RC24857 (W0909-022) September 8, 2009
Computer Science

IBM Research Report

Content-based Link Prediction for Patent Marketing

Claudia Perlich, Grzegorz Swirszcz, Rick Lawrence
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Cambridge - Haifa - India - T. J. Watson - Tokyo - Zurich

Content-based Link Prediction for Patent Marketing

Claudia Perlich, Grzegorz Świrszcz, Rick Lawrence
IBM T.J. Watson Research Center
Yorktown Heights, NY 10598, USA
{perlich,swirszcz,ricklawr}@us.ibm.com

ABSTRACT

Patent licensing is a significant source of revenue for a business with a patent portfolio as large as IBM's. Successful marketing of such a portfolio requires methodology to match the technology covered by each patent to the requirements of other companies which are potential licensees for this intellectual property. In this paper, we address this problem using a purely content-based recommendation methodology to identify new opportunities by matching companies and patents for which no prior linkages exists. In this context we highlight the important concept of learning 'equality' and explore why existing content-based approaches typically take an indirect approach of pre-defined similarities. We show theoretically and empirically that this equality concept is easily addressed by the standard feature-vector representation and second order polynomial kernel SVMs.

1. INTRODUCTION AND TASK

IBM has a portfolio of approximately 40K issued patents, and derives significant revenue from the licensing and assignment of a subset of patents in this portfolio. The efficient management of this intellectual property (IP) requires decisions at key points in the IP business process. Decisions need to be made as the whether to pay the fees necessary to maintain each patent in force. The broader challenge is identifying new opportunities to license a patent (or cluster of patents) to companies that would be likely to benefit from the patented technologies. In addition to licensing a patent via payment of fees for a specified time period, companies can also acquire the patent rights via reassignment from IBM. Both business models require the capability to identify candidate companies. Identifying such business opportunities is at the heart of our recommendation task. We are interested in three specific objectives: (1) identifying potential companies for a given patent, (2) identifying suitable patents for a particular company and (3) making aggregate decisions based on the value of a patent. The first two objectives involve link prediction [7, 4, 1] where we are interested in associating patents with companies.

The task of identifying patent opportunities has so far been performed manually by a group of domain experts who explored a number of possible matches. While fairly successful, this human-intensive approach does not scale to cover the extensive patent portfolio and the large potential customer base. We have access to a manually created database of matches. The goal of our work is to extend this existing set of links and in particular to identify new business opportunities. This implies that the candidate companies currently have NO existing association of patents. The same holds for a large body of our patents for which we have not yet identified potential customers.

Contrary to most existing work on recommendation, and in particular collaborative filtering (see for instance [2, 10]), we cannot take any advantage of the graph topology for the purpose of link prediction and ultimately for recommendation. Instead we have to focus exclusively on the information that is available at the nodes for patents and companies. This makes our task a purely content-based recommendation where we can use the existing links as learning examples in a supervision classification setting. Our presented approach is on the surface somewhat related to existing content-based recommendation approach but differs in more than one fundamental point. The majority of content-based recommendation approaches still require linkage to be present in the test cases, i.e. "Content-based methods make recommendations by analyzing the description of the items that have been rated by the user and the description of items to be recommended" [9].

In addition to the usage of linkage, another interesting observation in previous work is that the content is typically not presented directly to one classifier. Instead either (1) some similarity metric is defined ex-ante and pre-calculated on the content (see for instance [4]) or (2) one fairly simple model is built [9, 11] for each node (such as a user) to capture a node-specific function of user preference. We suggest that all these somewhat indirect approaches reflect intuitive solutions to one inherent property of recommendation concepts: similarity and equality. We explore in the next section the inherent obstacle that seems to have prevented the widespread usage of a simple feature-vector representation.

2. CONTENT-BASED CLASSIFICATION

Let's take a closer look at the task and potential solutions. We are trying to predict the existence of a link based on the features of the nodes it is connecting. It is fairly straight-

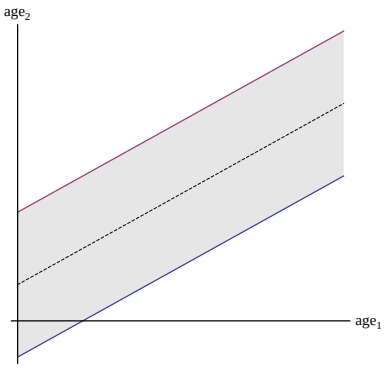

Figure 1: Concept of partners prefering similar age.

forward to construct a feature vector to capture the information available at the nodes. Since we want to predict the link between two nodes, we can concatenate the two feature vectors into one long vector and simply add the label from the training data whether or not a link existed between the two entities.

Although it ultimately involves link prediction, the question we are asking is whether this problem is more appropriately viewed as conventional supervised learning involving the direct classification of a single object? In particular, would we expect that the usual approaches (say feature selection and starting by looking at main effects and simple linear models) are as effective in this setting? We argue that different forces are at work in the case of link-based classification. They invalidate some of the usual methodology and direct us towards a specific type of model as first choice for content-based link prediction.

Consider the example of link prediction for one of many matchmaking sites. Let us assume that most people prefer a potential partner of a similar age. If we wanted to model such a concept with our initial setting of simply concatenating the feature vectors, we would be facing a task that is surprisingly hard to express and unlikely to be picked up by standard classification models such as logistic regression or decision trees. The reason is the particular interdependence of the two features: age_1 and age_2. Geometrically we are looking at a stripe in the two dimensional decision plane as shown in Figure 1. In the case of equality, the stripe is simply the identity line; in the relaxed case of relatively similar ages, the stripe widens and a preference of one gender for an age offset will shift the strip up or down.

This concept is closely related to the classical XOR problem [8] which is obviously not linearly separable. So how would we expect standard classification methods to perform on a stripes task?

1. **Naïve Bayes** is inherently unable to capture the class conditional dependence we need (the decision boundary is inherently linear).
2. A **Decision Tree** is conceptually able to represent arbitrary concepts provided sufficient amount of data. However, this is a hard problem for a tree for two reasons: each split is evaluated based on the performance of conditioning on a single variable and the splitting algorithm does not 'see' the dependence at this stage. In addition, it could have to painstakingly cut the diagonal line by small orthogonal approximation. If on top of this the dimensionality of the feature vector is large, trees are unlikely to perform well.
3. **Logistic Regression** is similar to naïve Bayes in its expressive power and will fail on a concept that is not linearly separable.
4. **KNN** is very efficient in dealing with inherently non-linear classification problems thanks to its ability to generate arbitrarily complex boundaries. This comes at a price however - the algorithm is very likely to overfit and is extremely sensitive to the choice of feature weighting.

2.1 Equality and Quadratic Expansion

We argue that while the age example is clearly contrived, many link prediction tasks have dependencies of this flavor. In our case we need the patent to 'match' the company: they must show some form of similarity/equality in the covered technology and the produced/used technology.

Similarly to our age stripe of similarity, this cannot be determined by a single linear inequality either. This suggests that a linear model would not be able to capture enough of the geometry of the data to provide a satisfactory classification. So what would it take for a linear model to express this type of concept? The difference of age_1-age_2 can be expressed linearly. But still the positive instances are around zero and the negative pairs are either very large or very small. In order to make this notion simpler, we can look at the square of the difference in age:

$$(age_1 - age_2)^2 = age_1^2 - age_2^2 - 2age_1age_2. \quad (1)$$

Now all positives have small values and all negatives have large ones and the concept becomes linearly separable. So if we were to introduce order 2 interaction effects between the two ages, a linear model can express the stripe concept easily. Moving to the general case of 'matching' between two objects, and in particular a patent and a company, it is important to observe that we do not require this concept to appear in the variables that correspond to the same word. Contrary to content-based recommendations that use precalculated distances [4], we do not want to assume which pairs will have such interactions but instead allow for all kind of pairs. We do not even require that all features are the same type (text) and would still like to account for 'matching' interactions. However, there is a very natural way to achieve just this. Instead of including order two interaction, we can just use an SVM with a quadratic kernel.

We tested the above outline intuition on our domain for the different kernels. The results, which we discuss in Section 4.2, indicate that indeed it is the closeness phenomenon that we need to utilize. Using SVM with degree 3 and higher kernel does not lead to improved performance and it suffers from the typical high-degree overfitting effect.

3. DATA DESCRIPTION

The starting point of our data collection is a manually assembled database where technology experts have linked a particular claim of an IBM patent to a company that is likely to be interested in the technology covered under this patent. We decided to remove patents from our analysis that were linked to more than 30 companies. These are very broad patents and we were concerned that they could bias our experiments. For most of the remaining patents (134), we have the complete content at our disposal. For our current experiments we focus only on the text in the abstract.

3.1 Company Information

As a first step we needed to identify relevant company information. Patent value analysis has traditionally looked at the IP 'footprint' of a company as a description of its technology and detailed business. While such information is in principle available, it does not suit our needs since many of the companies in question do not themselves have IP. Another alternative are general firmographic information such as sales, number of employees and SIC codes. This information, including a mapping of SIC to IPC, is available but is likely too coarse-grained and unreliable to provide good linkage with a given specific patent. We decided to look at a reliable and coherent source of company description instead. We found such data in two places: Wikipedia and Google Finance. Wikipedia provides fairly detailed description of companies but lacks the coverage of Google Finance. Google Finance provides a single paragraph with a short description for a large set of publicly traded as well as privately held companies. We collected those short summaries for 56 companies.

3.2 Text Preprocessing

We applied the standard text processing steps including stoplisting and stemming to both patent abstracts and company summaries. We further removed any word that did not appear at least 3 times in our corpus. This process left us with a vocabulary of 1288 patent and 386 company words.

3.3 Representation

For every of our 7504 pairs $\{\text{company}_\text{i}, \text{patent}_\text{j}\}$ we generated a vector $v_{i,j}$ of $1674 = 386 + 1288$ features as follows: We constructed an ordered list L of the total vocabulary as a concatenation of an ordered list of all selected words from the company description (after stoplisting and stemming) and an ordered list of selected words appearing in any patent abstract. Note that L might (and hopefully will) contain repetitions between words appearing in company descriptions and patent abstracts. Then, for a word $w_k \in L$

$$v_{i,j}[k] = \begin{cases} \sharp\text{times } \text{w}_\text{k} \text{ appears in } \text{c}_\text{i} & k = 1, \ldots, 386 \\ \sharp\text{times } \text{w}_\text{k} \text{ appears in } \text{p}_\text{j} & k = 387, \ldots, 1674 \end{cases}$$

where c_i and p_j denote the description of company_i and patent_j respectively. Then all features were normalized by removing their mean and dividing by the standard deviation.

4. EXPERIMENTS

We split our 7504 example pairs (134*56) into a training set of 5001 and a test set of 2503 examples. This split cleanly separates the patents, i.e. all links of a particular patent are either in the training or in the test set. However, a company can occur in a link (associated with a different patent) in both the training and test sets. Given the cross linkage it is impossible to build an entirely separate test universe both in terms of companies or patents without loss of significant numbers of links. We do not believe that this separation introduces significant biases to our results. This is confirmed by the low performance of a linear model.

Table 1: Model performance on the test set calculated at the max accuracy threshold.

Kernel	Weight	AUC	Precision	Recall	Lift
Linear	1	0.513	0	0	0
Linear	2	0.554	0	0	0
Linear	10	0.541	0	0	0
Quadratic	1	0.734	0.833	0.133	27
Quadratic	2	0.786	0.846	0.146	28
Quadratic	10	0.776	0.857	0.160	29
Poly deg 3	1	0.723	0.846	0.146	28
Poly deg 3	2	0.769	0.846	0.146	28
Poly deg 3	10	0.739	0.857	0.160	29
RBF γ=1	1	0.538	0.857	0.083	25
RBF γ=0.1	1	0.595	0.857	0.160	28
RBF γ=0.01	1	0.581	0.857	0.160	28

4.1 Classification Algorithms

We ran our experiments using SVM's[3] as implemented in the *SVMLight* package [5]. This choice reflects the structure of the domains (text with many features and limited number of observations) and our interest in exploring our hypothesis about the suitability of a simple feature vector representation of information from both link nodes in combination with different interaction effects and kernels. We decided to use the *SVMLight* inbuilt heuristic to select the penalty so that we can focus our attention on the impact of the choice of kernel. Our hypothesis was that the type of interactions that should be present in content-based link prediction are easily picked up by a quadratic kernel. We validated this intuition on four different types of kernels:

1. **Linear kernels** should only be able to model general popularity effects but should be unable to capture such equality-like interactions.
2. **Quadratic kernel** of degree two should be optimal for the suspect type of dependencies.
3. **Polynomial kernel** of degree greater two should not provide much additional predictive information while increasing the probability of overfitting.
4. **RBF kernel** has similar characteristics as a nearest neighbor classifier. It can express arbitrarily complex dependencies (far beyond what we are after) but is unable to weight the importance of different features.

4.2 Results

Table 1 shows our results for a variety of parameter settings and performance measures using the threshold of maximum accuracy (except for AUC which is threshold independent).

As expected, the linear model fails completely. The accuracy is maximized for a threshold which assigns all examples to the default class (no link). The AUC of close to 0.5 confirms that the model picks up close to no signal. Recall that the probabilistic interpretation of AUC is the probability that a positive test case has a higher score than a negative one.

The results of the quadratic kernel SVM are surprisingly good. The AUC is increasing when upweighting the positive class by a factor of 2. The overall result of AUC=0.786 is rather high for this domain. This is very strong evidence that indeed there are two-variable interactions that link a patent to a company. This could be learned easily once the kernel is expanded to consider second order terms. We observe that further increase of the weight on the positive training examples hurts the AUC but still improves the Precision, Recall, and Lift above the optimal threshold.

Adding another degree to the polynomial shows an overall decline in the ranking performance AUC, but the other measures remain equal. This confirms our notion that order 2 is the correct level of complexity and little is gained by further increase.

Something interesting is showing up in the results of the RBF kernel. While the AUC is overall rather low (below 0.6), the retrieval performance above the optimal threshold is very comparable to the previous two methods for larger parameter values of gamma. The weighting seems to have no impact on the RBF results so we did not show them in detail. The somewhat contradictory results can be explained by the company overlap between test and training set. RBF is able to identify some companies (not really links) that have many patent links - some of which are in the training and some in test. They fail however to predict links for new companies with few existing links in the training.

Overall we find strong evidence for our intuition of equality related two-way interactions between variables. Our very simple feature vector representation becomes highly predictive one we apply models that are of sufficient complexity.

Additional experiments We ran a few additional experiments to investigate the impact of some transformations. In particular we calculated the square root and the log of word occurrences. Quadratic kernels allow for separating more complicated regions than linear SVM, but at a price of increased sensitivity to outliers. Replacing features with square roots remedied that phenomenon. However, the impact of such transformations seemed minor. The performance increased slightly, but the overall results remained highly consistent with the ones reported above. We also considered transductive learning and included the test observations as unlabeled examples in the training (see [6]). This seems to hurt our performance, as the AUC of the quadratic kernel decreased to 0.759.

5. CONCLUSION AND FUTURE WORK

We have presented our initial work on the application of recommender systems to the emerging domain of patent marketing. We also identify what we believe to be a fairly universal property of a content interdependence in the context of link prediction. This type of dependence is related to the notion of equality and similarity and it is the opposite of the classical XOR problem. As such, it is inherently unsuitable for linear modeling approaches with a simple feature vector representation. We argue and show empirically that using a quadratic kernel one can capture the optimal level of complexity for this type of dependence, based on the simple representation introduced in this paper.

We are planning to extend this work in a number of directions. The short term goal of our project is to improve the prediction performance of our patent-company recommender by exploring additional features. This may include the IP footprint, the actual claim content of the patent, and the use of related patents (through citations) as well as related companies. In addition we will need to assess the reliability of the true 'out of universe' predictions on entirely new patents and companies. On the more scientific level, we are going to seek further validation of our conjecture that link-prediction tasks often exhibit the type of pairwise interactions that we implicitly observe in this domain. If this were to be the case, this would suggest the use a simple feature vector representation with quadratic kernels as a suitable modeling approach. A follow-up analysis would be the appropriate incorporation of this kind of content information with the more commonly used graph information into a single inference approach.

6. ACKNOWLEDGMENTS

We thank Joe Polimeni and Jim Ward for many discussions on the broad subject of patent licensing.

7. REFERENCES

[1] C. Basu, H. Hirsh, and W. Cohen. Recommendation as classification: using social and content-based information in recommendation. In *AAAI '98/IAAI '98*, pages 714–720, 1998.

[2] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering, 1998.

[3] C. Cortes and V. Vapnik. Support-vector networks. In *Machine Learning*, pages 273–297, 1995.

[4] M. A. Hasan, V. Chaoji, S. Salem, and M. Zaki. Link prediction using supervised learning. In *Proc. of SDM 06 workshop on Link Analysis, Counterterrorism and Security*, 2006.

[5] T. Joachims. Making large-scale support vector machine learning practical. In *Advances in kernel methods: support vector learning*, pages 169–184, Cambridge, MA, USA, 1999. MIT Press.

[6] T. Joachims. Transductive inference for text classification using support vector machines, 1999.

[7] X. Li and H. Chen. Recommendation as link prediction: a graph kernel-based machine learning approach. In *Proceedings of the 9th Joint Conference on Digital libraries*, pages 213–216, 2009.

[8] M. L. Minsky and S. A. Papert. *Perceptrons*. MIT Press, Cambridge, MA, 1969.

[9] M. J. Pazzani. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, 13(5-6):393–408, 1999.

[10] U. Shardanand and P. Maes. Social information filtering: algorithms for automatic word of mouth. In *Conference on Human Factors in Computing Science.*, pages 210–217, 1995.

[11] T. Zhang and V. S. Iyengar. Recommender systems using linear classifiers. *Journal Machine Learning Research*, 2:313–334, 2002.