Social Dating: Matching and Clustering

SangHyeon (Alex) Ahn
3900 North Charles St
Baltimore, MD 21218, USA
alexahn@jhu.edu

Jin Yong Shin
3900 North Charles St
Baltimore, MD 21218, USA
jshin44@jhu.edu

Abstract

Social dating is a stage of interpersonal relationship between two individuals for with the aim of each assessing the other's suitability as a partner in a more committed intimate relationship or marriage. Today, many individuals spend a lot of money, time, and effort for the search of their true partners. Some reasons for the inefficiency in seeking sexual partners include limited pool of candidates, lack of transparency in uncovering personalities, and the nature of time consumption in building relationships. Finding the right partner in a machine driven automated process can be beneficial towards increasing efficiency in the following aspects: reduced time and effort, a larger pool, and a level of quantitative/qualitative guarantees.

A binary classification prediction model predicting a potential match between a candidate and a partner can significantly improve social dating process in terms of increasing positive match outcomes. Also, clustering candidates who have similar demographic traits and preferences can help to narrow down the pool of potential partners for a given candidate.

In this paper, we modeled binary classification predictor for a potential match for a candidate, clustered candidates into similar demographic traits and preferences, and combined the two models for prediction model. In final stages, we have acquired a model with prediction accuracy near at 0.85.

1 Introduction

Purpose of the paper was to apply machine learning algorithms for Social Dating problems and build tools to be used for smarter decision making in matching process. We used the data from Speed Dating Experiment (of 21 waves) which includes each candidate's demographic attributes and preferences. We have selected relevant features to provide the most informative data and used SQL to inner join missing feature variables.

We implemented Machine Learning algorithms (without the use of external libraries) to perform (1) binary classifications to predict a date match between a candidate and a potential partner, and (2) clustering analysis on candidates to use narrow down a pool of candidates by demographic traits and partner preferences. We have further combined the two model for acquiring a better prediction model. For our binary label, we gave 1 as a match between two candidates and -1 as a non-match between the two.

We have constructed binary classification model using multiple algorithms each with varying parameters to select the best performing (highest prediction accuracy) model by tuning following parameters:

1. I: number of training iterations
2. λ: regularization terms

Also, clustering analysis is conducted under different λ values for λ-means clustering to observe how many different clusters we may acquire from the data.

Each model's effectiveness and usefulness was also evaluated to verify suitability and validity, using accuracy testings and 5-fold cross-validation.

2 Feature Engineering

2.1 Data

Speed Dating Experiment data of 21 waves is provided by the research paper from University of Columbia conducting gender differences in mate selection, by Ray Fisman and Sheena Iyengar. The entire data has approximately 5000 instances (4-minutes speed date instance) with 150 different attributes including id, gender, match, age, race, candidate attributes, partner attributes, candidate preferences, partner preferences, interest ratings, shared interest correlations, income and so on.

For our model, we have selected only the relevant features for our train model. In addition to binary label indicating a $match$, There are five large feature categories in our selected features:

1. Candidate Demographic
2. Candidate Attributes & Preference
3. Partner Demographic
4. Partner Attributes & Preference
5. Interests & Characteristics

Demographic features include gender, age, race, field of study, income range, career field.

Attributes include candidate's own measures of attractiveness, sincerity, humor, intelligence, and ambitiousness.

Preferences include the desirable attributes (dimensions are equal as above) of a potential partner as well as level of shared interests.

Interests & Characteristics includes a correlation between two candidates' preferred interests and activities, importance of race, importance of religious background, frequency of dates, and frequency of going out.

2.2 Refining and Sampling the data

To create and collect the full data set with non-missing values and proper feature columns, we constructed a relational database in SQL and performed inner join function to compile a full database as described above.

In order to perform analysis, we duplicated the database for two scenarios, one for classification using full data, and another for clustering analysis for a single candidate.

For both database, we performed 5-fold random sampling, then divided train data to test data, at size of $8 : 2$ ratio (training:8, testing:2) for cross validation.

3 Algorithms

We developed two different algorithms for (1) classification model (2) clustering analysis.

1. **Binary classification** models explored:

   We explored three different classification models for our binary prediction model for predicting a candidate as a potential match for a person.

   (a) ***Margin Perceptron***
   Perceptron is a mistake driven online learning algorithm that performs like a single neuron. Prediction label, $\hat{y_i}$, for an instance x_i is computed as a sign value of the dot product between weight, w, and instance's feature vector.

   $$\hat{y_i} = sign(w \cdot x_i) \tag{1}$$

   In the training algorithm, w is updated whenever the classifier makes an incorrect prediction ($y \neq \hat{y}$), at each iteration in training stage. Margin Perceptron, however, updates w whenever margin (dot product value) is not satisfied at a label prediction even when the prediction is correct, to ensure labeling with at least a margin.

(b) ***SVM (Pegasos)*** Support Vector Machine is a linear classifier using max-margin principle. SVM classifies data by constructing a hyperplane in high dimensional space that segregates data, while retaining max-margin between the hyperplane to each data points. The margin is enforced by the objective function solved by QP solver (or other optimization method described below).

$$f(w) = \min_{w} \lambda \frac{1}{2}||w||^2 + \frac{1}{N}\sum_{i=1}^{N} l(w;(x_i, y_i)) \tag{2}$$

$$l(w;(x_i, y_i)) = max0, 1 - y < w, x >) \tag{3}$$

where $< \cdot, \cdot >$ refers to the inner product of two vectors. Here, bias term is omitted with the assumption that hyperplane crosses the origin.

Pegasos is a version of SVM refering to Primal Estimated sub-Gradient Solver, which takes a stochastic gradient descent to optimize the objective function stated above. Pegasos takes sub-gradient of $f(w;i)$ at iteration i, where learning rate continuously decreases at each iteration (function of time) to guarantee convergence. This algorithm adheres to online-training method with stochastic-optimization step rather than batch-optimization.

(c) ***K-Nearest Neighbors***

K-Nearest Neighbors algorithm is a classification method using labels of training instances to predict latent instance label by distances to the training instances (i.e. Euclidean distance).

K-Nearest Neighbors is an instance-based learning method which predicts a label by the labels of the nearest neighbors to the latent instance.

$$\hat{y} = \arg\max_{y'} \sum_{i \in Xn} [y_i = y'] \tag{4}$$

2. **Clustering** analysis method explored:

   Clustering is performed on all candidates who participated in the Speed Dating Experiment, to be able to group people into specific cluster which gathers people with similar demographic traits and partner preferences to each cluster.

   (a) ***λ-means***

   λ-means clustering is an unsupervised learning algorithm based on Expectation-Maximization algorithm, which is an iterative method of assigning expected cluster (by closest distance) for each instance then maximizing (expected) likelihood by updating the estimator (cluster mean) with maximum likelihood estimator at each iteration. λ-means clustering is a modified version of K-means clustering where λ is used to formulate different number of clusters without a fixed number of clusters. Whenever distance is larger than λ, the algorithm creates another cluster to assign instances.

   E-step:

$$k = \arg\min_{j} ||x_i - \mu^j|| \&\& ||x_i - \mu^j|| \tag{5}$$

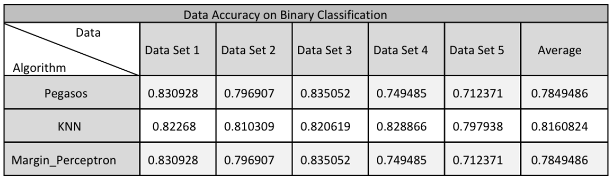
   M-step:

$$\mu^k = \frac{\sum_{i=1}^{n} r_{ik} x_i}{\sum_{i=1}^{n} r_{ik}} \tag{6}$$

   where r_{ik} is an indicator having a value of 1 is ith instance belongs to kth cluster and 0 otherwise.

4 Implementation and Method Results

In this section, we discuss our methods and approaches used for selecting which binary classification model to predict potential match between a candidate and an opposing partner. Also, we give a detailed analysis on our reasoning behind choosing parameters for both classification model and clustering analysis.



Data Accuracy on Binary Classification						
Data / Algorithm	Data Set 1	Data Set 2	Data Set 3	Data Set 4	Data Set 5	Average
Pegasos	0.830928	0.796907	0.835052	0.749485	0.712371	0.7849486
KNN	0.82268	0.810309	0.820619	0.828866	0.797938	0.8160824
Margin_Perceptron	0.830928	0.796907	0.835052	0.749485	0.712371	0.7849486

Figure 1: Binary Classification prediction accuracy using different algorithms

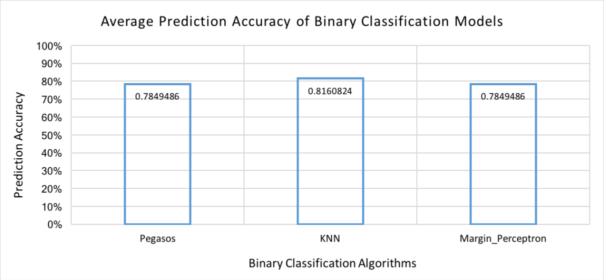


Figure 2: Binary Classification prediction accuracy using different algorithms

4.1 Binary Classification Model

For choosing binary classification model, we conducted 5-fold cross validation on 3 different algorithms: (1) Margin Perceptron, (2) SVM (Pegasos) and (3) K-Nearest Neighbors.

Figure 1 and Figure 2 shows a changing behavior of prediction values by different algorithms for binary classifications using default parameters (training iterations=10, and $\lambda = 1.0$). Notice that KNN performs best, then the other two models perform similarly.

Plot below show changes in prediction accuracy at different number of iterations.

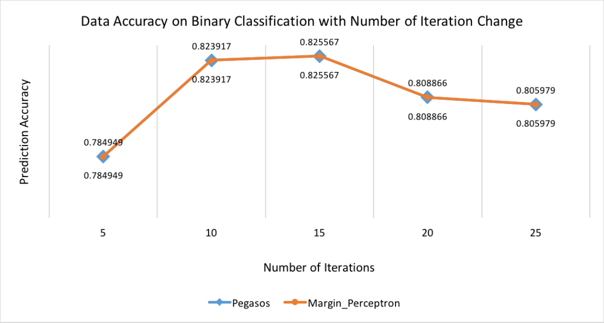


Figure 3: Binary Classification at different iterations

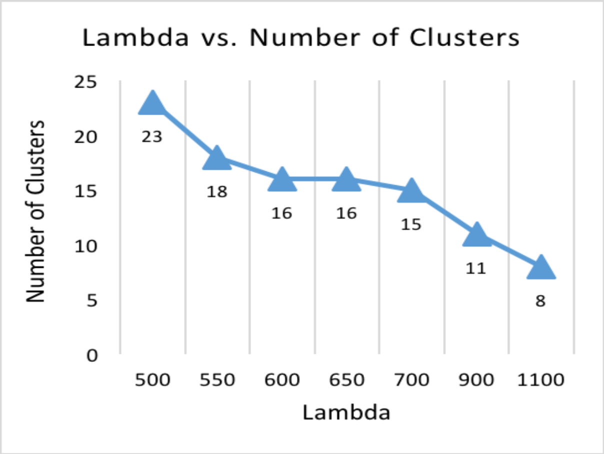


Figure 4: Number of clusters at different λ

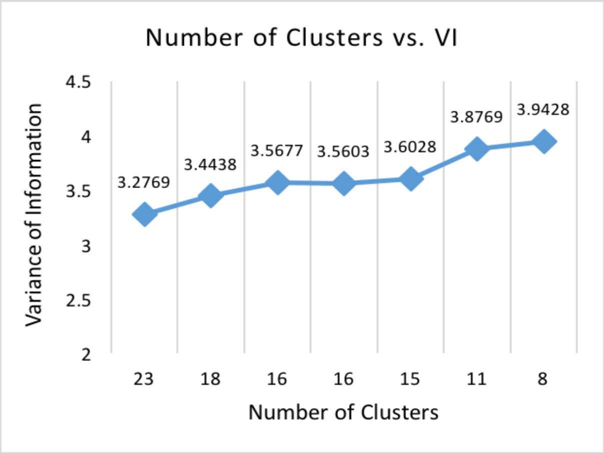


Figure 5: Variance of Information at different number of clusters

4.2 Clustering Analysis

For clustering of candidates by demographic traits and preferences, we explored various values of λ to observe data distribution behaviors.

Figure 4 shows changes in number of clusters at different λ values.

Figure 5 shows changes in Variance of Information (VI) at different cluster numbers.

Figure 6 shows changes in prediction accuracy for binary classification model using clusterID's as features, at different cluster numbers.

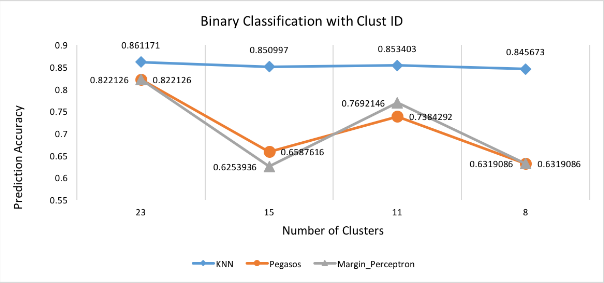


Figure 6: Variance of Information at different λ

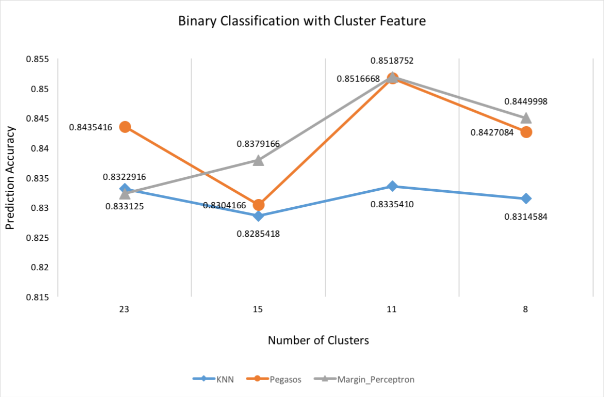


Figure 7: Combined Model prediction accuracy with feature dimension with different number of clusters

4.3 Combined Model

Lastly, we combined the two models illustrated above to increase the performance of our prediction model. We have used the cluster assignments as a additive feature dimension in the classification model using three different algorithms.

The Figure 7 shows changes in prediction accuracy using different number of clusters as an additive feature dimension.

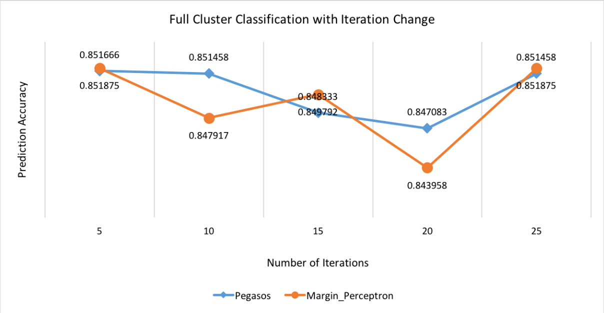


Figure 8: Combined Model prediction accuracy at different iteration numbers

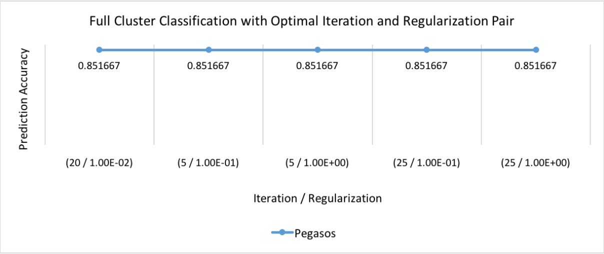


Figure 9: Combined Model prediction accuracy at different iterations and regularization constant

The Figure 8 shows changes in prediction accuracy at different iteration numbers.

The Figure 9 shows changes in prediction accuracy at different regularization constant.

5 Evaluation and Model Selection

As mentioned above, we performed 5-fold cross validation by randomly sampling the data into 5 different sets of train and test sets (8:2 ratio) to account for overfitting issues.

From the results presented above we can draw the following observations:

1. **Binary Classification Model**

   At default parameter setting, resulting prediction accuracy is the following:

   (a) KNN: 0.816
   (b) Pegasos: 0.785
   (c) Margin Perceptron: 0.785

   We observe that KNN performs at best in terms of prediction accuracy, and that other two algorithms behaves similarly. However, we also observed that KNN runs at a much slower rate compared with the other two algorithms.

2. **Clustering Analysis**

   We observe the following properties (VI and number of clusters) at different λ values:

   (a) $\lambda = 500$: 23 unique clusters / VI: 3.27
   (b) $\lambda = 700$: 15 unique clusters / VI: 3.60
   (c) $\lambda = 900$: 11 unique clusters / VI: 3.87
   (d) $\lambda = 1100$: 8 unique clusters / VI: 3.94

When using Cluster ID's as features for binary classification model we observe that KNN seems to perform best, however, this is due to the fact that there are only two features for calculating distances, and that there are a lot more labels for no match (-1) compared with match (1) that predicting as all non-match gives back good prediction accuracy.

3. **Combined Model**

   In the combined model, we observe a noticeable contribution to prediction model with additive feature on dimension over cluster ID's.

   We observe the following (best) prediction accuracy at each number of clusters, where Pegasos and Margin Perceptron outperform KNN:

   (a) 23 unique clusters : 0.844 (Pegasos) | 0.832 (Margin Perceptron)
   (b) 15 unique clusters : 0.830 (Pegasos) | 0.838 (Margin Perceptron)
   (c) 11 unique clusters : 0.852 (Pegasos) | 0.852 (Margin Perceptron)
   (d) 8 unique clusters : 0.843 (Pegasos) | 0.844 (Margin Perceptron)

   Generally speaking, both algorithms perform best when number of cluster equals 11, which is at $\lambda = 900$. In terms of algorithmic usage, Pegasos algorithm which is a version of SVM enables us to tune parameters including regularization terms while Margin Perceptron does not. Hence, Pegasos gives us a lot more flexibility which we can change the learning rate as well as regularization term to increase prediction accuracy in the future.

Therefore, from our experiment and observations, we chose Pegasos algorithm with Clustering analysis at $\lambda = 900$ that gives us feature dimension of 11 different cluster ID assignments. Optimal number of iterations used for this algorithm was at 25 iterations. Regularization factor did not seem to greatly affect the predictability. The resulting prediction accuracy is at 0.852, for correctly predicting a match between a candidate and a potential partner.

6 Conclusion

In conclusion, we have successfully modeled and chosen a prediction model using Pegasos Support Vector Machine (supervised binary classification algorithm) and λ-means clustering (unsupervised clustering algorithm), then we compiled a combined model using cluster ID's as features.

For tuning parameters we have chosen each by the results presented in section 4:

1. Pegasos: Iterations=25, Regularization=1.0

2. λ-means: $\lambda = 900$

3. Combined Model: added feature dimension on cluster assignments by λ-means clustering

Our final combined model gives us a prediction accuracy value of 0.852 which we believe is a strong predictor.

Acknowledgments

We thank professor Mark Dredze from Johns Hopkins University for supervising our project and helping us formulate the project idea using machine learning algorithm.

References

Mark Dredze. 2016. *Introduction to Machine Learning* Baltimore, MD.

Ray Fisman and Sheena Iyengar. 2005. *GENDER DIFFERENCES IN MATE SELECTION: EVIDENCE FROM A SPEED DATING EXPERIMENT** New York, NY.

Appendix

Lambda Mean Clustering

Lambda Value	Number of Clusters	VI
500	23	3.2769
550	18	3.4438
600	16	3.5677
650	16	3.5603
700	15	3.6028
900	11	3.8769
1100	8	3.9428

Cluster Binary Classification

Data Algorithm	Lambda	Data Set 1	Data Set 2	Data Set 3	Data Set 4	Data Set 5	Average
KNN	500	0.861171	0.861171	0.861171	0.861171	0.861171	0.861171
	700	0.850997	0.850997	0.850997	0.850997	0.850997	0.850997
	900	0.853403	0.853403	0.853403	0.853403	0.853403	0.853403
	1100	0.845673	0.845673	0.845673	0.845673	0.845673	0.845673
Pegasos	500	0.822126	0.822126	0.822126	0.822126	0.822126	0.822126
	700	0.604407	0.673662	0.582371	0.616999	0.816369	0.6587616
	900	0.800000	0.800000	0.646073	0.800000	0.646073	0.7384292
	1100	0.485923	0.668405	0.668405	0.668405	0.668405	0.6319086
Margin_Perceptron	500	0.822126	0.822126	0.822126	0.822126	0.822126	0.822126
	700	0.576076	0.655824	0.422875	0.655824	0.816369	0.6253936
	900	0.800000	0.800000	0.800000	0.646073	0.800000	0.7692146
	1100	0.668405	0.485923	0.668405	0.668405	0.668405	0.6319086

Full Cluster Classification

Data Algorithm	Lambda	Data Set 1	Data Set 2	Data Set 3	Data Set 4	Data Set 5	Average
KNN	500	0.827083	0.834375	0.835417	0.832292	0.836458	0.833125
	700	0.832292	0.820833	0.83125	0.832292	0.826042	0.8285418
	900	0.84375	0.820833	0.846875	0.83333	0.822917	0.8335410
	1100	0.844792	0.854167	0.828125	0.817708	0.812500	0.8314584
Pegasos	500	0.843750	0.842708	0.83125	0.841667	0.858333	0.8435416
	700	0.850000	0.802083	0.821875	0.837500	0.840625	0.8304166
	900	0.866667	0.851042	0.859375	0.847917	0.833333	0.8516668
	1100	0.848958	0.862500	0.841667	0.828125	0.832292	0.8427084
Margin_Perceptron	500	0.843750	0.842708	0.77500	0.841667	0.858333	0.8322916
	700	0.850000	0.839583	0.821875	0.837500	0.840625	0.8379166
	900	0.866667	0.851042	0.860417	0.847917	0.833333	0.8518752
	1100	0.848958	0.862500	0.848958	0.828125	0.836458	0.8449998

Full Cluster Classification Iteration Change

Data Algorithm	Number of Iterations	Data Set 1	Data Set 2	Data Set 3	Data Set 4	Data Set 5	Average
Pegasos	5	0.866667	0.851042	0.859375	0.847917	0.833330	0.851666
	10	0.866667	0.850000	0.859375	0.847917	0.833333	0.851458
	15	0.852083	0.851042	0.858333	0.847917	0.832292	0.848333
	20	0.860417	0.835417	0.858333	0.847917	0.833333	0.847083
	25	0.866667	0.851042	0.858333	0.847917	0.833333	0.851458
Margin_Perceptron	5	0.866667	0.851042	0.860417	0.847917	0.833333	0.851875
	10	0.862500	0.851042	0.858333	0.847917	0.819792	0.847917
	15	0.865625	0.851042	0.851042	0.847917	0.833333	0.849792
	20	0.866667	0.812500	0.859375	0.847917	0.833333	0.843958
	25	0.866667	0.851042	0.860417	0.847917	0.833333	0.851875

Full Cluster Classification Regularization Coefficient Change

Data Algorithm	Regularization Parameter	Data Set 1	Data Set 2	Data Set 3	Data Set 4	Data Set 5	Average
Pegasos	1.00E-04	0.866667	0.851042	0.859375	0.847917	0.833330	0.851666
	1.00E-03	0.866667	0.851042	0.859375	0.847917	0.833333	0.851667
	1.00E-02	0.866667	0.851042	0.858333	0.847917	0.832292	0.851250
	1.00E-01	0.866667	0.851042	0.859375	0.832292	0.833333	0.848542
	1.00E+00	0.866667	0.851042	0.859375	0.847917	0.833333	0.851667

Full Cluster Classification with Optimal Iteration and Regularization Pair

Data Algorithm	Iter / reg pair	Data Set 1	Data Set 2	Data Set 3	Data Set 4	Data Set 5	Average
Pegasos	(20 / 1.00E-02)	0.866667	0.851042	0.859375	0.847917	0.833333	0.851667
	(5 / 1.00E-01)	0.866667	0.851042	0.859375	0.847917	0.833333	0.851667
	(5 / 1.00E+00)	0.866667	0.851042	0.859375	0.847917	0.833333	0.851667
	(25 / 1.00E-01)	0.866667	0.851042	0.859375	0.847917	0.833333	0.851667
	(25 / 1.00E+00)	0.866667	0.851042	0.859375	0.847917	0.833333	0.851667

Data Accuracy on Binary Classification with Number of Iteration Change

Data Algorithm	Number of Iterations	Data Set 1	Data Set 2	Data Set 3	Data Set 4	Data Set 5	Average
Pegasos	5	0.830928	0.796907	0.835052	0.749485	0.712371	0.784949
	10	0.825773	0.813402	0.839175	0.822680	0.818557	0.823917
	15	0.836082	0.810309	0.836082	0.816495	0.828866	0.825567
	20	0.834021	0.817526	0.836082	0.824742	0.731959	0.808866
	25	0.810309	0.810309	0.837113	0.827835	0.744330	0.805979
Margin_Perceptron	5	0.830928	0.796907	0.835052	0.749485	0.712371	0.784949
	10	0.825773	0.813402	0.839175	0.822680	0.818557	0.823917
	15	0.836082	0.810309	0.836082	0.816495	0.828866	0.825567
	20	0.834021	0.817526	0.836082	0.824742	0.731959	0.808866
	25	0.810309	0.810309	0.837113	0.827835	0.744330	0.805979