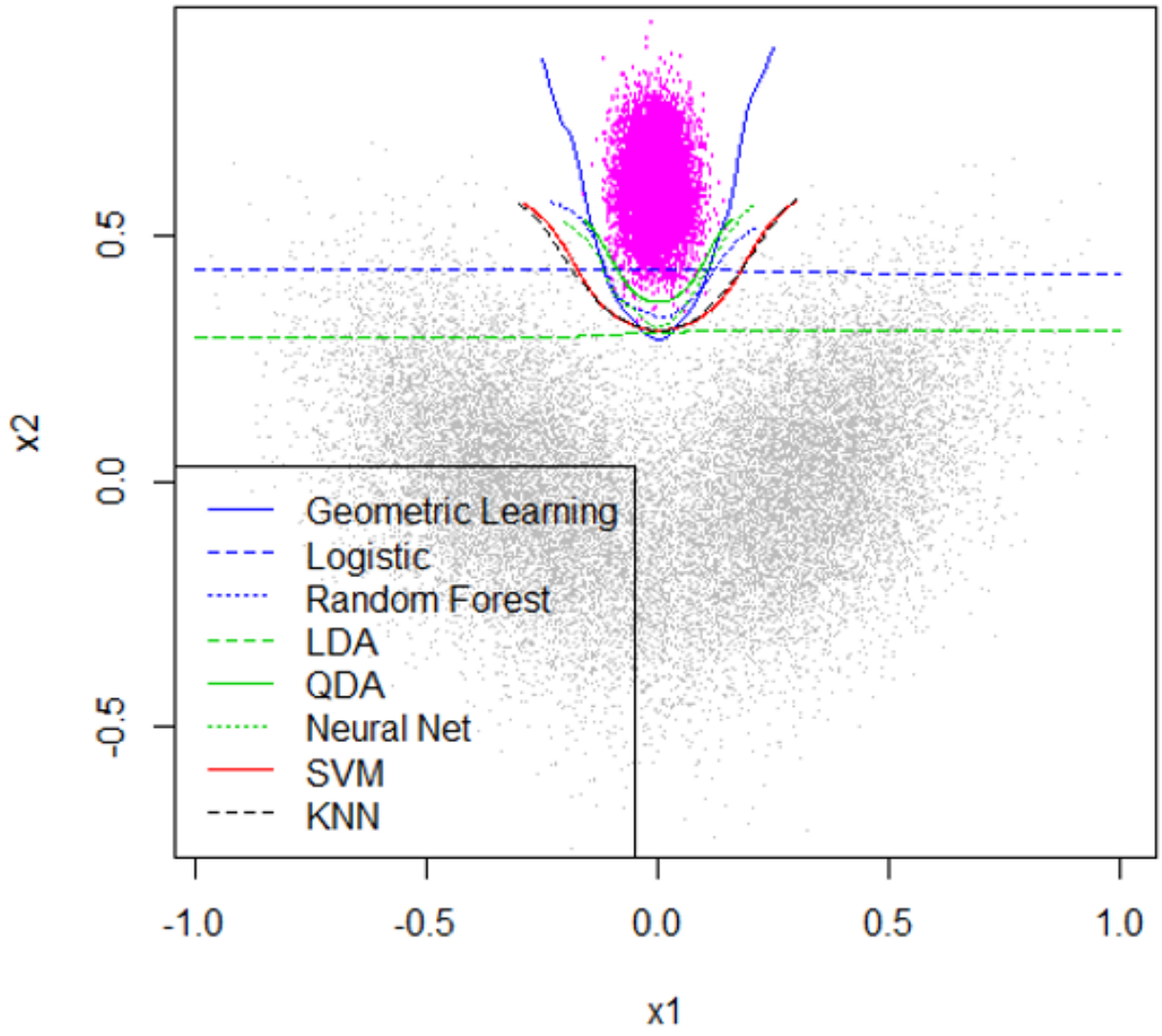

Figure 3: Scatterplot of Simulated Dataset 1 with Class Separator by the Proposed Geometric Classification Method and Other Standard Methods

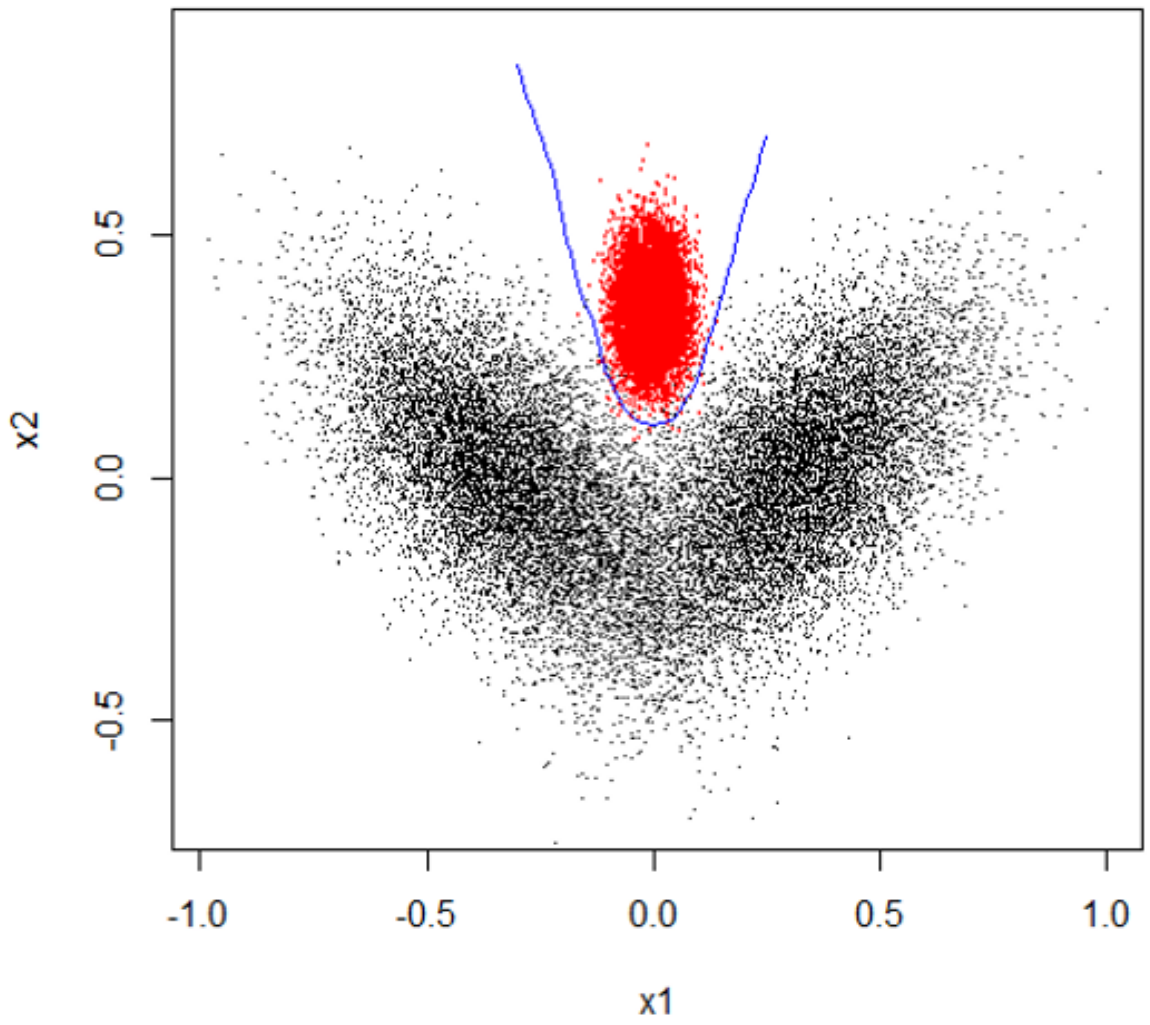

Figure 4: Scatterplot of Simulated Dataset 2 with Class Separator by the Proposed Geometric Classification Method

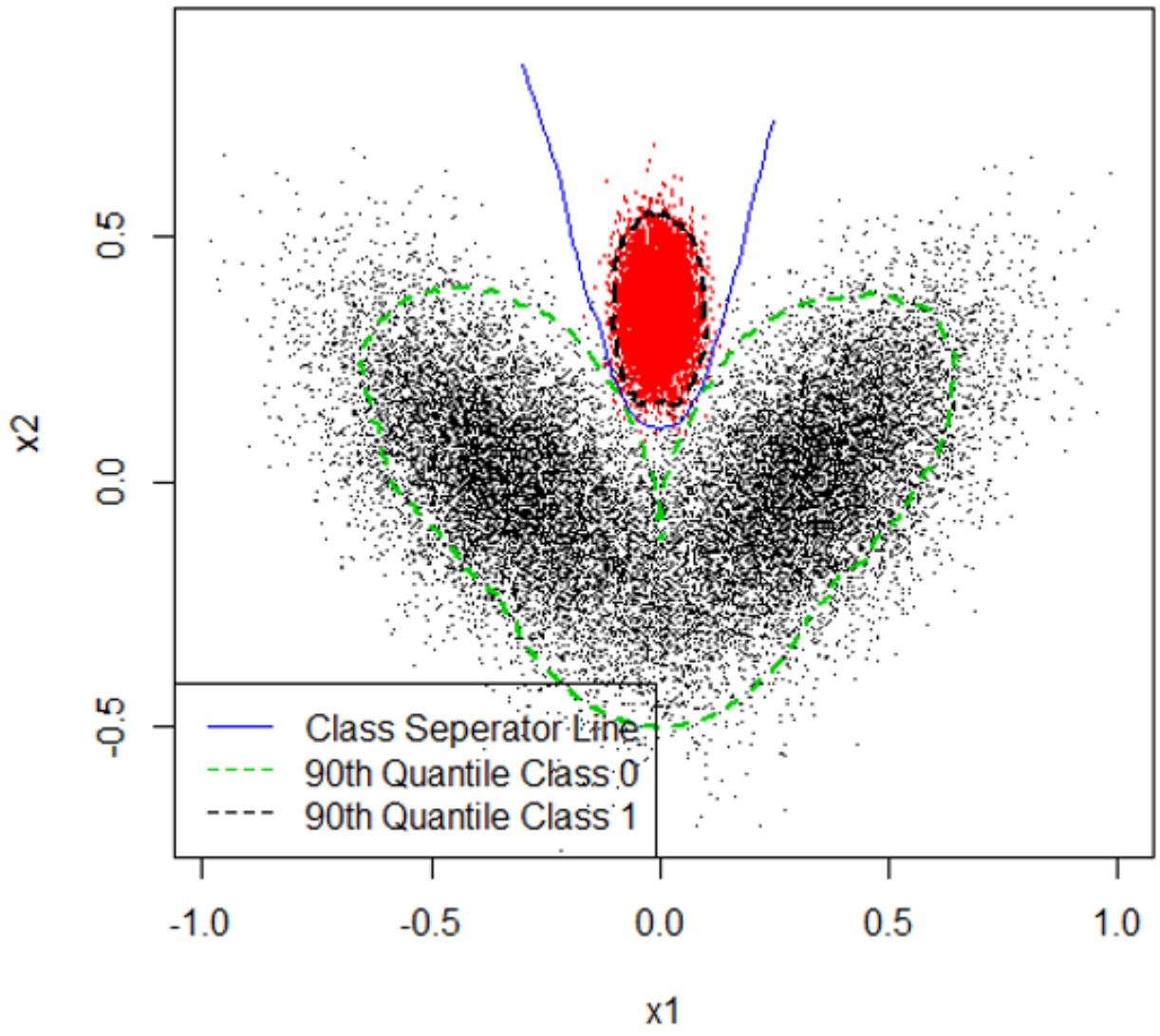

Figure 5: Scatterplot of Simulated Dataset 2 with Class Separator by the Proposed Geometric Classification Method and the Iso-depth Contours

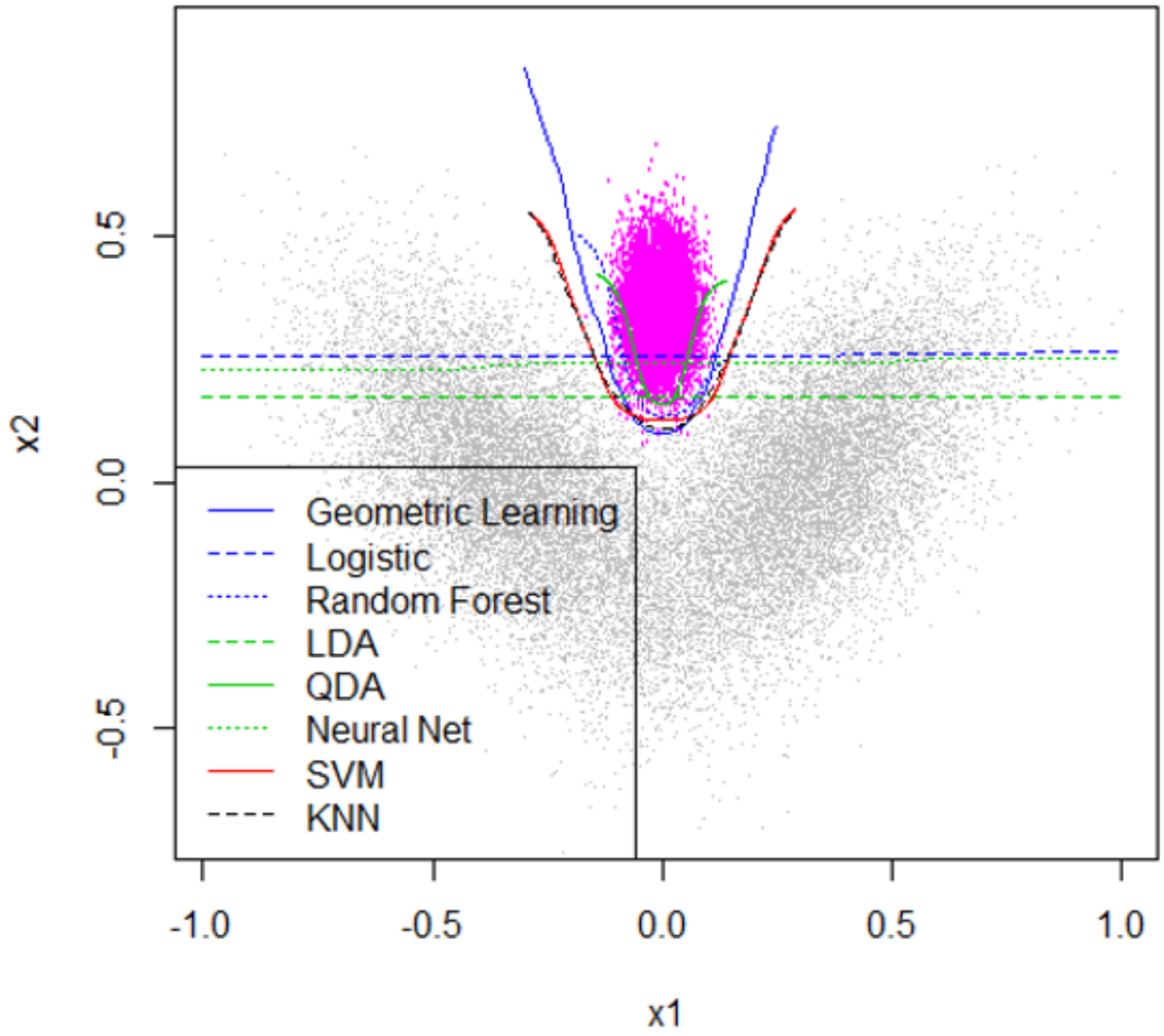

Figure 6: Scatterplot of Simulated Dataset 2 with Class Separator by the Proposed Geometric Classification Method and Other Standard Methods