LARGE MARGIN METHODS FOR CONSTRUCTING POLYTOPE CLASSIFIERS

István PILÁSZY
Advisor: Tadeusz DOBROWIECKI

I. Introduction

There are many tasks that require the prediction of missing data. Some of them are text document categorization, spam filtering, handwritten digit recognition, credit screening, weather forecasting, etc. The missing data for these tasks are the labels of the documents, classification as spam or not spam, the value of digit, decision to grant or not to grant credit, or the weather for the next day.

Many methods have been developed to yield faster or more accurate solutions to these problems [1].

In this paper we propose a new machine learning (ML) algoritm for binary classification tasks. The algorithm is a modification of a multiclass classification method, suggested in [2]. We will show some preliminary results, based on artificial datasets.

II. Machine learning methods

To be able to compare our new method to the existing ones, we need some classification of classification algorithms. *Supervised machine learning* is the learning of a mapping from input-output patterns. [3] We distinguish between binary classification and multiclass classification. In case of binary classification, we choose between two classes, the positive (e.g. spam) and the negative (e.g. not spam), as opposed to multiclass classification, when we have more classes. In both cases each example belongs to exactly one class.

One of the most important and simplest classifiers are the linear classifiers. They require the examples to be represented in $\mathbb{R}^n$, and use the following decision function:

$$y = \operatorname{sign}(\mathbf{w}^{\mathrm{T}}\mathbf{x} + b) \quad \mathbf{w} \in \mathbb{R}^n,\ \mathbf{x} \in \mathbb{R}^n,\ b \in \mathbb{R} \tag{1}$$

The $\mathbf{w}$ vector and the b scalar are the parameters of the model. There are a plenty of methods to find a suitable $(\mathbf{w}, b)$ solution: Support Vector Machines, Centroid, Adaline, Perceptron, Naive Bayes, Logistic Regression, etc. [4][3][1].

III. Support Vector Machines

During the last few years Support Vector Machines (SVMs) have gained a lot of popularity, and have been proven as one of the most powerful learning algorithms for various problems. Suppose we are provided with examples $\mathbf{x}_i$ and their labeling $y_i \in \{-1, 1\}$. The H hyperplane separating positive and negative examples is formulated by:

$$H = \{\mathbf{x} \mid \mathbf{w}^{\mathrm{T}}\mathbf{x} + b = 0 \quad \mathbf{x} \in \mathbb{R}^n\} \tag{2}$$

In case of an SVM, $\mathbf{w}$ and b are the solutions of the following convex quadratic programming (QP) problem:

$$\text{minimize: } V(\mathbf{w}, b) = \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + C\sum_i \xi_i \quad \text{subject to: } (\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b)\cdot y_i \geq 1 - \xi_i,\ \ \xi_i \geq 0 \tag{3}$$

Where C is a user-defined constant, which should be set to trade off between large margin ($1/||\mathbf{w}||$) and small training error ($\sum_i \xi_i$). Here ξ_i is a slack variable representing the training error on the ith example.

The dual of this QP is the following optimization problem [4]:

$$\text{maximize: } W(\alpha) = -\frac{1}{2}\sum\sum \alpha_i \alpha_j y_i y_j \mathbf{x}_i \mathbf{x}_j \tag{4}$$

$$\text{subject to: } 0 \leq \alpha_i \leq C, \ \sum \alpha_i y_i = 0 \tag{5}$$

The optimal $\mathbf{w}$ is equal to $\sum \alpha_i \mathbf{x}_i y_i$ [4]. Those vectors, for which $\alpha_i > 0$, are called support vectors. Only these vectors influence the model.

IV. Multiclass SVMs

There are many approaches to solve multiclass problems. Almost all of them rely on the binary classification algorithms. In [2] a different approach is proposed:

Let $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_i, y_i), \ldots, (\mathbf{x}_m, y_m)\}$ be the set of training examples, and $Y = \{1, ..., k\}$ be the class labels ($y_i \in Y$) and $M = \{(1, \mathbf{w}_1, b_1), ..., (j, \mathbf{w}_j, b_j), \ldots, (k, \mathbf{w}_k, b_k)\}$ be the multiclass model. For a given example $\mathbf{x}$, the decision function is defined as:

$$f(\mathbf{x}) = \arg\max_{j \in Y} \mathbf{w}_j^{\mathrm{T}} \mathbf{x} + b_j \tag{6}$$

The M model is the solution of the following QP-problem:

$$\text{minimize: } V(M) = \sum_{j \in Y} \frac{1}{2} \mathbf{w}_j^{\mathrm{T}} \mathbf{w}_j + C \sum_{i=1}^{m} \xi_i \tag{7}$$

$$\text{subject to: } \forall_{i=1}^{m} \forall_{j \in Y \setminus y_i} \quad (\mathbf{w}_{y_i}^{\mathrm{T}} \mathbf{x}_i + b_{y_i}) - (\mathbf{w}_j^{\mathrm{T}} \mathbf{x}_i + b_j) \geq 1 - \xi_i, \ \ \xi_i \geq 0 \tag{8}$$

V. Polytope classifiers

Convex polytope is the convex hull of a finite set of points [5]. Convex polytopes can be represented as the intersection of a finite number of half-spaces. Note that in case of SVMs SVM partitioned the space into two halves: one for positive examples, the other for negatives. Consider the example depicted in fig.1a. This problem cannot be solved with a linear classifier. Nevertheless SVM has a non-linear extension, which is able to solve non-linearly separable problems using the following function:

$$f(\mathbf{x}) = \text{sign}\left(b + \sum \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i)\right) \tag{9}$$

Where $K(\cdot, \cdot)$ is a user-supplied kernel-function, which must hold some properties [4]. Roughly speaking, the more similar the $\mathbf{x}$ and $\mathbf{x}_i$, the higher the value of $K(\mathbf{x}, \mathbf{x}_i)$. However, when we are given a lots of training examples, to decide a category of an unseen example, the method needs to calculate $K(\mathbf{x}, \mathbf{x}_i)$ for all i, and this could be slow, compared to the linear approach (in that case, $K(\cdot, \cdot)$ is the scalar product, and the support vectors can be summed up).

To overcome this problem, we may use *convex polytope classifiers*: given a convex polytope, it classifies points inside the polytope as negative (inner class), otherwise positive (outer class).

In [6] there are some negative results for constructing classifiers: „It is NP-complete to recognize whether two sets of points in general space can be separated by two hyperplanes." But „For every

fixed k in any fixed dimension, it takes polynomial time to recognize whether two sets of points can be separated with k hyperplanes." Thus in arbitrary high dimensions we can give up fixing k for a small (minimal) value. Note that if a binary classification problem can be solved with a convex polytope classifier, it is very easy to construct one:

- for each $\mathbf{x}_i$ point in the outer (positive) class, separate it from the inner class. This yields $\mathbf{w}_i$,b_i.
- for an unseen point $\mathbf{x}$, compute $\mathrm{sign}(\max_i \mathbf{w}_i^{\mathrm{T}}\mathbf{x} + b_i)$, which indicates the class label.

VI. Proposed method

The multiclass SVM described in section IV. can easily be adopted to construct a binary convex polytope classifier: We create a separate class for each positive example (but do nothing on negatives): we ignore the constraint of eq.8 between positive classes. Thus, each positive example will be separated from each negative (if feasible), but there may be mistakes between positive class labels (but we do not take care of it).

Formally: suppose we are given with a binary classification task ($Y = \{-1, 1\}$) and the outer class is the positive, and contains m^+ examples. If not, then negate the class labels. Let the first m^+ example be positive, and the rest negative. Let $Y = \{-1\} \cup \{1, \ldots, m^+\}$, and $\forall_{i=1}^{m^+}\ y_i := i$. The M model is the solution of the following QP-problem:

$$\text{minimize: } V(M) = \sum_{j \in Y} \frac{1}{2}\mathbf{w}_j^{\mathrm{T}}\mathbf{w}_j + C\sum_{i=1}^{m}\xi_i \tag{10}$$

$$\text{subject to: } \forall_{i=1}^{m^+} \quad (\mathbf{w}_{y_i}^{\mathrm{T}}\mathbf{x}_i + b_{y_i}) - (\mathbf{w}_{-1}^{\mathrm{T}}\mathbf{x}_i + b_{-1}) \geq 1 - \xi_i, \tag{11}$$

$$\forall_{i=(1+m^+)}^{m}\forall_{j=1}^{m^+} \quad (\mathbf{w}_{y_i}^{\mathrm{T}}\mathbf{x}_i + b_{y_i}) - (\mathbf{w}_j^{\mathrm{T}}\mathbf{x}_i + b_j) \geq 1 - \xi_i, \ \ \forall_{i=1}^{m}\xi_i \geq 0 \tag{12}$$

The classification of an unseen example is performed by: $f(\mathbf{x}) = \mathrm{sign}\left(\arg\max_{j \in Y} \mathbf{w}_j^T\mathbf{x} + b_j\right)$.

Note that the optimal solution can be found in polynomial time, and the proposed method is a large margin method.

VII. Preliminary results

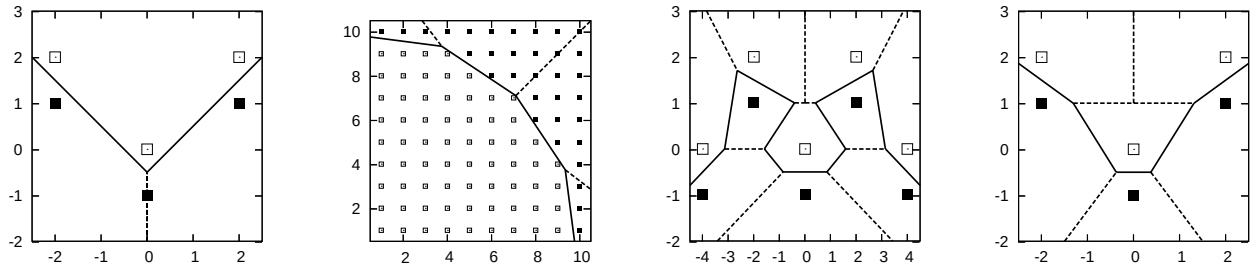
To demonstrate the capabilities of the proposed method, we have evaluated it on some two-dimensional artificial datasets.

In fig.1a there is a very simple classification problem. Linear SVMs are unable to solve it. The proposed approach can solve it ($C = 10$, see fig.1a). With a Gaussian kernel ($K(\mathbf{x}_1, \mathbf{x}_2) = \exp(-\gamma||\mathbf{x}_1 - \mathbf{x}_2||^2)$) SVM finds a similar (but smooth) solution. In fig.1b negative examples are inside a circle, positives are outside (100 examples total, $C = 10000$). Our method solves it with 4 weight vectors (plus bias), SVM with a gaussian kernel uses 93 support vectors and one bias. This means a factor of 7-8 in speed and memory consumption when classifying an unseen example. We can generalize the proposed approach: we create a separate class even for each negative example (fig.1c-d, $C = 100$).

VIII. Conclusions

Advantages of the proposed algorithm:

- It can easily be generalized to solve even more general problems: if we create separate classes for each example (not only for positives), it is even more general (i.e. not only convex polytopes). Multiclass (not only binary) polytope classification tasks can also be solved.
- The classification phase is fast, the learning method is deterministic, the optimum point always exists, and can be found in polynomial time.



a) b) c) d)

Figure 1: Classification problems and their solutions. Dashed line means inner-class separation

- It solves the convex separability problem with a large-margin method.
- The solution is ,,intelligent", i.e. it can merge positive classes, and uses a few number of facets.

Disadvantages:

- It is terribly slow when the outer class contains a lots of points.
- We do not know in advance, how many facets are required (we could not know). We have no control over the number of facets.
- Maximizing the margin has no mathematical background as opposed to SVMs (i.e. minimizes the expected classification error).

IX. Future work

The results are very promising, however we need to do further investigations:

- We should try out the kernelized version of multiclass classification kernel functions, to create even more general classifiers (possibly with few support vectors).
- Speed up the proposed method by simplifying the QP-problem: Some clustering on the outer class would speed up the method because of the fewer classes. We could use a small C and solve the QP problem. From this, we hope to have a small number of positive classes. We could consider only these classes as the covering of the positive set, and solving this new problem (fewer positive classes) with a higher C. Another approach is to cut down with some hyperplane as many outer points as we can, create a cluster for them, and repeating this greedy method until each point is in a cluster.
- There are more open-source implementations of the multiclass SVM, with lots of speed-up tricks. Can they be modified to construct polytope classifiers while preserving their speeds?
- If we put some labeled points in the space, and assign each point of the space to the nearest of these points, we get a nearest neighbour classifier. The relationship between k-NN and polytope classifiers should be investigated.

References

[1] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2nd Edition)*, Wiley-Interscience, 2000.

[2] K. Crammer and Y. Singer, "On the algorithmic implementation of multiclass kernel-based vector machines," *Journal of Machine Learning Research*, 2:265–292, Dec. 2001.

[3] S. Russell and P. Norvig, *Artifical Intelligence: A Modern Approach*, Prentice-Hall, Englewood Cliffs, NJ, second edition, 2002.

[4] H. Gábor, "Support vektor gépek (support vector machines)," 2002,
URL: http://www.mit.bme.hu/˜horvath/supportvector.pdf.

[5] H. S. M. Coxeter, *Introduction to Geometry*, Wiley, New York, second edition, 1969.

[6] N. Megiddo, "On the complexity of polyhedral separability," *Discrete and Computational Geometry*, 3:325–337, 1988.