Title

xxx[1], yyy[1], and zzz[1]

[1]Faculty of Mathematics and Information Science, Warsaw University of Technology
ul. Koszykowa 75, 00-662 Warsaw, Poland
{xxx,yyy,zzz}@mini.pw.edu.pl

Abstract. In the article we present

1 Introduction

2 Literature Review

3 Preliminaries

3.1 Ellipsoids for Foreign Elements Rejection

In computational geometry, the smallest enclosing box problem is that of finding the oriented minimum bounding box enclosing a set of points. In many cases there's a need for computing convex hull of set of points and testing inclusions of other points in the same space. This can become very complex problem especially in higher dimensions, so more often an approximation of the hull is used. This helps to reduce time needed for computations, since most of the methods have lower construction and inclusion-testing complexities. Some of such approaches include using figures like hypercubes, diamonds, balls or ellipsoids to successfully enclose given set of points.

When comparing highlights and drawbacks of each method, from computational complexity, ease of testing point inclusion and algorithm implementation point of view, ellipsoids seem very reasonable to choose. Constructed ellipsoid is superior to the minimal cuboid in many ways. It is unique, better approximation of the object it contains and if $E(S)$ is the bounding ellipsoid for a point set S with convex hull $C(S)$ in dimension d, then:

$$\frac{1}{d}E(S) \subseteq C(S) \subseteq E(S)$$

where scaling is with respect to the center of $E(S)$.

MVEE problem is solved by several known algorithms that can be categorized as first-order, second-order interior-point or combination of the two. For small dimensions d, the MVEE problem can be solved in $O(d^{O(d)}m)$ operations using randomized or determistic algorithms[2]. In this paper the algorithm based on Khachiyan solution is used.

An ellipsoid in center form is given by

$$E = \{x \in \mathbb{R}^n | (x - c)^T A(x - c) \leq 1\}$$

where $c \in \mathbb{R}^n$ is the center of the ellipse E and $A \in \mathbb{S}^n_{++}$ Points lying inside the ellipse satisfy

$$(x_i - c)^T A(x_i - c) \leq 1$$

The pseudocode for the MVEE algorithm used in this paper is as follows:

```
while err > tolerance
        X = Q * diag(u) * Q'
        M = diag(Q' * inv(X) * Q)
        [maximum j] = max(M)
        step_size = (maximum - d -1)/((d+1)*(maximum-1))
        new_u = (1 - step_size)*u
        new_u(j) = new_u(j) + step_size
        count = count + 1
        err = norm(new_u - u)
        u = new_u
end
```

3.2 Native Elements Classification

Support Vector Machines are a set of supervised learning methods used for classification, regression and outliers detection. They are effective in high dimensional spaces, memory efficient and quite versatile with many kernel functions that can be specified for the decision function. Although in some cases, where number of features is much greater than the number of samples, this method can give poor results, and is not cost-efficient when calculating probability estimates, it is well suited for problem presented in this paper. Given a set of training examples, each marked for belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other. For multi-class classification "one-against-one" approach can used. For n classes n * (n - 1) / 2 classifiers are constructed, each trains data from two different classes. The order for classes 0 to n is "0 vs 1", "0 vs 2", ..."0 vs n", "1 vs 2", ..."n - 1 vs n".

Random Forests method is based on classification trees, which are used to predict membership of objects in the classes. For vector of independent variables representing one object they calculate the value of the class the object belongs to by dividing value space into two or more subspaces. More precisely, an input data is entered at the top of the tree and as it traverses down the tree the data gets bucketed into smaller and smaller sets. The main principle behind the Random Forest method is that a group of "weak learners" can come together to form a "strong learner". After a large number of trees is generated, they vote for the most popular class. We call these procedures random forests.

K-Nearest Neighbors

3.3 Quality Evaluation

- CC (Correctly Classified) - the number of correctly classified patterns, i.e. native patterns classified as native ones with the correct class,
- TP (True Positives) - the number of native patterns classified as native (no matter, into which native class),
- FN (False Negatives) - the number of native patterns incorrectly classified as foreign,
- FP (False Positives) - the number of foreign patterns incorrectly classified as native,
- TN (True Negatives) - the number of foreign patterns correctly classified as foreign.

$$\text{Accuracy}=\frac{\text{TP+TN}}{\text{TP+FN+FP+TN}}$$

$$\text{Strict Accuracy}=\frac{\text{CC+TN}}{\text{TP+FN+FP+TN}}$$

$$\text{Native Precision}=\frac{\text{TP}}{\text{TP+FP}}$$

$$\text{Native Sensitivity}=\frac{\text{TP}}{\text{TP+FN}}$$

$$\text{Strict Native Sensitivity}=\frac{\text{CC}}{\text{TP+FN}}$$

$$\text{Fine Accuracy}=\frac{\text{CC}}{\text{TP}}$$

$$\text{Foreign Precision}=\frac{\text{TN}}{\text{TN+FN}}$$

$$\text{Foreign Sensitivity}=\frac{\text{TN}}{\text{TN+FP}}$$

$$\text{F–measure}=2\cdot\frac{\text{Precision}\cdot\text{Sensitivity}}{\text{Precision}+\text{Sensitivity}}$$

- *Strict Accuracy* is the absolute measure of the classifier's performance. It is the ratio of the number of all *correctly* classified patterns, i.e. native patterns classified to their respective classes and rejected foreign ones to the number of all patterns being processed.

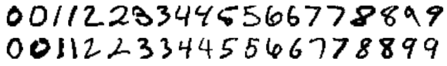

Fig. 1. ...

- *Accuracy* is a characteristic derived from strict accuracy by ignoring the need to classify native patterns to their respective classes; in other words, it is sufficient to correctly identify whether a pattern is native or foreign one. This measure describes the ability to distinguish between native and foreign patterns.
- *Native Precision* is the ratio of the number of not rejected native patterns to the number of all not rejected patterns (i.e. all not rejected native and foreign ones). Native Precision evaluates the ability of the classifier to distinguish native patterns from foreign ones. The higher the value of this measure, the better ability to distinguish foreign elements from native ones. Native Precision does not evaluate how effective identification of native elements is.
- *Native Sensitivity* is the ratio of the number of not rejected native patterns to all native ones. This measure evaluates the ability of the classifier to identify native elements. The higher the value of Native Sensitivity, the more effective identification of native elements. Unlike the Native Precision, this measure does not evaluate the effectiveness of separation between native and foreign elements.
- *Strict Native Sensitivity* takes only correctly classified native patterns and does not consider native patterns, which are not rejected and assigned to incorrect classes, unlike *Native Sensitivity*, where all not rejected native patterns are taken into account.
- *Fine Accuracy* is the ratio of the number of native patterns classified to correct classes, i.e. assigned to their respective classes, to the number of all native patterns not rejected. This measure conveys how precise is correct classification of not rejected patterns.
- *Foreign Precision* corresponds to Native Precision.
- *Foreign Sensitivity* corresponds to Native Sensitivity.
- Precision and Sensitivity are complementary and there exists yet another characteristic that combines them: the *F–measure*. It is there to express the balance between precision and sensitivity since these two measures affect each other. Increasing sensitivity can cause a drop in precision since, along with correctly classified elements, there might be more incorrectly classified,

4 Experiments

4.1 Presentation of Datasets

Figure 1 presents native and foreign patterns ...

Table 1. Results for classification with rejection on train and test sets of native patterns in comparison with classification results without rejection mechanism. RF - results for random forest, SVM - results for Support Vector Machines, ...

	no rejection			with rejection		
Basic Classifier	RF	SVM	KNN	RF	SVM	KNN
Data Set	Native Patterns, Train Set					
Fine Accuracy						
Strict Native Sensitivity						
Native Sensitivity						
Data Set	Native Patterns, Test Set					
Fine Accuracy						
Strict Native Sensitivity						
Native Sensitivity		–	–		–	–

4.2 Impact on Classification

4.3 Rejection Quality

Table 2. Results of classification with rejection on the set of native patterns supplemented with different sets of semi-synthetic foreign patterns....

Basic Classifier	RF	SVM	KNN	RF	SVM	KNN
Data Set	xxx			x x		
Strict Accuracy						
Accuracy						
Native Precision						
Native Sensitivity						
Foreign Precision						
Foreign Sensitivity						
Native F-measure						
Foreign F-measure						
Data Set	yyy			zzz		
Strict Accuracy						
Accuracy						
Native Precision						
Native Sensitivity						
Foreign Precision						
Foreign Sensitivity						
Native F-measure						
Foreign F-measure						

5 Conclusion

Proposed ...

In future ...

Acknowledgment

The research is partially supported by the .....

References

1. Hempstalk, K., Frank, E., Witten, I., *One-class classification by combining density and class probability estimation*, Machine Learning and Knowl. Disc. in Databases, pp. 505-519, 2008.
2. Michael J. Todd, E.Alper Yildirim, *On Khachiyan's Algorithm for the Computation of Minimum Volume Enclosing Ellipsoids*, September 30, 2005, article link: http://people.orie.cornell.edu/miketodd/TYKhach.pdf