Predict the rising stars in research

Vamsi Chunduri, Vamsi Simma Krishna

May 8, 2018

Abstract

Online bibliographic databases are powerful resources for research in data mining and social network analysis especially co-author networks. Predicting future rising stars is to find brilliant scholars/researchers in co-author networks. In this paper, we propose a solution for rising star prediction by applying machine learning techniques. For classification task, discriminative and generative modeling techniques are considered and two algorithms are chosen for each category. The author, co-authorship and venue based information are incorporated, resulting in eleven features with their mathematical formulations. Extensive experiments are performed to analyze the impact of individual feature, category wise and their combination w.r.t classification accuracy. Then, two ranking lists for top 30 scholars are presented from predicted rising stars. In addition, this concept is demonstrated for prediction of rising stars in database domain. Data from Altmetrics databases are used for algorithms' experimental analysis.

1 Introduction

Finding rising stars is a challenging and interesting task which is being investigated recently in co-author networks. Rising stars are authors who have a low research profile in the start of their career but may become experts in the future. Predicting the fast-rising young researchers (Academic Rising Stars) in the future provides useful guidance to the research community, e.g., offering competitive candidates to university for young faculty hiring as they are expected to have success academic careers. Finding such Rising starts within the organizational domains is the great need of current era, so that the organizations can put efforts to maximize the expertise of rising stars in order to get the optimal performance in future. In this work we would like to predict the stars for given a set of young researchers who have published the first first-author paper recently, we solve the problem of how to effectively predict the top k researchers who achieve the highest citation increment in Δt years.

Now-a-days, many online databases such as Altmetrices store large numbers of scientific publications. These databanks provide useful information such as the author, venue, publication's title, year, altmetrics scores, readers of various outlets, blogs, demographics. Additional features such as co-authorship information, co-citation relations in research community may be exploited to facilitate further novel services for online databases.

The academic social networks are typically based on co-authorship and co-citation relationship among researchers and publications. These networks usually have more stable and less dynamic structures as compared to other networks such as social tagging (Tang et al. 2008). There are online services that process the stored information such as Altmetrics and Microsoft Academic Search.1 Some portray co-author relationships in a star topology such as Social graph and Instant graph search.2 However there is a little work done for differentiating different authors and modeling the evolution of author's research profile based over time and progress.

2 Related Work

As already mentioned, there is little work done for predicting rising stars. So initially we discuss types (discriminative and generative) of machine learning approaches and their applications in social networks.

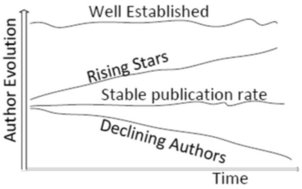


Figure 1: Four basic author's evolution behavior over time.

2.1 Application of generative and discriminative classification technique

"Generative" is a model that formulates joint probability distribution over instances and label sequences. Two algorithms for this model are considered here for classification that is Bayes Network (BN) and Naive Bayes (NB). BN is already applied for anomaly detection (Mascaro et al. 2014), trust building for electronic markets and communities , modeling for consensus between expert finding (Lopez-Cruz et al. 2014), simple and complex emotions topic analysis and context adaptive user interface (Song and Cho 2013). Similar to BN, NB embedded the concept of independence with Bayesian theorem. It employs the concept of conditional probability and successfully implemented for feature subset selection (Bermejo et al. 2014). It is also combined with decision tree for multi-class classification task (Farid et al. 2014). For disambiguation on the affiliations of authors, it was successfully applied (Cuxac et al. 2013). However both (NB and BN) were never applied for rising star classification.

"Discriminative" is based on modeling the dependence of unknown variable on known variable or data. Maximum entropy markov model (MEMM) and CART are used in this research for classification of rising star. Previously, MEMM was successfully implemented for dynamic process monitoring and diagnosis (Li et al.2014), for noise robust speech recognition (Cui et al. 2013) and for Blind separation of non-stationary sources (Gu et al. 2013). CART is a decision tree structure and it is continuously being improved. Examples are multi-labels image annotation (Fakhari and Moghadam 2013) and behavior and credit scoring (Kao et al. 2013). However both (MEMM and CART) were never applied for rising star classification.

2.2 Methods

In this work, two types of classification models are considered to learn the desired predictive function FRS (.) and two algorithms are chosen for each model category. In the next section, these algorithms' implementation is examined and results are critically analyzed. The mathematical formulation for these methods is presented next.

Discriminative methods

Support vector machines (SVMs)

Support vector machines (SVMs) are a type of learning model used for classification and regression analysis. In an SVM data points are represented as points in space in such a way that points from different categories are separated by a plane. You can think of this like a line through data points that separates data of different classes.

One-class SVMs are a special case of support vector machine. First, data is modelled, and the algorithm is trained. Then when new data are encountered their position relative to the "normal" data (or inliers) from training can be used to determine whether it is "out of class" or not - in other words, whether it is unusual or not. Because they can be trained with unlabeled data they are an example of unsupervised machine learning.

The SVM algorithm is implemented in practice using a kernel.

The learning of the hyperplane in linear SVM is done by transforming the problem using some linear algebra, which is out of the scope of this introduction to SVM.

A powerful insight is that the linear SVM can be rephrased using the inner product of any two given observations, rather than the observations themselves. The inner product between two vectors is the sum of the multiplication of each pair of input values.

For example, the inner product of the vectors [2, 3] and [5, 6] is 2*5 + 3*6 or 28.

The equation for making a prediction for a new input using the dot product between the input (x) and each support vector (xi) is calculated as follows:

$$f(x) = B0 + sum(ai * (x, xi))$$

This is an equation that involves calculating the inner products of a new input vector (x) with all support vectors in training data. The coefficients B0 and ai (for each input) must be estimated from the training data by the learning algorithm.

Classification and regression tree (CART)

CART is basically a non parametric learning approach that results in either regression or classification tree depending variables (features) are either categorical or numeric (Chrysos et al. 2013; Speybroeck 2012). The method of CART contains three steps (Loh 2011).

1.Construction of maximum tree.
2.Selection of right tree size.
3.Classify new data using already constructed tree.

In a simple form, our aim is to predict a response or class y from input vector (X1,...,Xm). A binary tree is then constructed and a test is performed on each internal node to create a left or right sub branch of tree. This process is repeated until leaf node is constructed. CART solves the following maximization problem at each node.

$$Argmax = I(tp) - PL * I(t1) - Pr(tr)$$

where I (tp) is an impurity function, tp is parent node; tl and tr are left and right child nodes. PL and PR are probabilities of left and right child nodes. The gini/towing splitting rules are used for impurity calculation at each node.

Generative methods

Naives Bayes (NB)

The NB is a probabilistic classification method that applies naive hypothesis with Bayes algorithm for every pair of features. It can handle both continuous and categorical independent variables and assumes that features are statistically independent (Ma et al. 2013). Given a class label y and a feature vector X = x1, x2,..., xm, the Bayes theorem is described as

$$P(y|X1, X2, X3, ..., Xm) * P(X1, X2, X3, ..., Xm|y)P(y)$$

However it assumes that conditional probabilities for independent variables are statistically independent (Chen et al. 2009). Therefore we simplifies the expression into product form as

2.3 Work Contributions

1.The first attempt to consider author's citations, co- author's order of appearance and author's influence and weight.
2. Mathematical formulations for the computation of weighted mutual weight and influence.
3. Performance evaluation of proposed and baseline methods in terms of average number of papers and average number of citations.
4. Qualitative analysis of top ranked 10 authors in terms of their achievements.

2.4 Data

To evaluate the performance of applied classifiers, the dataset is build by filtering authors' information from the alt-metrics dataset provided to us in the big data course at NIU. This dataset consists of articles and citations, which consists of features including; alt-metrics scores, readers of various outlets, blogs, demographics, and where it has been cited. Consisting of 9 million JSON files in the dataset. Each JSON file represented a tuple in the Alt-metric data, which we processed to create the final dataset. We extracted 50,000 tuples using random and after cleaning 33,125 tuples were obtained.

2.5 Data feature extraction

The data we used for the project is the Altmetric Dataset.Altmetrics include a wide variety of counts, from coverage in mainstream news and social media (tweets,shares, likes, etc), author Details, score, blogs , readers and demographics to citations on Wikipedia.

The dataset consists of 9 million json files with each json file representing a tuple in the data. We used python code to extract the required attributes from the altmetric data. The data extraction is a lengthy process as we have to loop through 9 million entries to extract the desired attributes. About 50,000 tuples of required attributes were selected randomly using the python code. Attributes such as AltmetricID ,AltmetricRank, Score ,DOI, ISSN ,Authors, Journal, Pubdate, Citelike ,Mendeley, and Policy were extracted from the Altmetric dataset. Using these features the desired features to perform Machine Learning Techniques to classify the data are calculated.

All values 'NaN' values are replaced with zeroes.

The data obtained is continous data which is suitabe for regression .But the data is converted to categorical values by normalizing the features based on minimum and maximum values so the normalized data rests between 0 and 1, to which classification can be done . The resulting dataframe is saved to a .csv file .Any repeating values do to any bugs in the code while extractig are carefully checked using pandas and removed.

2.6 Construction of feature space

In this section, feature set is formed based on contents and graph information. In the next section, four classifiers are trained using this feature information. Then these trained classifiers are evaluated for unseen data. The feature information's are categorized in three types as shown in Table 1. The mathematical formulation and brief description of each feature is also defined here.
The features used are:
1. No of author publications.
2. Author publications 5-10
3. Author publications 10-15
4. Mutual weight
5. Author Influence
6. Simpson Diversity
7. Co-author citations

2.6.1 Author Influence

If a junior researcher is able to work together with an expert or capable to perform numerous contributions in team work then he/she has bright chances to be a future expert. Consider two authors g and h with 4 and 3 publications respectively. Both are coauthors in 2 publications, the influence of an author to other author can be calculated as

$$Influence(Pg, Ph) = (ah, ag)/Pag = 2/4$$

$$Influence(Ph, Pg) = (ag, ah)/Pah = 2/3$$

where Pag and Pah are total number of publications for authors ag and ah. Hence author ah influences to ag with 0.5 score and ag to ah with 0.66 score.

2.6.2 Simpson Diversity

Simpson's Diversity Index is a measure of diversity which takes into account the number of species present, as well as the relative abundance of each journal. As journal richness and evenness increase, so diversity increases.

$$D = 1 - \frac{\sum n(n-1)}{\sum N(N-1)}$$

The value of D ranges between 0 and 1. With this index, 1 represents infinite diversity and 0, no diversity.

2.6.3 Co-Author citations

If a junior researcher has initially few citations collaborates with senior scholar, then there are more chances for junior scholar to get more citations in collaboration. The co-author citations can be computed as

$$CC(ag) = \sum ab + \sum ac + \sum ad$$

where CC (ag) is the co-author citations for author ag, and (Rab , Rac , Rad) is the sum of total papers' citations of the co-authors (ab, ac and ad) of author ag.

Performance Evaluation

In this work, the performance of applied classifiers is analyzed by Precision, Recall and F1 evaluation metrics. We mainly used F1 score to examine the effects of different features for rising star classification accuracy and prediction. The mathematical definition for these metrics are described as

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

2.7 Ranking of predicted rising stars

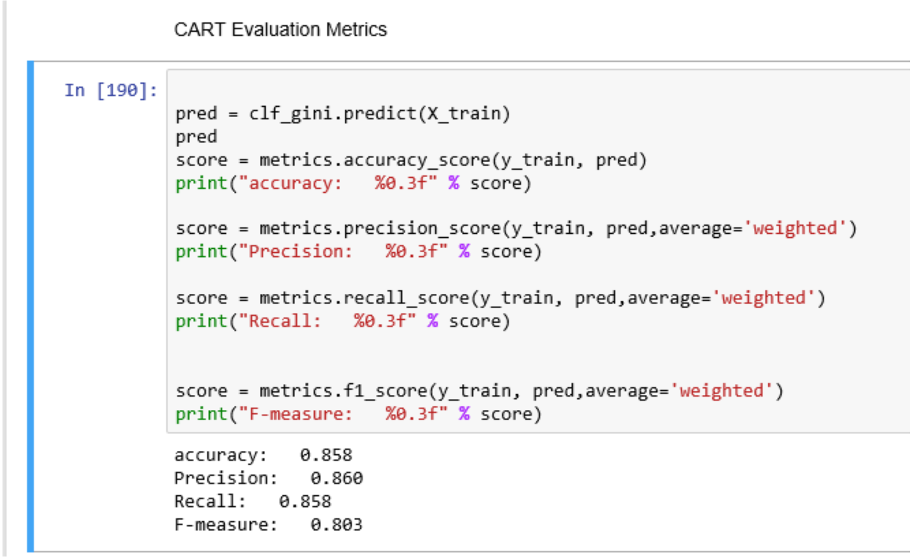
In this section, ranking of top thirty rising stars are presented for the list of predicted future rising stars. The authors are ranked by sorting score in decreasing order. The formulation of rising star score is derived in two steps; first values of all features are normalized in the range of 0–1, For all other feature values higher is better so they are added. The mathematical formulation is given by

$$RisingStarScore = Noofauthorpublications + Authorpublications5-10 + Authorpublications10-15 + Mutualweight-$$

2.8 Results

2.9 Conclusion

In this paper, discriminative and generative machine learning techniques are used for prediction of rising stars in research. Three classes of features are explored i.e. Author, Co-author. SVM, CART and Naive Bayes are chosen for experiment and results analysis. Two types of data sets are made using total citations and average relative increase in citations as measures. It is found that SVM performs better as compared to other models for total citations and CART performs better as compared to other models for average mutual weights. At the end, the application of this concept to database domain is also found functional.



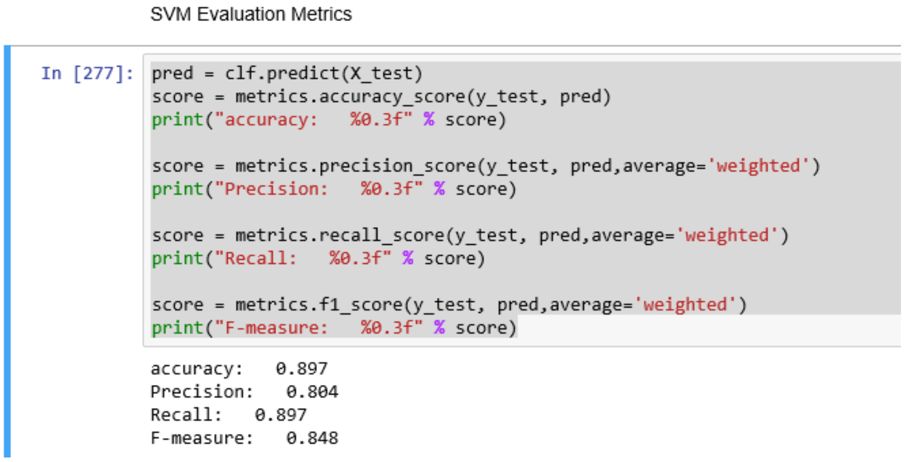
CART Evaluation Metrics

In [190]:

```
pred = clf_gini.predict(X_train)
pred
score = metrics.accuracy_score(y_train, pred)
print("accuracy:   %0.3f" % score)

score = metrics.precision_score(y_train, pred,average='weighted')
print("Precision:   %0.3f" % score)

score = metrics.recall_score(y_train, pred,average='weighted')
print("Recall:   %0.3f" % score)


score = metrics.f1_score(y_train, pred,average='weighted')
print("F-measure:   %0.3f" % score)
```

```
accuracy:   0.858
Precision:   0.860
Recall:   0.858
F-measure:   0.803
```

Figure 2: CART Evaluation



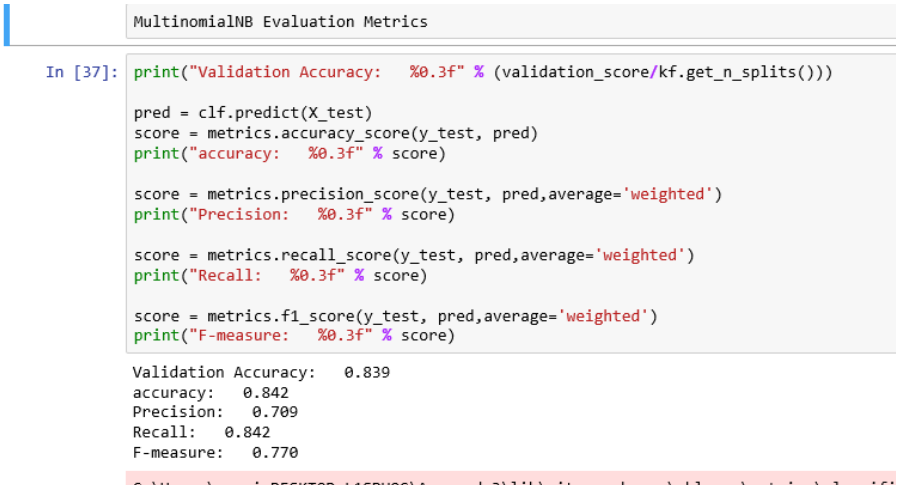
SVM Evaluation Metrics

In [277]:

```
pred = clf.predict(X_test)
score = metrics.accuracy_score(y_test, pred)
print("accuracy:   %0.3f" % score)

score = metrics.precision_score(y_test, pred,average='weighted')
print("Precision:   %0.3f" % score)

score = metrics.recall_score(y_test, pred,average='weighted')
print("Recall:   %0.3f" % score)

score = metrics.f1_score(y_test, pred,average='weighted')
print("F-measure:   %0.3f" % score)
```

```
accuracy:   0.897
Precision:   0.804
Recall:   0.897
F-measure:   0.848
```

Figure 3: SVM evaluation



MultinomialNB Evaluation Metrics

In [37]:

```
print("Validation Accuracy:   %0.3f" % (validation_score/kf.get_n_splits()))

pred = clf.predict(X_test)
score = metrics.accuracy_score(y_test, pred)
print("accuracy:   %0.3f" % score)

score = metrics.precision_score(y_test, pred,average='weighted')
print("Precision:   %0.3f" % score)

score = metrics.recall_score(y_test, pred,average='weighted')
print("Recall:   %0.3f" % score)

score = metrics.f1_score(y_test, pred,average='weighted')
print("F-measure:   %0.3f" % score)
```

```
Validation Accuracy:   0.839
accuracy:   0.842
Precision:   0.709
Recall:   0.842
F-measure:   0.770
```

Figure 4: Multinomial NB

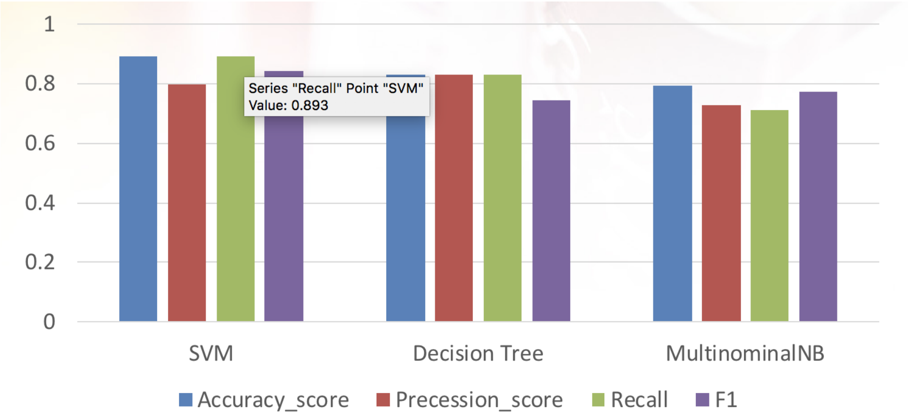


Figure 5: Comparsion between methods

References

[1]Bermejo, P., Gamez, J. A., Puerta, J. M. (2014). Speeding up incremental wrapper feature subset selection with Naive Bayes classifier. Knowledge-Based Systems, 55, 140–147.

[2]Chen, J., Huang, H., Tian, S., Qu, Y. (2009). Feature selection for text classification with Naive Bayes. Expert Systems with Applications, 36(3), 5432–5435.

[3]Chrysos, G., Dagritzikos, P., Papaefstathiou, I., Dollas, A. (2013). HC-CART: A parallel system implementation of data mining classification and regression tree (CART) algorithm on a multi-FPGA system. ACM Transactions on Architecture and Code Optimization, 9(4), 47.

[4]Constantinou, A. C., Fenton, N. E., Neil, M. (2012). pi-football: A Bayesian network model for fore- casting Association Football match outcomes. Knowledge-Based Systems, 36, 322–339.

[5]Cui, X., Afify, M., Gao, Y., Zhou, B. (2013). Stereo hidden Markov modeling for noise robust speech recognition. Computer Speech and Language, 27(2), 407–419.

[6]Cuxac, P., Lamirel, J.-C., Bonvallot, V. (2013). Efficient supervised and semi-supervised approaches for affiliations [7]disambiguation. Scientometrics, 97(1), 47–58.

[8]Daud, A., Abbasi, R., Muhammad, F. (2013). Finding rising stars in social networks. Database Systems for Advanced Applications (LNCS), 7825, 13–24.

[9]Daud, A., Li, J., Zhou, L., Muhammad, F. (2010). Temporal expert finding through generalized time topic modeling. Knowledge-Based Systems (KBS), 23(6), 615–625.