A Unique Transductive Decision Tree

S.Mohammad.Ghouse[1], E.Madhusudhana Reddy[2], R.Venkata Subbaryudu[3]

[1]Department of Computer Science & Engineering,
[2]Professor, MITS, Mpl

***Abstract*-During the last years, semi-supervised learning has emerged as an exciting new direction in machine learning research. We propose a method to use decision tree Classifier for Semi- Supervised learning, we label the unlabelled patterns using the labeled patterns and then compare these method with the traditionally Existing methods as graph mincut, spectral graph partisan, ID3,C4.5,CART, Nearest Neighbour Classifier and we are going to prove our Proposed method is more scalable than the Existing methods.**

***Key words:* semi supervised Learning, Decision Tree, Transductive Learning**

I. INTRODUCTION

Semi-supervised learning [1][2] deals with learning problems like those when the available data-set is having two parts, viz., a subset consisting of labeled data (training set) and the remaining part consisting of unlabeled data (test set). Transductive inference is limited to predicting labels for unlabeled data (test set) alone. This is in contrast to inductive methods of finding a classifier, by using the given training set, which is applicable for the entire feature space. Inductive learning is often an ill-posed problem [3][4]. Semi-supervised learning can use the location of test points (which is additional information than using only the training set) along with the training set. It is shown that for high dimensional problems having small training sets (when compared to test sets), transductive inference works better than inductive methods like conventional classifiers [1].

We present a new method for transductive learning, which can be seen as a transductive version of the Decision tree .Unlike for many other transductive learning methods, the training problem has a meaningful relaxation that can be solved globally optimally using spectral methods. We propose an algorithm that robustly achieves good generalization performance and that can be trained efficiently.

II. NOTATION AND DEFINITIONS

1) Class labels: The set of class labels, –]r +,1 _. For simplicity, a two class problem is assumed.

2) Training set: L] , , . . . , , _ is the training set, where xi is a d-dimensional feature-vector, and yi is its class label. This is also called the labeled set. l is the training set size. Training patterns with class-label is the subset $L_+ \subseteq L$, and that with class-label r is .

3) Test set: U ⁻]x_{l+1}, . . . , x_n_. This is also called the unlabeled set. u is the test set size. we have n̄ +l u.

4) Distance function: ^^$x_i - x_j$^^ is the distance between two patterns x_i and x_j. If feature-vectors are from a Euclidean space, then this is Euclidean distance. Otherwise, an appropriate distance is used (like matching coefficient) based on the feature space.

5) Inductive learner: This is a function $f^{(L)} \Omega = R^d$ .This function is learned using the training set L and can be used to predict a label for any $x \in R^d$.

6) Transductive learner: This is a function $g^{(L\cup U)} = U \rightarrow \Omega$. Note, here the domain of the function is limited to U. This function can use labeled as well as unlabeled set in predicting class label for a test pattern.

III SEMI-SUPERVISED CLASSIFICATION

While semi-supervised classification is a relatively new field, the idea of using unlabeled samples to augment labeled examples for prediction was conceived several decades ago. The initial work in semi-supervised learning is attributed to Scudders for his work on “selflearning”. An earlier work by Robbins and Monro on sequential learning can also be viewed as related to semi-supervised learning. Vapnik’s Overall Risk Minimization (ORM) principle advocates minimizing the risk over the labeled training data as well as the unlabled data, as opposed to the Empirical Risk Minimization, and resulted in transductive Support Vector Machines. Fig.1 gives the basic idea of how unlabeled data could be useful in learning a classifier. Given a set of labeled data, a decision boundary may be learned using any of the supervised learning methods (Fig. 1(a)). When a large number of unlabeled data is provided in addition to the labeled data, the true structure of each class is revealed through the distribution of the unlabeled data (Fig. 1(b)). The unlabeled data defines a “natural region” for each class, and the region is labeled by the labeled data. The task now is no longer just limited to separating the labeled data, but to separate the regions to which the labeled data belong. The definition of this "region" constitutes some of the fundamental assumptions in semi-supervised learning.

Existing semi-supervised classification algorithms may be classified into two categories based on their underlying assumptions. An algorithm is said to satisfy the manifold assumption if it utilizes the fact that the data lie on a low-dimensional manifold in the input space. Usually, the underlying geometry of the data is captured by representing the data as a graph, with samples as the vertices, and the pairwise similarities between the samples as edge-weights. Several graph based algorithms such as Label propagation [11], Markov random walks, Graph cut algorithms, Spectral graph transducer, and Low density separation proposed in the literature are based on this assumption.

The second assumption is called the cluster assumption. It states that the data

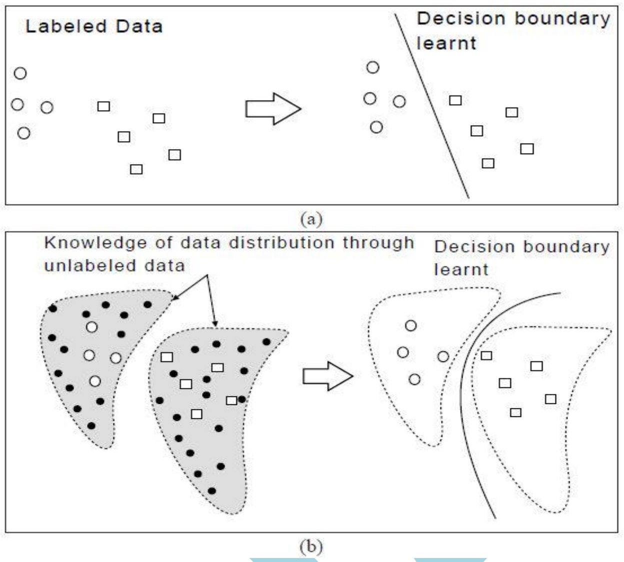


Fig 1: Utility of the unlabeled data in learning a classifier. (a) Classifier learned using labeled data alone. (b) Utility of unlabeled data. The filled dots show the unlabeled data.
The gray region depicts the data distribution obtained from the unlabeled data.

samples with high similarity between them, must share the same label. This may be equivalently expressed as a condition that the decision boundary between the classes must pass through low density regions. This assumption allows the unlabeled data to regularize the decision boundary, which in turn influences the choice of the classification models. Many successful semi-supervised algorithms like TSVM and Semi-supervised SVM follow this approach. These algorithms assume a model for the decision boundary, resulting in an inductive classifier.

Classification is a classical problem in machine learning and

data mining [1].there are some training data tuple and each having a class and it is represented by a feature vector. Here the constraint is to build an model to predicts the class label of an unknown test tuple based on the feature vector of an tuples. Among these one of the most power full classification models are decision tree model. The decision trees are very popular because they are easy to understand. Many algorithms, such as ID3 [2] and C4.5 [3], have been developed for building decision tree. These algorithms are widely brought and used in a wide range of applications such as image recognition, medical diagnosis [4], and credit rating of loan applicants, scientific tests, fraud detection, and target marketing. In traditional decision tree classification

In conventional decision tree classification an attribute of a tuple is categorical or numerical. But the data uncertainty in common in many applications. The value of a feature/attribute is thus best captured not by a single point value, but by a range of values giving rise to a probability distribution. A simple way to handle data uncertainty is to abstract probability distributions by summary statistics such as means and variances. We call this approach averaging. An another way is to handle the complete information carried by the probability distribution to build a decision tree is called efficient distribution based .In this paper we study how to constructing decision tree classifiers on data with uncertain numerical attributes. Our goal are 1) To develop an algorithm for constructing decision trees for uncertain data by using the efficient Distribution-based approach,2)to verify whether the Efficient distribution based approach could lead to a higher classification accuracy than with averaging approach, and 3)to build a theoretical foundation on pruning techniques and that can greatly improve the computational efficiency of the Efficiency Distribution-based algorithms.

III. MEASUREMENT ERRORS:

Due to measurements errors data may often imprecise. An example A new tympanic thermometer is analyzed and tested experimentally. An electrically calibrated pyro electric detector of special configuration is employed to determine a person's body temperature. An energy-storage, power-supply- isolated capacitor is used as the electrical heating reference. The new thermometer design has accuracy within 0.1 °C with a 90% confidence and is immune to ambient temperature, detector aging, and parameter variations. An equivalent-circuit model is established in the analysis to account for the heat exchanges among the tympanum, the surroundings, and the detector as well as for the electro thermal behavior of the detector. The model provides effective simulation of the thermometer with PSPICE. Critical parameters governing the accuracy and the limitation of the tympanic thermometer are also pointed out by the simulation.

2 Data staleness

Consider some applications the data values and the recorded information is always not stable, it is always stale.one example is vehicle tracking system. Where it can measure the movements of vehicles [6] , from which the uncertain data may arises. A typical uncertainty model requires knowledge about the moving speed of the device and whether its movement is restricted or unrestricted. A 2D probability density function is defined over a bounded region to model such uncertainty.

3 Repeated measurements

Perhaps the most common source of uncertainty comes from repeated measurements. For example, a patient's body temperature could be taken multiple times during a day; an anemometer could record wind speed once every minute; the space shuttle has a large number of heat sensors installed all over its surface. When we inquire about a patient's temperature, or wind speed, or the temperature of a certain section of the shuttle, which values shall we use? Or, would it be better to utilize all the information by considering the distribution given by the collected data values?

As a more elaborate example, consider the "Breast Cancer" dataset reported in [7]. This dataset contains a number of tuples. Each tuple corresponds to a microscopic image of stained cell nuclei. A typical image contains 10–40 nuclei. One of the features extracted from each image is the average radius of nuclei. We remark that such a radius measure contains a few sources of uncertainty: (1) an average is taken from a large number of nuclei from an image, (2) the radius of an (irregularly-shaped) nucleus is obtained by averaging the length of the radial line segments defined by the centroid of the nucleus and a large number of sample points on the nucleus' perimeter, and (3) a nucleus' perimeter was outlined by a user over a fuzzy 2D image. From (1) and (2), we see that a radius is computed from a large number of measurements with a wide range of values. The source data points thus form interesting distributions. From (3), the fuzziness of the 2D image can be modelled by allowing a radius measure be represented by a range instead of a concrete point-value.

Yet another source of uncertainty comes from the limitation of the data collection process. For example, a survey may ask a question like, "How many hours of TV do you watch each week?" A typical respondent would not reply with an exact precise answer. Rather, a range (e.g., "6–8 hours") is usually replied, possibly because the respondent is not so sure about the answer himself. In this example, the survey can restrict an answer to fall into a few pre-set categories (such as "2–4 hours", "4–7 hours", etc.). However, this restriction unnecessarily limits the respondents' choices and adds noise to the data. Also, for preserving privacy, sometimes point data values are transformed to ranges on purpose before publication.

From the above examples, we see that in many applications, information cannot be ideally represented by point data. More often, a value is best captured by a range possibly with a pdf.

1) A basic algorithm for constructing decision trees out of uncertain datasets.
2) A study comparing the classification accuracy achieved by the Averaging approach and the Distribution-based approach.
3) A set of mathematical theorems that allow significant pruning of the large search space of the best split point determination during tree construction.
4) Efficient algorithms that employ pruning techniques derived from the theorems.
5) A performance analysis on the various algorithms through a set of experiments.

IV TRADITIONAL DECISION TREES

In our model, a dataset consists of d training tuples, {t1,t2,….td}, and k numerical feature attributes,A1…. Ak. The domain of attribute Aj is dom(Aj).Each tuple ti is associated with a feature vector Vi = (vi.1, vi.2,…….vi.k) and a class label ci, where vi,j εdom(Aj) and ci εC, the set of all class labels. The classification problem is to construct a model M that maps each feature vector (vx,1….. vx,k) to a probability distribution Px on C such that given a test tuple t0 = (v0,1……. v0,k, c0), P0 =M(v0,1….. v0,k) predicts the class label c0 with high accuracy. We say that P0 predicts c0 if

c0 = arg maxc C

P0(c).

In this paper we study binary decision trees with tests on numerical attributes. Each internal node n of a decision tree is associated with an attribute A jn and a split point znεdom(A jn), giving a binary test v0,jn zn. An internal node has exactly 2 children, which are labeled "left" and "right", respectively. Each leaf node m in the decision tree is associated with a discrete probability distribution Pm over C. For each c ε C, Pm(c) gives a probability reflecting how likely a tuple assigned to leaf node m would have a class label of c.

To determine the class label of a given test tuple t0 =(v0,1…..
v0,k;,?), we traverse the tree starting from the root node until a leaf node is reached. When we visit an internal node n, we execute the test v0;jn _ zn and proceed to the left child or the right child accordingly. Eventually, we reach a leaf node m. The probability distribution Pm associated with m gives the probabilities that t0 belongs to each class label c εC. For a single result, we return the class label c εC that maximizes Pm(c).

1 Handling Information

Under our uncertainty model, a feature value is represented not by a single value, vi,j , but by a pdf, fi;j . For practical reasons, we assume that fi;j is non-zero only within a bounded interval [ai;j ; bi;j]. (We will briefly discuss how our method scan be extended to handle pdf's with unbounded domains in Section VII-C.) A pdf fi;j could be programmed analytically i fit can be specified in closed form. More typically, it would be implemented numerically by storing a set of s sample points x ε[ai;j ,bi;j]with the associated value fi;j (x), effectively approximating fi;j by a discrete distribution with s possible values. We adopt this numerical approach for the rest this paper. With this representation, the amount of information available is exploded by a factor of s. hopefully; the richer information allows us to build a better classification model. On the down side, processing large numbers of sample points is much more costly. In this paper we show that accuracy can be improved by considering uncertainty information. We also propose pruning strategies that can greatly reduce the computational effort.

For each internal node n (including the root node), to
determine Φn(c; tx,wx), we first check the attribute Ajn and split point zn of node n. Since the pdf of tx under attribute Ajn spans the interval [ax;jn,bx;jn], we compute the "left"
probability pL=$\int_{ax,jn}^{zn} fx$,jn(t) dt (or pL = 0 in case zn < ax;jn) and the "right" probability pR= 1 - pL. Then, we split tx into 2 fractional tuples tL and tR. Tuples tL and tR inherit the class label of tx as well as the pdf's of tx for all attributes except Ajn.The tuple tL is assigned a weight of wL = wx . pL and its pdf for Ajn is given by

FL,jn(x) = { fx,jn(x)/wL if x ε[ax,jn,zn]
0 otherwise

the tuple tR is assigned a weight and pdf analogously. We define Φn(C;tj,wx)=PL.ΦnL(c;tL,wL)+pR.ΦnR(C;tR.wR) where nL and nR are the left child and the right child of node respectively

For every leaf node m, recall that it is associated with aprobability distribution Pm over C. We define Φm(c; tx;wx) =wx _Pm(c). Finally, for each class c, let P(c) = Φr(c; t0, 1.0),where r is the root node of the decision tree. Obtained this way, each probability P(c) indicates how likely it is that the
test tuple t0 has class label c. These computations are
illustrated in Figure 1, which shows a test tuple t0 with one feature whose pdf has the domain [-2:5, 2]. It has a weight of
1.0 and is first tested against the root node of the decision tree.
Based on the split point -1, we find that pL = 0.3 and pR = 0.7. So, t0 is split into two tuples tL and tR with weights wL = 0.3
and wR = 0.7. The tuple tL inherits the pdf fromt0over the sub
domain [-2.5,-1], normalised by multiplying by a factor of
1=wL. Tuple tR inherits the pdf from t0 in a= similar fashion. These tuples are then recursively tested down the tree until the leaf nodes are reached. The weight distributed in such a way down to each leaf node is then multiplied with the probability of each class label at that leaf node. These are finally summed up to give the probability distribution (over the class labels) for t0, giving P(A) = 0:59; P(B) = 0:41.

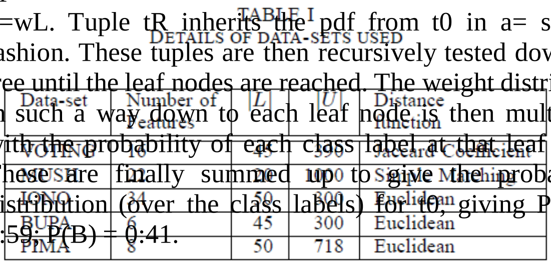


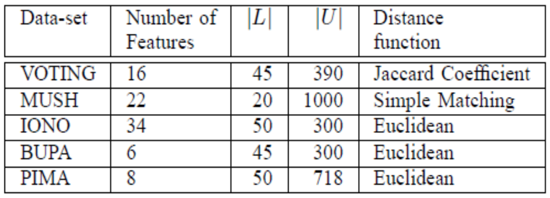
TABLE I
DETAILS OF DATA-SETS USED

Data-set	Number of Features	$\lvert L \rvert$	$\lvert U \rvert$	Distance function
VOTING	16	45	390	Jaccard Coefficient
MUSH	22	20	1000	Simple Matching
IONO	34	50	300	Euclidean
BUPA	6	45	300	Euclidean
PIMA	8	50	718	Euclidean

V. EXPERIMENTAL RESULTS

Experiments are conducted with five standard data- sets which are drawn from from the data-sets available at UCI Machine Learning Repository [13]. Properties of the datasets along with distance function used are shown in Table I.

TABLE I
DETAILS OF DATA-SETS USED



Data-set	Number of Features	$\lvert L \rvert$	$\lvert U \rvert$	Distance function
VOTING	16	45	390	Jaccard Coefficient
MUSH	22	20	1000	Simple Matching
IONO	34	50	300	Euclidean
BUPA	6	45	300	Euclidean
PIMA	8	50	718	Euclidean

Note, same data-sets are used in [7] and [8]. The classifiers used for the comparison purpose are, (i) graph mincut-±opt [7] (a transductive classifier), (ii) randomized graph mincut [8] (a transductive classifier), (iii) spectral graph partitioning [9] (a transductive classifier), (iv) ID3 (a decision tree based classifier, an inductive classifier) [14][15], (v) S-TDT (the proposed method of this paper, a transductive classifier). Classifiers for comparison are chosen so as to compare with other transductive methods which are similar to the proposed method of the paper. Two well known induction based classifiers viz., ID3 and 3-NNC are also used for the comparison purpose.
It can be seen that the proposed S-TDT method is comparable with other classifiers and in some cases shows

TABLE I DETAILS OF DATA-SETS USED

Data-set	Mincut	Rand. Graph mincut	Spectral Graph parti.	ID3	ST-DT
VOTING	91.3	91.2	85.9	86.4	91.4
MUSH	97.7	94.2	91.6	93.3	91.5
IONO	81.6	82.8	79.7	88.6	88.9
BUPA	59.3	63.5	61.6	55.3	54.8
PIMA	72.3	67.5	68.2	70.0	73.1

Note, same data-sets are used in [7] and [8]. The classifiers used for the comparison purpose are, (i) graph mincut-±opt [7] (a transductive classifier), (ii) randomized graph mincut [8] (a transductive classifier), (iii) spectral graph partitioning [9] (a transductive classifier), (iv) ID3 (a decision tree based classifier, an inductive classifier) [14][15], (v) 3- NNC (3-nearest neighbor classifier, an inductive classifier) [16], (vi) SI-TNNC (the proposed method of this paper, a transductive classifier). Classifiers for comparison are chosen so as to compare with other transductive methods which are similar to the proposed method of the paper. Two well known induction based classifiers viz., ID3 and 3-NNC are also used for the comparison purpose.

VI. CONCLUSION

For some applications, the examples for which a prediction is
needed are already known when training the classifier. We have given a method using Transduction .

REFERENCES

[1]. O. Chapelle, B. Scholkopf, and A. Zein, Semi-Supervised Learning. Cambridge, Massachusetts: The MIT Press, 2006.

[2]. D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. S. lkopf, "Learning with local and global consistency," in Advances in Neural Information.

[3]. V. Vapnik, Statistical Learning Theory. John Wiley & Sons: A Wileyinterscience Publication, New York, 1998.

[4]. V. Vapnik, Estimation of Dependences Based on Empirical Data, 2nd ed. New York: Springer Series in Statistics, Springer-Verlag, 2006.

[5]. K. Bennett, "Combining support vector and mathematical programming methods for classification," in Advances in kernel methods – support vector learning, B. Scholkopf et al., Ed. MIT-Press, 1999.

[6]. T. Joachims, "Transductive inference for text classification using support vector machines," in Sixteenth International Conference on Machine Learning. Bled Slovenia: Morgan Kaufmann, 1999, pp. 200–209.

[7]. A. Blum and S. Chawla, "Learning from labeled and unlabeled data using graph mincut," in Eighteenth International Conference on Machine Learning. Morgan Kaufmann, 2001, pp. 19–26.

[8]. A. Blum, J. Lafferty, M. Rwebangira, and R. Reddy, "Semi-supervised learning using randomized mincuts," in International Conference on Machine Learning. Morgan Kaufmann, 2004.

[9]. T. Joachims, "Transductive learning via spctral graph partitioning," in International Conference on Machine Learning, 2003, pp. 290–297.

[10]. X. Zhu, Z. Gharahmani, and J. Lafferty, "Semi-supervised learning using Gaussian fields and harmonic functions," in 20th International Conference on Machine Learning, 2003, pp. 912–919.

[11]. A. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," ACM Computing Surveys, vol. 31, no. 3, pp. 264–323, 1999.

[12]. N. Cristianini and J. Shawe-Taylor, An Introduction to Support Vector Machines and Other Kernel-based Learning Methods, 1st ed. Cambridge University Press, 2000.

[13]. P.M.Murphy, UCI Repository of Machine Learning Databases [http://www.ics.uci.edu/mlearn/MLRepository.html], Department of Information and Computer Science, University of California, Irvine, CA,

[14]. R. O. Duda, P. E.Hart, and D. G. Stork, Pattern Classification, 2nd ed. John Wiley & Sons: A Wiley-interscience Publication, 2000.

[15]. J. Han and M. Kamber, Data Mining: Concepts and Techniques, Academic Press, 2001.

[16]. B. V. Dasarathy, "Data mining tasks and methods: Classification Nearest-neighbor approaches," in Handbook of data mining and knowledge discovery. New York: Oxford University Press, 2002, pp. 288–298.

[17]. T. H. Cormen, C. E. Leiserson, and R. L. Rivest, Introduction to Algorithms. Cambridge, MA, U.S.A: The MIT Press, 1990.

[18]. R. Motwani and P. Raghavan, Randomized Algorithms. Cambridge, UK: Cambridge University Press, 1995.