Learning from Positive and Unlabelled Examples Using Maximum Margin Clustering

Sneha Chaudhari[1,*] and Shirish Shevade[2,**]

[1] IBM Research Lab, Bangalore, India
snechaud@in.ibm.com
[2] Indian Institute of Science, Bangalore, India
shirish@csa.iisc.ernet.in

Abstract. Learning from Positive and Unlabelled examples (LPU) has emerged as an important problem in data mining and information retrieval applications. Existing techniques are not ideally suited for real world scenarios where the datasets are linearly inseparable, as they either build linear classifiers or the non-linear classifiers fail to achieve the desired performance. In this work, we propose to extend maximum margin clustering ideas and present an iterative procedure to design a non-linear classifier for LPU. In particular, we build a least squares support vector classifier, suitable for handling this problem due to symmetry of its loss function. Further, we present techniques for appropriately initializing the labels of unlabelled examples and for enforcing the ratio of positive to negative examples while obtaining these labels. Experiments on real-world datasets demonstrate that the non-linear classifier designed using the proposed approach gives significantly better generalization performance than the existing relevant approaches for LPU.

Keywords: Learning from Positive and Unlabelled Examples, Maximum Margin Clustering, Least Squares Support Vector Classifier.

1 Introduction

Many applications of information retrieval and data mining face binary classification problems which typically involve datasets consisting of a small set of positive examples and a large number of unlabelled examples. This problem of Learning from Positive and Unlabelled examples (LPU) occurs in situations where either characterizing negative examples is difficult or their annotation is expensive. Consider the real world application of Junk Mail Filtering [1]. Here, the junk messages serve as positive examples as they can be distinguished from legitimate mails in terms of style and vocabulary; they are independent of individual users and, hence, easier to characterize and annotate. Consequently, the

[*] This work was done while the author was a graduate student in the Dept of CSA, IISc, India.

[**] The author gratefully acknowledges the support for this work from Infosys Ltd, India.

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 465–473, 2012.
© Springer-Verlag Berlin Heidelberg 2012

aim is to learn to filter junk mails automatically to improve the usability of an e-mail client.

Motivation and Related Work: Many of the existing approaches for handling the problem of LPU [2], [3] construct a linear classifier. These approaches do not achieve the desired performance for some real world scenarios, as linear classifiers are not sufficient where the datasets are linearly inseparable. To remedy this, Support Vector Machines (SVM) based approaches have been proposed which can obtain a non-linear classifier by employing a kernel function. However, as observed in [4], SVM based approaches suffer from the risk of premature convergence due to the asymmetry of the hinge loss function of SVMs. Further, existing techniques do not enforce the class balance ratio, i.e., ratio of positive to negative examples in the unlabelled data, which is useful for avoiding trivial solutions and obtaining better generalization performance.

For example, consider a one-class SVM proposed in [5] which uses only positive examples for learning, resulting in poor performance. Further, iterative SVM based approaches have been proposed where the final classifier is either the last classifier obtained after convergence [6], or a selected classifier from the set of classifiers built [7]. However, for training the SVM, these methods obtain the labels of unlabelled examples using different techniques. A cost asymmetric SVM formulation, called Biased-SVM (BSVM) is proposed in [3]. BSVM uses two parameters to assign a higher weight to positive errors in comparison to negative errors. Further, it uses the Naive Bayes (NB) classifier for initializing the labels of unlabelled examples. One more approach based on similar ideas of BSVM method is given in [8], where a probabilistic approach is followed to assign the weights to positive and unlabeled examples. Another interesting approach is presented in [9], where a Positive Naive Bayes (PNB) classifier is constructed by adapting the NB classifier to handle the problem of LPU. Recently, a practical approach for Maximum Margin Clustering (MMC) has been proposed in [4]. MMC performs clustering by finding a decision surface passing through low density region in the data. The optimization problem in MMC is non-convex and an iterative procedure is adopted in [4], using Support Vector Regression with Laplacian loss to avoid premature convergence.

Contributions: In this work, we extend the idea of iterative learning adopted in [4] to the problem of LPU and design a *non-linear* classifier. The classifier is designed using Least Squares SVM (LS-SVM) [10] method. It effectively handles the non-convexity of the optimization problem involved, by virtue of a symmetric loss function. Positive examples and the class balance ratio are used to determine the bias term in the classification model. This helps to avoid trivial solutions and improve the performance on unseen data. As the class balance ratio is not exactly known in practice, we experimentally show that the proposed approach is useful even if the value is approximately known. Further, appropriate initialization of the labels of unlabelled examples is crucial in this problem set-up and we propose a simple technique for this purpose, which is effective in improving the performance. Though maximum margin classification ideas have been used in past for semi-supervised learning, to the best of our knowledge, *Maximum Margin Clustering* has not been explored before to handle the problem of LPU. Experimental results on real-world datasets demonstrate that the proposed approach is useful for designing a non linear classifier with significantly improved generalization performance than existing techniques such as Iterative SVM (ISVM), BSVM and PNB.

2 Proposed Approach: Maximum Margin Clustering with Least Squares SVM (MCLS)

The problem of learning from positive and unlabelled training examples is to obtain a binary classifier, given a training set consisting of N examples, where the first L examples, $\{x_i, +1\}_{i=1}^{L}$ are positive and the remaining $U = N - L$ examples, $\{x_i\}_{i=L+1}^{N}$, are unlabelled. In this work, we design a non-linear support vector classifier of the form f(x) = $w^\top \varphi(x) + b$, where $\varphi(x)$ is a non-linear function. The underlying optimization problem is given in (1).

$$\begin{aligned} \min_{w,b,\{y_i\}_{i=L+1}^{N}} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{L} l(w^T\varphi(x_i)+b) + \sum_{i=L+1}^{N} l(y_i(w^T\varphi(x_i)+b)) \\ s.t. \quad & y_i \in \{+1,-1\} \quad \forall i = L+1 \longrightarrow N \\ & \frac{1}{U}\sum_{i=L+1}^{N} max(0, sign(w^T\varphi(x_i)+b)) = r \end{aligned} \tag{1}$$

where, C is a positive hyper-parameter which controls the trade-off between smoothness and fitness and $l(t)$ is a loss function; for example, the hinge loss function in SVMs is $l(t) = max(0, 1-t)$. As we can see, this optimization problem is a variant of Transductive SVM formulation [11], which introduces separate terms in the objective function for positive and unlabelled examples. Also, notice that it is important to add the second constraint, which specifies that a fraction r of unlabelled data is to be labelled positive. This user defined parameter ensures that the class balance ratio is maintained in the set of unlabelled examples.

This non-convex optimization problem (1) is hard to solve. Hence, we employ a practical approach to obtain a solution to LPU. The proposed approach adopts an iterative procedure that learns a non-linear LS-SVM classifier. In each iteration, we first fix the labels of unlabelled examples and optimize with respect to w and consequently, fix w and find new labels of unlabelled data. Precisely, the proposed approach consists of the following main steps : (i) We initialize the labels of unlabelled data using the algorithm explained in Subsection 2.1. (ii) To optimize with respect to w, we train LS-SVM classifier using a labelled training set obtained in (i). We make use of LS-SVM as a classification algorithm as it avoids poor local minima due to a symmetric loss function. We explain this in detail in Subsection 2.2. (iii) We determine the new labels of unlabelled data using the decision function of the LS-SVM classifier. However, these labels are computed such that the second constraint in (1) is satisfied. The proposed approach maintains r by appropriately determining the bias parameter, b. The procedure is described in Subsection 2.3. Finally, these steps are repeated until the labels of unlabelled examples remain constant in successive iterations or maximum nunber of iterations is reached. This procedure is given succinctly in Algorithm 1. Now we discuss each aspect of the method in detail in the following subsections.

2.1 Initialization of Labels of Unlabelled Data (ILU)

The initialization of labels of unlabelled examples (step 1 in Algorithm 1) is very crucial as they are used to train the LS-SVM. We propose a method for obtaining these labels which is effective in improving the accuracy. Initially, k-means clustering is performed on the training data. Each cluster is determined as positive or negative, according to the number of positive examples present in that cluster. To obtain negative examples, some examples are chosen from each of the negative clusters which are farthest from the centroid of positive examples. The intuition is to select those examples from the unlabelled data, which have higher probability of belonging to the negative class. The number of examples to be selected depends on the value r of the dataset. Now, any supervised classification technique can be used for training where the input is the set of positive examples and the selected negative examples; we use SVM in our algorithm. Finally, the labels of all the unlabelled examples are obtained using the decision function of the classifier.

2.2 Non-linear LS-SVM Classifier

The LS-SVM formulation for a completely labelled dataset $\{x_i, y_i\}_{i=1}^N$ can be given as follows:

$$\begin{aligned} \min_{w,b,\xi_i} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i^2 \\ \text{s.t.} \quad & y_i - (w^\top \varphi(x_i) + b) = \xi_i \quad \forall i \end{aligned} \tag{2}$$

The main benefit of LS-SVM classifier is that it is well suited for solving this problem due to symmetry of its loss function [4]. The loss remains the same, if the label of an unlabelled example is changed during training. This encourages necessary flipping of labels and classifier improves over the initial labels. This in turn helps to avoid many poor local minima and obtain a better solution.

2.3 Maintaining Class Balance Ratio

Maintaining the class balance ratio, r in the labels of unlabelled data (step 4 in Algorithm 1) is necessary to avoid trivial solutions such as assigning all examples to one class to obtain an unbounded margin hyperplane. The proposed algorithm performs a simple, efficient and easy to implement computation of the bias value (b) of LS-SVM to maintain this ratio. At the same time, the algorithm also tries to set b such that the labels of positive examples remain constant while finding the labels of unlabelled examples, a critical necessity for applications of LPU. The algorithm uses r and a tolerance parameter which decides the trade-off between maintaining the class balance and correctly classifying the positive examples. The algorithm sorts $w^T\varphi(x)$ values and sets the bias value to the $w^T\varphi(x)$ value satisfying r. The algorithm now checks if all the positive examples are correctly classified. Otherwise, the bias is changed in the range of the tolerance parameter such that maximum number of positive examples are correctly classified.

Algorithm 1 MCLS

```
Input: Training set $\{x_i, +1\}_{i=1}^{L} \cup \{x_i\}_{i=L+1}^{N}$, where $y_i \in \{+1, -1\}, \forall i = L+1 \longrightarrow N$
Output: Classifier : f(x) = $w^\top\varphi(x) + b$
Find labels of unlabelled examples ($\{\bar{y_i}\}_{i=L+1}^{N}$) using algorithm described in 2.1
while TRUE do
    Perform LS-SVM training using $\{x_i, +1\}_{i=1}^{L} \cup \{x_i, \bar{y_i}\}_{i=L+1}^{N}$ and compute $w$
    Compute the bias value ($\hat{b}$) using the method described in 2.3
    Obtain new labels: $\hat{y_i}$ using $w$ and bias value $\hat{b}$
    i.e. $\hat{y_i} = \text{sign}(w^\top\varphi(x_i) + \hat{b}) \quad \forall i = 1 \longrightarrow N$
    if $\bar{y_i} == \hat{y_i}\ \forall i = 1 \longrightarrow N$ then
        Break
    else
        $\bar{y_i} = \hat{y_i}\ \forall i = 1 \longrightarrow N$
    end if
end while
$b = \hat{b}$
```

3 Experimental Evaluation

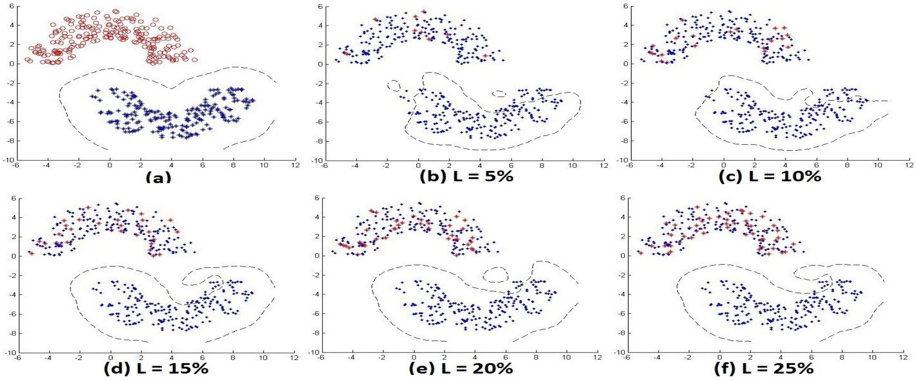
The experimental study was conducted on seven real world datasets, as given in Table 1. The six datasets in Table 1 except ionosphere are available at http://theoval.cmp.uea.ac.uk/∼ gcc/matlab/default.html#benchmarks. The ionosphere dataset has been taken from the UCI machine learning repository [12]. MCLS, Naive Bayes (NB), Iterative SVM (ISVM) and Positive Naive Bayes (PNB) were implemented in Matlab (version R2010a). For all the experiments, we used RBF kernel function defined as : $K(x_i, x_j) = exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2})$. Also, the generalization performance of the classifiers designed using different techniques was studied as we increased the number of positive examples (L). In particular, we chose 5, 10, 15, 20 and 25% of actual positive class examples in the training set.

Demonstration of MCLS on a Toy Dataset: We consider a two dimensional toy dataset to demonstrate the decision boundaries obtained by MCLS. The dataset consists of 400 examples with $r = 0.5$. Figure 1(a) shows the decision boundary obtained using a completely labelled training set. The rest of the

Table 1. Details of Datasets. N: Total number of examples, TR: Number of training examples, TS: Number of test set examples, **r**: Class balance ratio, ATS: Accuracy on test set

Dataset	**N**	**TR**	**TS**	**r**	**ATS(%)**
banana	5300	3533	1767	0.4483	90.26
thyroid	215	143	72	0.3	97.22
heart	270	180	90	0.4444	83.22
pima	768	512	256	0.349	74.6
waveform	5000	3333	1667	0.3294	90.53
ringnorm	7400	4934	2466	0.495	98.78
ionosphere	351	234	117	0.641	96.583

plots, Figures 1(b)-1(f), show the decision boundaries given by MCLS for different values of L. The plots clearly show the efficacy of the proposed algorithm. In particular, for $L = 15\%$ (Figure 1(d)), the decision boundary is very close to the one obtained using the completely labelled data. We also show decision boundaries given by MCLS and other techniques in Figure 2 with L=15%. Note that MCLS obtains a reasonable decision boundary than other existing techniques, when only 15% positively labelled examples are used.



(a)

(b) L = 5%

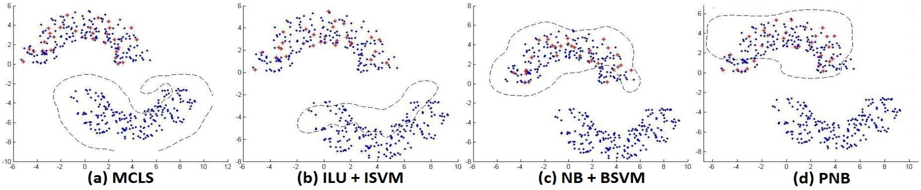
(c) L = 10%

(d) L = 15%

(e) L = 20%

(f) L = 25%

Fig. 1. Decision boundary obtained by MCLS algorithm as L increases. Positive and Unlabelled examples are shown by red stars and blue dots respectively. (a) Shows the decision boundary obtained using labelled training set.



(a) MCLS

(b) ILU + ISVM

(c) NB + BSVM

(d) PNB

Fig. 2. Decision boundary obtained by MCLS and existing techniques with L=15%

Generalization Performance of MCLS: In Table 2, we report the test set accuracies of MCLS as a function of the number of positive examples, compared with following methods (1) ILU (Subsection 2.1) + ISVM [6]. Here, after initialization using ILU, SVM is trained iteratively and the last classifier obtained after convergence is selected. (2) NB + BSVM [3] and (3) PNB [9]. MCLS shows significantly better accuracies for all datasets when compared to the rest of the three algorithms. The iterative SVM does not perform well as it faces the problem of getting stuck in poor local minima. The BSVM method, though assigns different weights to positive and unlabelled examples, does not focus on maintaining the r fraction in the labels of unlabelled data. The PNB method does not show comparable performance. Further, the difference in the accuracies is prominent for small values of L. This demonstrates the applicability of MCLS for datasets with small number of positive examples. For the datasets heart, pima, ionosphere and banana, the performance using $L = 25\%$ is comparable with that obtained using a completely labelled training set.

Performance Evaluation of ILU: To evaluate the algorithm described in Subsection 2.1, we compared the accuracies obtained over unlabelled data with one popular approach proposed in [3] for initialization of labels. The authors construct a NB Classifier by treating all unlabelled examples as negative. The results are given in Table 3. ILU outperforms NB on almost all the datasets. Also, ILU shows greater increase in the accuracy as we increase L compared to NB. The reason is that NB is constructed by treating all unlabelled examples as negative whereas ILU algorithm constructs SVM classifier by extracting negative examples from unlabelled examples.

Table 2. Comparison of Test set Accuracies of MCLS, Iterative SVM (ISVM), Biased SVM (BSVM), Positive Naive Bayes (PNB) Algorithms

L	10%				15%				20%				25%			
Dataset	MCLS	ILU + ISVM	NB + BSVM	PNB	MCLS	ILU + ISVM	NB + BSVM	PNB	MCLS	ILU + ISVM	NB + BSVM	PNB	MCLS	ILU + ISVM	NB + BSVM	PNB
banana	**80.9**	72	70	63.7	**81.8**	73.1	75.3	67.6	**84.6**	77.5	81.4	68.9	**87.4**	82.1	85.7	70.5
thyroid	81.9	**84.2**	80.5	74.4	86.6	**87.3**	86.1	76.3	**90.7**	90.3	87.5	79.2	**93**	91.6	91	80.9
pima	**67.9**	64.8	64.4	60.9	**68.3**	66.7	67.5	65.1	**71.4**	67.8	68.1	66.4	**73**	68.7	69.9	69.9
ionosphere	**82.3**	75.2	78.6	71.8	**88**	76.3	77.7	76.1	**90.5**	80.3	85.4	79.4	**94**	81.2	90.4	83.7
heart	**72.1**	67.7	67.9	63.2	**73.3**	68.8	69.9	66.5	**78.1**	74.1	73.6	70.7	**81.4**	77.8	78	72.2
waveform	**78.3**	75.1	70.4	70.8	**79.4**	75.9	70.8	72.6	**82.4**	77.3	70.9	74.2	**82.7**	78.3	71.1	74.9
ringnorm	**92.5**	87.7	88.6	87	**92.6**	89.6	89.5	88.2	**94.4**	90.9	91.4	89.7	**94.8**	91.1	92	90.2

Table 3. Comparison of Accuracies over unlabelled data of ILU and NB

L	**5%**		**10%**		**15%**		**20%**		**25%**	
Dataset	**ILU**	**NB**	**ILU**	**NB**	**ILU**	**NB**	**ILU**	**NB**	**ILU**	**NB**
banana	**72.3**	54.5	**73.8**	59.3	**74**	62.5	**77.1**	64.8	**77.6**	64.2
pima	**66.1**	58.3	**66**	59.3	**66.9**	58.9	**67.7**	60.4	**66.9**	60.7
heart	**68.3**	51.3	**69.5**	60.4	**70.6**	58.3	**73.9**	62	**77.6**	67.7
ionosphere	**67.3**	55.6	**72.3**	65.3	**74.2**	65.6	**75.8**	63.9	**76.4**	69.5
ringnorm	**90**	86	**90.5**	85	**90.4**	87.1	**91**	87.4	**91.1**	87.6
thyroid	**78.6**	58.6	**79.7**	73.2	**83.4**	82.3	**88.2**	86	**89.1**	86.7
waveform	**76.3**	74.6	**78.2**	76.1	**78.8**	77.4	**79.8**	78.5	77.3	**79.7**

Variation of r Fraction: The parameter r (fraction of positive examples in unlabeled data) is typically not exactly known in practice. We therefore conducted an experiment to study the generalization performance of the classifier designed using the proposed method, when r is varied in a small interval around its true value. The results are reported in Table 4 for four datasets. It is evident from this table that is no significant degradation in the generalization performance in small neighborhood of r. Thus, the proposed approach is useful even if the value of parameter r is approximately known.

Table 4. Test set Accuracy of MCLS as a function of parameter r

Dataset	**r**	**r-0.15**	**r-0.1**	**r-0.05**	**r+0.05**	**r+0.1**	**r+0.15**
banana	**84.6**	79.7	81	82.5	81.8	80.7	78.4
pima	**71.4**	69.2	70.7	71	69.1	68.7	67.5
heart	**78.1**	71.1	73.3	75.5	75.5	72.2	68.8
thyroid	**90.7**	86.1	86.8	87.5	88.8	87.5	83.3

4 Conclusion

In this work, we consider the problem of learning from positive and unlabelled examples by proposing a new approach to build a Least Squares support vector classifier, based on Maximum Margin Clustering. The proposed approach is particularly useful for real world applications where there is necessity of non-linear classifiers with good generalization performance. The proposed method gives significantly better accuracy than exiting techniques, especially with small number of positive examples. We also performed experiments with different values of r in the range of $r \pm 0.15$. The proposed approach showed minor degradation in the performance as r was varied in the specified range. Thus, the proposed approach is an useful alternative for learning from positive and unlabelled examples.

References

1. Schneider, K.-M.: Learning to Filter Junk E-Mail from Positive and Unlabeled Examples. In: Su, K.-Y., Tsujii, J., Lee, J.-H., Kwong, O.Y. (eds.) IJCNLP 2004. LNCS (LNAI), vol. 3248, pp. 426–435. Springer, Heidelberg (2005)
2. Zhang, B., Zuo, W.: Learning from Positive and Unlabeled Examples: A Survey. In: Yu, F., Luo, Q. (eds.) International Symposium on Information Processing, pp. 650–654. IEEE Computer Society (2008)
3. Liu, B., Dai, Y., Li, X., Lee, W.S., Yu, P.S.: Building Text Classifiers Using Positive and Unlabeled Examples. In: Proceedings of the 3rd IEEE International Conference on Data Mining, pp. 179–188 (2003)
4. Zhang, K., Tsang, I.W., Kwok, J.T.: Maximum Margin Clustering Made Practical. IEEE Transactions on Neural Networks 20(4), 583–596 (2009)
5. Manevitz, L.M., Yousef, M.: One-class SVMs for Document Classification. Journal of Machine Learning Research 2, 139–154 (2001)
6. Zhang, B., Zuo, W.: Reliable Negative Extracting Based on kNN for Learning from Positive and Unlabeled Examples. Journal of Computers 4(1), 94–101 (2009)

7. Yu, H., Han, J., Chang, K.C.C.: PEBL: Positive Example Based Learning for Web Page Classification using SVM. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 239–248. ACM Press, New York (2002)
8. Elkan, C., Noto, K.: Learning Classifiers from Only Positive and Unlabeled Data. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2008, pp. 213–220. ACM, New York (2008)
9. Calvo, B., Larraaga, P., Lozano, J.A.: Learning Bayesian Classifiers from Positive and Unlabeled Examples. Pattern Recognition Letters 28(16), 2375–2384 (2007)
10. Suykens, J.A.K., Vandewalle, J.: Least Squares Support Vector Machine Classifiers. Neural Processing Letters 9, 293–300 (1999)
11. Joachims, T.: Transductive Inference for Text Classification using Support Vector Machines. In: Proceedings of the Sixteenth International Conference on Machine Learning, pp. 200–209. Morgan Kaufmann Publishers Inc., San Francisco (1999)
12. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository (2007)