Comparative study of record matching classification algorithms with Active learning

Atul Kumar, Prasanth Thomas

University of Southern California
Computer Science Department
Los Angeles, CA 90089
{atulk, prasantt}@usc.edu

Abstract

Finding duplicate rows in the database is very important in the field of data integration and data cleaning. This record linkage problem has many steps. In this paper we concentrate on the record matching step of the record linkage process. Record matching is always a tough task. The main problem is that the matching pairs of rows are very less in number, like 1% of the total number of rows. But training a classifier for better accuracy requires a lot of labeled data. There is a general practice in machine learning to use active learning when the labeled data is less. Here we try to exploit the same technique for the record matching problem. We use SVM and decision tree as classifiers. Our goal is to decide how these two classifiers perform with the help of active learning. Our active learning system first learns on initial data and then selects a new data row on every round for learning. This new data row is selected from the set of unlabeled test data as per the performance of the active learning committee classifiers. The row giving the worst performance is selected for labeling and included in the training set. We compare our active learning system with a system which randomly selects new training examples. We found that active learning is always doing better than random selection. On restaurant data set the SVM classifier works better and on the CORA dataset decision tree perform better.

Introduction

Nowadays in many applications we need to integrate data on the internet from two different sources. There might be duplicate entries in the data. This problem of matching duplicate pairs in the two data sources is called Record Linkage Problem. The record linkage problem has many steps.

- Schema alignment: In this step attributes of the two data sets are matched.
- Parsing: In this step data is parsed and tokenize.
- Blocking: In this step highly unlikely candidate record pairs are removed. This way we can reduce the size of training pairs.
- Field similarity: The field similarity between pairs is scored.
- Record Matching: In this step the feature vector obtained from the above step is sent to classifier to get the overall score for the candidate pair.

In the data set used in our experiment the attributes are already matched. So we do not need to do the Schema alignment step. We also skip the blocking step in our project.

Problem Statement

We want to see how the two classifiers, SVM and decision tree, perform under active learning. We also want to see in which database we get more performance improvement using active learning.

In our project we implemented an active learning system for record matching. Using this system we compare the performance of the two classifiers, decision tree and support vector machine, on the two datasets CORA and Restaurant. For the active learning part we used committee vote approach where there are four classifiers each of which learns on a single attribute. For the evaluation purpose we have also implemented a randomize selection system. The performance of the classifiers is compared on the basis of the F – Measure of matching pair class. The description of these systems and active learning parts are described below.

Classifiers

We use decision tree and support vector machine as classifiers. These two classifiers are already very popular in research related to record linkage.

Decision tree: Decision tree maps from features of an instances to the class of that instances. String similarity measure is used as feature. The decision tree makes decision on the features which give most information gain. We used J48 of WEKA for decision tree.

Support vector machine: String similarity measure is used as component of a feature vector, the SVM is trained on this feature vector to discriminate between duplicate pair to non duplicate pairs. Classification is based on the distance of feature vector from the separating hyper plane. We used SMO of WEKA for SVM.

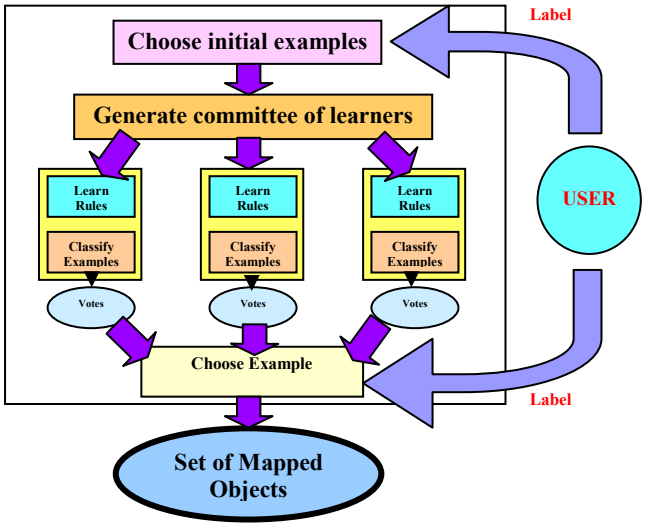
Data sets

Restaurant: It is a database of 864 restaurant names and addresses containing 112 duplicates obtained by integrating records from Fodor's and Zagat's guidebooks.
Link to this dataset is:
http://www.cs.utexas.edu/users/ml/riddle/data/restaurant.tar.gz

CORA: It is a collection of 1295 distinct citations to 122 Computer Science research papers from the Cora Computer Science research paper search engine.
Link to this dataset is:
http://www.cs.utexas.edu/users/ml/riddle/data/cora.tar.gz

Active Learning System

Our project is concentrated on the record matching part of the record linkage problem. To solve this part of the problem we build an active learning system. Below is the basic block diagram of our system. (This is taken form the class lecture slide of record linkage).



The various part of this system is described below.

Mapping the row pairs to training instances

For mapping the row pairs to training instances we take every possible pair from the dataset and map them into their string similarity measure for each field one by one. The process is explained in the figure below.

Restaurant Name	Address	City	Cuisine
Fenix	8358 Sunset Blvd. West	Hollywood	American
Fenix at the Argyle	8358 Sunset Blvd.	W. Hollywood	French (new)

↓

Restaurant Name	Address	City	Cuisine
0.910	0.954	0.992	0.09

This mapping process gives the number of instances as equal to the quadratic to the number of original rows. As an example after mapping of 864 original rows we get 864 x 864 instances. This increase in instances also increases the running time of our system.

The field similarity measure is calculated using string similarity functions. We use TFIDF for CORA dataset because the attributes in CORA database contains lots of words, so it is logical to use TFIDF measure which uses bag of word concept. For the restaurant dataset we use Levenshtein distance.

Both the training and test data is mapped using the same method.

Selecting initial training example TR

For initial set of training data we select 2 instances from the matching pair and 2 instances from the non-matching pair. This training set is represented by TR. Rest of the data is used for testing the performance of the classifiers. We represent this data set as TE.

Active Learning

For active learning we used a committee of four learners. These four learners learn on the first four features of the record in the training data. That is the first learner will learn on feature number 1, the second learner will learn on feature number 2 etc.

After learning, each of the learners classifies all the data instances in the test set. Based on the classification by the

four classifiers the information gain of each instance is calculated as per the following formula:

C = number of classifiers correctly classify the instance
I = number of classifiers incorrectly classify the instance

P = C/I

Information gain: -p * log p – (1-p) * log (1-p)

In our case if two classifiers classify correctly and two classifiers classify incorrectly, then the information gain is highest. We use the same classifier type for all the committee members as used for the overall system.

We select the instance which is having highest information gain. After this the data instance is sent to the user for labeling. The user labels the instance and includes this data instance into the training data set TR and removes it from the test dataset TE. In the next round the classifiers would learn on this updated training data. This way on each round we add one data instance in the training data. This process will go on until either a user satisfy with the error in the test data or there is no new data instance to choose for labeling form the test data set.

Randomize selection

For the evaluation of our active learning system we build another system which bypass the active learning block and select one instance from the test data randomly from the test set and put it into the training set in every round.

Evaluation

For the comparison we used F - Measure on the matched pair label. F - Measure is better than accuracy here as the number of matching pairs in the overall data set is very less (in the order of 1% of whole data). So only F - Measure can give good result.

R = fraction of duplicates correctly classified
P = fraction correct amongst all instances actually labeled duplicate

$$F = \frac{2R*P}{P + R}$$

Experimental setup

For the evaluation purpose we run the experiments with CORA and Restaurant data sets with SVM and decision tree classifiers using active learning and random selection systems. So there are 8 set of experiments need to run. Each experiment is run for 20 rounds. We run the experiments for 5 rounds using random initialization and took the average of all 5 rounds.

The X – axis shows F- Measure for duplicate classification. The Y – axis shows the number of rounds. In each round one labeled training example is added into training set. One graph each for one data set with classifiers is shown.

Comparison between Active learning vs. Random selection

Restaurant Data Set

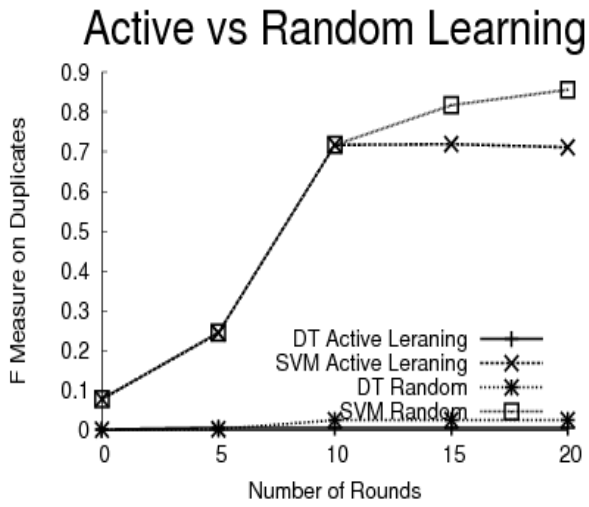


Figure 1

CORA Data Set

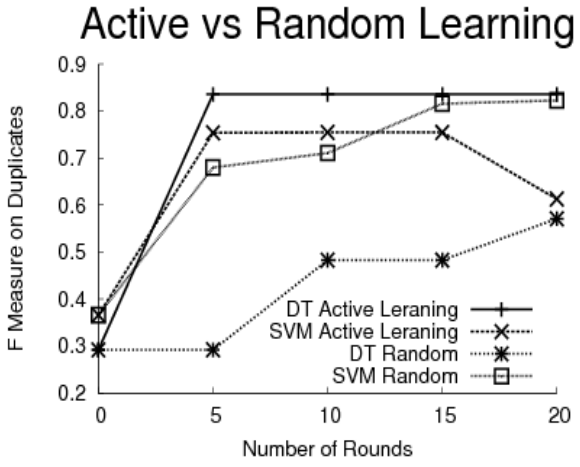


Figure 2

Comparison between SVM vs. Decision Tree using Active learning

Restaurant Data Set

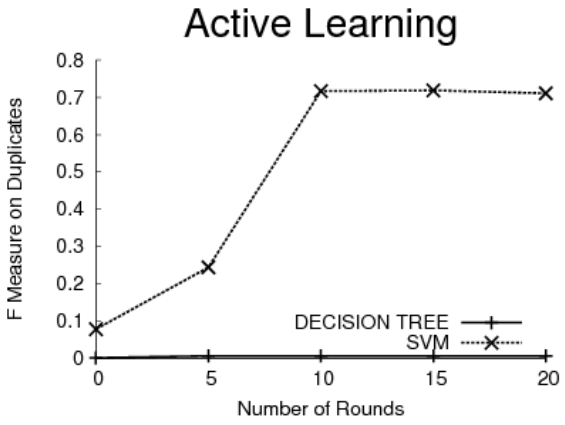


Figure 3

CORA Data Set

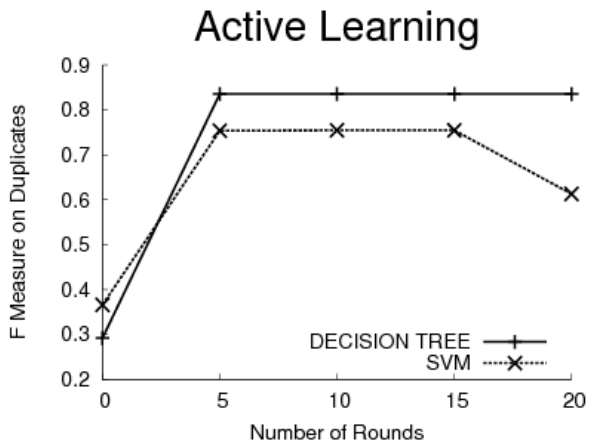


Figure 4

Comparison between SVM vs. Decision Tree using Random Selection

Restaurant Data Set

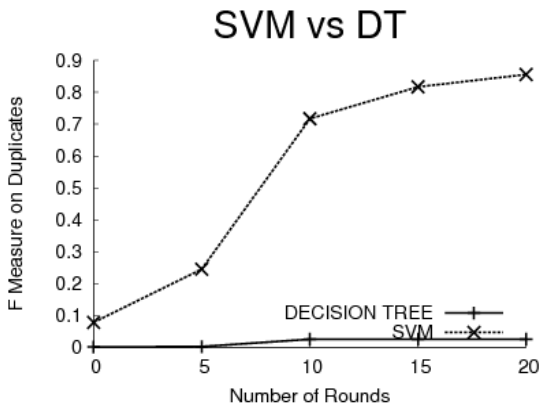


Figure 5

CORA Data Set

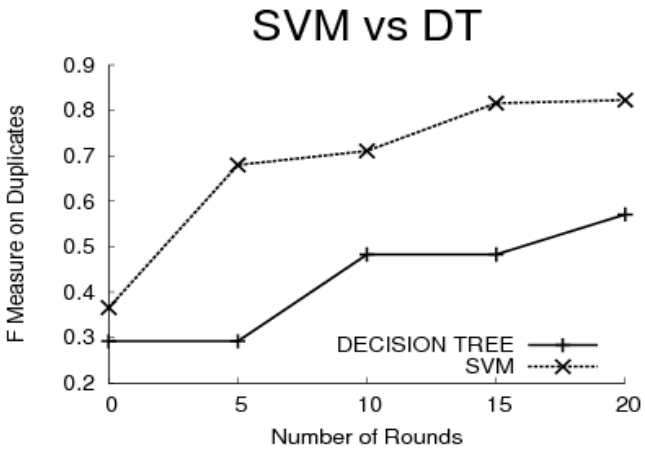


Figure 6

Conclusion

As seen from the graph in figure 5 and 6, SVM performs better than decision tree when the new instance to be added in the training set is selected using random selection.

From figure 3 and 4 we can see that with active learning decision tree perform better than SVM with CORA data set. But with restaurant data set SVM perform better than decision tree.

Figure 1 and 2 suggest that our active learning system is doing comparatively better than the random selection system in the case of both data sets. Because of the long running time we are not able to show the convergence of the result. But the initial trend suggests that the active learning system would beat the random selection.

The active learning scheme we employed here is very naïve. And still we are getting the good results. This encourages us to try different active learners in the future. We also try to use blocking scheme so that the number of data instances will reduce which will reduce the running time and we could run our experiments for more number of rounds.

Related Work

In the field of record linkage there are lots of works where people have used active learning. Because the number of matching pairs in a data set is very less, around 1%, the labeling of most of the data is very tough. This scarcity of good training data encourages people to use active learning at various phase of record linkage problem.

[**Bilenko, 2003]** describes a system in which they learn the string similarity functions for the field matching problem using active learning. They have also tried the SVM and decision tree classifiers for this purpose. In this paper at the end they have suggested a way of combining the learned field similarity measure for record matching. They have learned the weight using SVM and decision tree but they did not use active learning for record matching, active learning is used only for learning string matching function for field matching.

[**Tejada, 2002]** described a way to use active learning for learning the string transformation function. This is again done at the field similarity level and not at the record matching level.

References

Record Linkage Lecture slide from class

Sheila Tejada, Craig A. Knoblock, Steven Minton: *Learning domain-independent string transformation weights for high accuracy object identification.* KDD 2002: 350-359.

Mikhail Bilenko, and Raymond J. Mooney, 2003. *Adaptive Duplicate Detection Using Learnable String Similarity Measures.* Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2003).

Mikhail Bilenko and Raymond J. Mooney, 2003. *On Evaluation and Training-Set Construction for Duplicate Detection.* Proceedings of the KDD-2003 Workshop on Data Cleaning, Record Linkage, and Object Consolidation, pp. 7-12, Washington DC, August 2003.

William Cohen Pradeep and William W. Cohen and Pradeep Ravikumar and Stephen E. Fienberg, 2003. *A Comparison of String Metrics for Matching Names and Records.* KDD Workshop on Data Cleaning and Object Consolidation, 2003.