Liav Hen, 315708776

MATHEMATIC METHODS - FINAL PROJECT

ABSTRACT

I chose to review the paper "Few-Sample Feature Selection via Feature Manifold Learning", by Cohen et al. The authors suggest a supervised algorithm to perform feature selection, termed "Manifest". By using the class labels, Manifest finds and scores the most discriminative features of the data, with respect to the different classes. In my project, I suggest examining the quality of the feature selection by constructing a weak-classifier based on each feature, and plugging it to AdaBoost, a learning algorithm that integrates multiple weak classifiers into a stronger classifier with boosted performance. This way, the classification error will be used to evaluate the algorithm.

In my work, I conduct a comparison between Manifest and another well-known supervised feature-selection algorithm ReliefF, which is considered state-of-the-art in the task of supervised feature selection. The experiments show that Manifest is indeed competitive with ReliefF, even with small number of training samples.

BACKGROUND

MANIFEST – MANIFOLD-BASED FEATURE SELECTION

The ManiFeSt algorithm for feature selection works by constructing a feature space representation for each class using a kernel. It then computes the differences between the feature spaces in a Riemannian manner and uses spectral analysis to calculate a feature score that indicates the discriminative power of each feature. The algorithm selects the features with the highest scores as the most influential features.

Given two classes, the algorithm works as follows:

1. The labeled data X is divided into classes, constructing $X^{(1)} \in \mathbb{R}^{N_1 \times d}$, $X^{(2)} \in \mathbb{R}^{N_2 \times d}$, where $N_1 + N_2 = N$ relate to the number of samples in each class (and together).
2. Using a $d \times d$ RBF kernel, a Symmetric Positive Semi-Definite (SPSD) matrices are constructed for each class ℓ:
$$K_\ell[i,j] = \exp\left\{-\frac{\left\|x_i^{(\ell)} - x_j^{(\ell)}\right\|^2}{2\sigma_\ell^2}\right\}, i,j = 1,\dots,d$$
3. The mean M of K_1, K_2 is computed as the mid-point of the geodesic path between K_1, K_2 on the SPD manifold:
$$M = \gamma_{K1\to K2}(1/2) = K_1^{\frac{1}{2}}\left(K_1^{-\frac{1}{2}} K_2 K_1^{-\frac{1}{2}}\right)^{\frac{1}{2}} K_1^{\frac{1}{2}}$$
4. The difference D is computed as the distance between K_1 or K_2 to the plane that is tangent to the SPD manifold and includes M by the following formula:
$$D = \log_M K_1 = -\log_M K_2 = M^{\frac{1}{2}} \log\left(M^{-\frac{1}{2}} K_1 M^{-\frac{1}{2}}\right) M^{\frac{1}{2}}.$$
5. Using spectral analysis, the eigenpairs $\lambda_i^{(D)}, \phi_i^{(D)} \in \mathbb{R}^d$ of D are computed for $i = 1{:}\,d$.
6. The proposed score $\boldsymbol{r} \in \mathbb{R}^d$ is (where $\odot$ denotes element-wise multiplication):
$$\boldsymbol{r} = \sum_{i=1}^{d} |\lambda_i^{(D)}| \cdot (\phi_i^{(D)} \odot \phi_i^{(D)})$$

ADABOOST

AdaBoost, short for Adaptive Boosting, is an ensemble learning technique that builds a strong classifier by combining multiple weak learners. It begins by assigning equal weights to all training examples and then trains a series of weak classifiers on the data. Each weak classifier is given a weight based on its performance, and the algorithm focuses more on misclassified examples in subsequent iterations by adjusting their weights. This iterative process continues until a predetermined number of classifiers is reached or until perfect predictions are achieved. Finally, the predictions of all weak classifiers are combined using a weighted sum to make the final classification.

EXPERIMENTS SETUP

The experiments entailed the classification of hand-written digits from the widely recognized MNIST dataset within a particular binary context, as also demonstrated by the paper's author, which is distinguishing between the digits 4 and 9. In this framework, each pixel in the image is denoted as a feature. Consequently, the feature selection algorithms (Manifest and ReliefF) primarily focus on identifying the most distinguishing pixels among the hand-written samples of 4s and 9s.

Let us visualize the score that each algorithm assigns each pixel after observing 40 and 1000 samples:

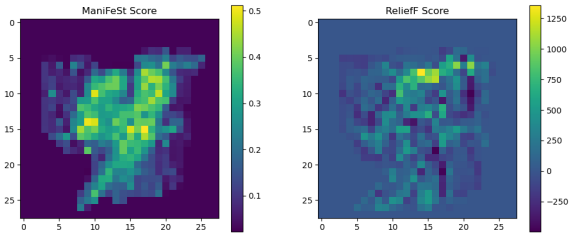

FIGURE 1: FS SCORES AFTER OBSERVING 40 SAMPLES (RIGHT: MANIFEST, LEFT: RELIEF)

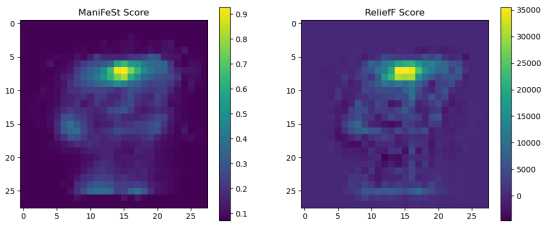

FIGURE 2: FS SCORES AFTER OBSERVING 1000 SAMPLES (RIGHT: MANIFEST, LEFT: RELIEF)

Now, an AdaBoost classifier was trained for $T = 30$ time-steps, based on the following $2M$ weak-classifiers; For each feature f_j suggested by the FS algorithm ($j = 1{:}M$, where M is the number of selected features), two decision rules were built[1]:

(1) Classify the sample as "positive" if $f_j \geq 128$, else "negative".
(2) Classify the sample as "positive" if $f_j < 128$, else "negative".

Let us denote AdaBoost(A) as an AdaBoost classifier that was trained in the described manner based on feature selection perform by some algorithm A.

This test was conducted for various amounts of selected features, as well as various amounts of training data for the feature selection algorithms. For each number of features and sample size, 10 identical experiments were conducted and averaged. As mentioned, besides AdaBoost(Manifest) and AdaBoost(ReliefF), I tested also AdaBoost(Random) which is based on random selection of features (hence is unsupervised).

[1] The inspiration to integrate AdaBoost in this manner stemmed from an assignment provided in the "Computational Learning Theory" course by Dr. Roi Livni. The assignment involved implementing AdaBoost using randomly selected pixels as the basis in the manner described above, so the framework is based on that.

RESULTS

First, the following heatmaps describe the mean error rate of AdaBoost(Manifest), AdaBoost(ReliefF), AdaBoost(Random), as function of the number of selected features and the number of observed samples. In this visualization we are interested to zoom-in on the few-samples behavior of the algorithms.

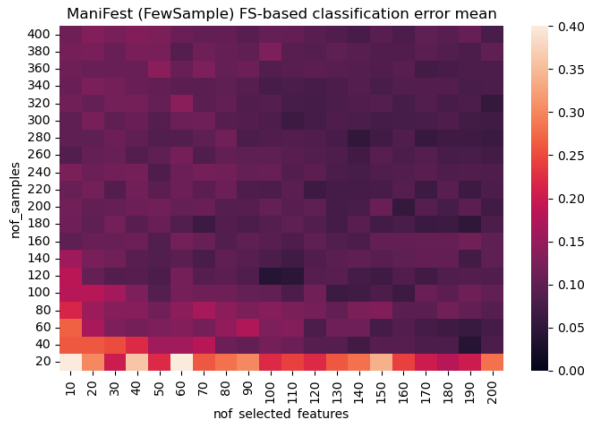

FIGURE 3: MEAN ERROR OF ADABOOST(MANIFEST)

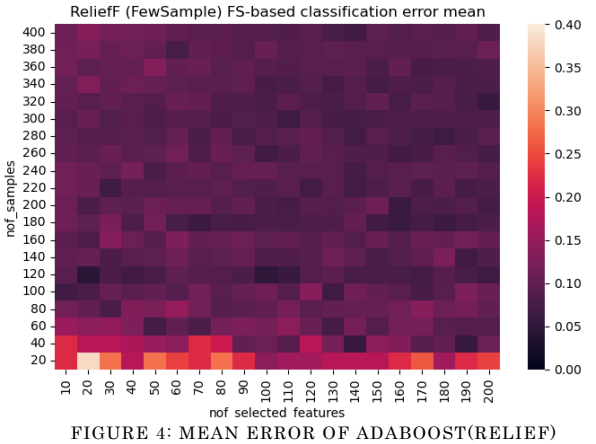

FIGURE 4: MEAN ERROR OF ADABOOST(RELIEF)

To further show the dependency of each algorithm in the number of selected features or number of training samples, I used violin plots:

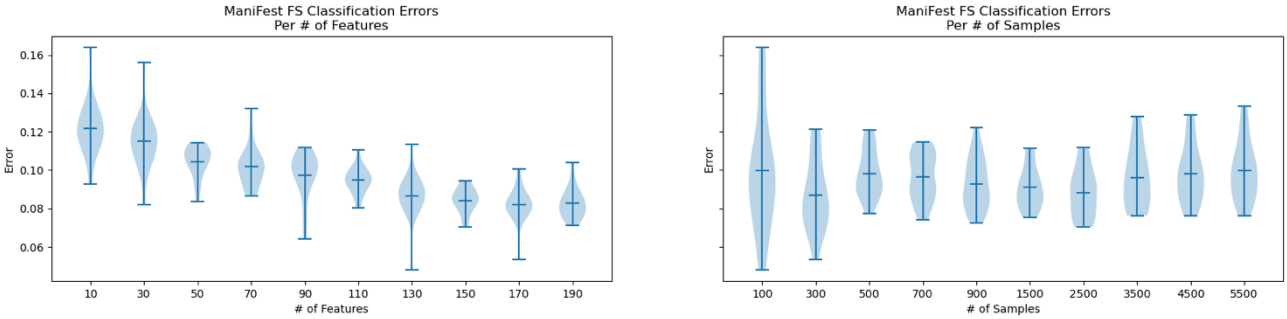

FIGURE 5: ADABOOST(MANIFEST) DISTIRBUTION AS FUNCTION OF NUMBER OF (LEFT) FEATURES AND (RIGHT) SAMPLES

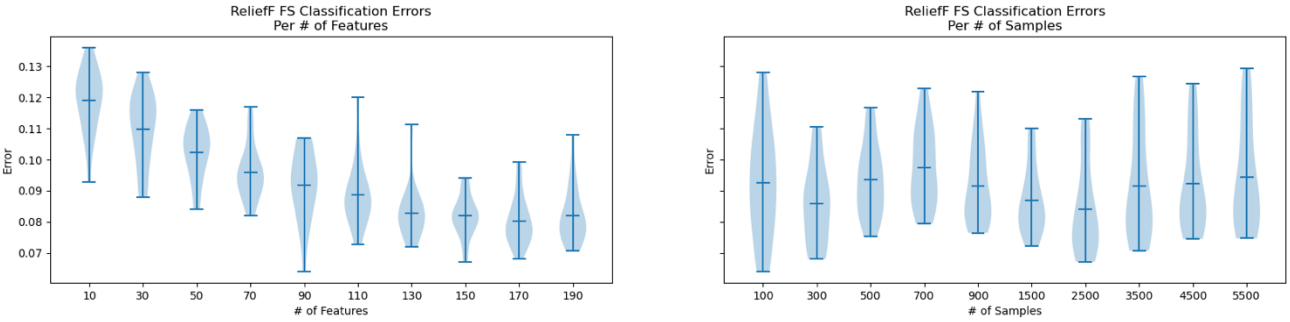

FIGURE 6: ADABOOST(RELIEF) DISTIRBUTION AS FUNCTION OF NUMBER OF (LEFT) FEATURES AND (RIGHT) SAMPLES

CONCLUSIONS

First, let us provide plots similar to those above, but for random choice of feature to conduct the classification with:

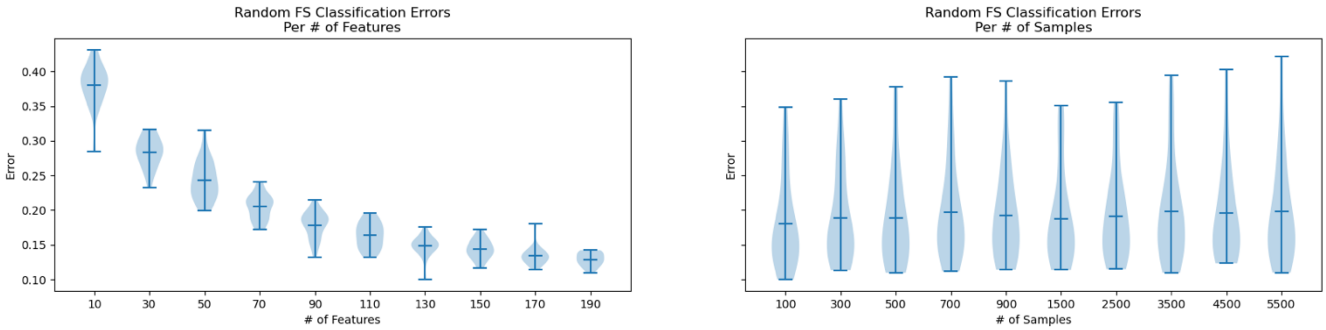

FIGURE 7: ADABOOST(RANDOM) DISTIRBUTION AS FUNCTION OF NUMBER OF (LEFT) FEATURES AND (RIGHT) SAMPLES

1. There is no doubt that employing algorithmic feature selection methods significantly enhances performance compared to random selection.
2. In this task, wrapped by AdaBoost, **ReliefF achieves slightly better mean performance** overall.
3. The decline in error for AdaBoost(Manifest) is less rapid with an increase in the number of features compared to AdaBoost(ReliefF). This suggests that **Manifest identifies fewer features as discriminative.** This deduction is supported by Figure 2, which illustrates that fewer features receive high scores in Manifest compared to ReliefF.
4. **ReliefF appears to exhibit greater stability, as evidenced by its smaller variance in classification error compared to Manifest.** This variance arises from the random selection of training samples in each experiment, as well as variations in their sizes across experiments. This deduction is supported by the narrower distributions of each "violin" plot in Figure 5 compared to those in Figure 6 (when taking into account the scale of the vertical axis).

It's important to clarify that the authors of the paper presented three demonstrations on synthetic yet sophisticated toy problems, conducted one classification task on synthetic data (Gisette), and performed two classification tests on real albeit small datasets (Colon & Prostate). Furthermore, they utilized the SVM classifier, known for its robustness compared to weaker classifiers like a simple threshold decision rule based on a single feature or an ensemble of such rules in AdaBoost. Despite the challenges posed by real-world datasets such as MNIST and the reliance of the classifier on features rather than a complex function of them, Manifest demonstrates competitive performance with the state-of-the-art ReliefF, as outlined in the paper.

PROPOSITION FOR FUTURE WORK

Throughout the paper, the authors chose M as the mid-point on the geodesic path between the matrices K_1, K_2, but the any point on the geodesic path can be generally expressed with the parameterization of $t \in [0,1]$ as:

$$M(t) = \gamma_{K1 \to K2}(t) = K_1^{\frac{1}{2}} \left(K_1^{-\frac{1}{2}} K_2 K_1^{-\frac{1}{2}} \right)^t K_1^{\frac{1}{2}}$$

Further research may examine the relation between t, the path parameter, to the ratio $R_1 \triangleq \frac{N1}{N_1+N_2}$, which is the fraction of the data that belongs to the first class. It may be that in the general case of imbalanced data, which is not discussed in the paper (although N_1, N_2 are not constrainted to be equal), a different choice of for t may induce enhanced performance.

This assumption requires theoretical foundation, similar to the results presented in Section 4 of the paper, and should also be supported by numerical experiments showing the dependency of the classification error given t, R_1.