



# Upstream, Downstream or Competitor? Detecting Company Relations for Commercial Activities

Yi-Pei Chen, Ting-Lun Hsu<sup>(✉)</sup>, Wen-Kai Chung, Shih-Chieh Dai,  
and Lun-Wei Ku

Institute of Information Science, Academia Sinica, Taipei, Taiwan  
{ypc,hsutingl,wkchung,sjdai,lwku}@iis.sinica.edu.tw

**Abstract.** Due to intricate network in industry business and high cost of supervision, financial institutions usually focus on supervising core enterprises in a supply chain instead of all corporations, which indirectly lower the strength and efficiency of financial institutions as a role of capital supervisor and credit-risk transformer. Furthermore, banks require these corporations to provide correct information by themselves, which lacks of the objectivity of the source information and increases the supervision cost for these banks. Thus, we summarize a company relation detection task in hope to exposing more information about companies to investors and banks by learning a system from public available datasets. We regard this task as an classification problem, and our system can predict relations between any two companies by learning on both structured and unstructured data. To the best of our knowledge, it's the first time to implement deep learning technique to this task. A F1 score 0.769 is achieved from our system.

**Keywords:** Business dashboards · Commercial activity ·  
Company relation detection · Knowledge graph ·  
Multi-relational graph embedding · Deep learning application

## 1 Introduction

Traditionally, suppliers and buyers in a supply chain have competing financial interests: while buyers want to pay as late as possible, suppliers want to receive money as early as possible. Supply chain finance (SCF) is a solution to bridge these conflicting interests. By transforming risks from upstream suppliers and downstream buyers to professional finance institutions, these experts such as banks can supervise all players in the chain, thus providing short-term credit for optimizing working capital for both sides.

Nonetheless, a typical SCF can be very large and complicated. Therefore, banks usually focus on supervising core enterprises and related business instead

---

Y.-P. Chen, T.-L. Hsu, W.-K. Chung and S.-C. Dai—Equal contribution.

of all companies in the same chain due to the difficulty and expensive cost of information collection. To alleviate this problem and expose more information of companies to banks and investors, we focus on predicting the relation between any two given companies by leveraging news articles and public-available dataset (datasets are detailed in the section Dataset).

Generally speaking, while we know the high level concept about who are suppliers and buyers in a specific industry chain, we cannot know this kind of information for a specific company: who are indeed the suppliers and clients of a particular company. Take the semiconductor industry as example, we know that IC design companies should be upstream suppliers of IC manufacturing companies, but given any two specific companies in this industry, we don't know if they really have relationship with each other.

Thus, by using public available datasets such as news data and government released data, we hope to predict the relations between two given company entities. We simplify the task as a classification problem and tries to predict the following four types of relations: upstream supplier, downstream retailer, competitor and no relation.

The task is separated into two part. First, we utilize the datasets to learn embeddings for companies, which can encode the information from both news and government released data. Thereafter, we build a classifier to predict the relationship between any two given companies.

In the following, Sect. 2 introduces the background of related work; Sect. 3 describes the dataset we use; Sect. 4 details the proposed system; Sect. 5 puts experiment; followed by result and analysis in Sect. 6, and finally Sect. 7 gives summary of this paper.

## 2 Background

In real world, companies that are traded at stock exchange or over-the-counter markets are required by law to report financial statements. Despite this, information about suppliers and clients of a company are not compulsory, and most of the time companies regard such information as secrets from their potential competitors, which makes it more impenetrable to the public. Thereafter, past researches usually pre-assumed upstream, downstream and competitor relations of companies based on the industry they belong to and the products they make.

Hsieh et al. [1] predefined upstream and downstream groups based on industry chain and applied data mining technique to trading data in order to find stock price relations between these groups.

## 3 Dataset

To train our system, we crawl both structured and unstructured data. For the unstructured data, the financial news data is crawled. And for the structured data, Taiwan company corpus and relation data between companies are crawled. Following details each of the above datasets.

**Table 1.** A segment of an article from Economic Daily News. It is originally written in Chinese. Here we translate the article to English for demonstration purpose.

---

### The Release of iPhone X

The new model iPhoneX was released yesterday. Highlight of this phone includes wireless charging, face recognition and Augmented Reality. All these functions should have a big influence on the cellphone market, which should also benefit the related manufactures in Taiwan. Though manufactures in Taiwan are not the main suppliers of chips, TSMC, Pegatron, Largan Precision, and Foxconn still play important roles in the market. (...skip...) However, the problems of face recognition still exist: the high cost and low yield rate. As a result, iPhone 8 and iPhone 8+ keep the original home button for fingerprint recognition

---

First, financial news data is crawled from a local media Economic Daily News<sup>1</sup> in Taiwan. An article from Economic Daily News is shown in Table 1 for illustration. The dataset contains 22,400 news from year 2016 to 2017 containing 352,470 Chinese and English words and numbers. To parse Chinese language, we use CKIP parser [2] to segment news data into word tokens. The number of total unique words (vocab size), including English words and numbers, is 38,372. Next, We crawl the company corpus from Taiwan Stock Exchange (TWSE) (both listed equities and TPEX equities)<sup>2</sup> to identify company entities in our news dataset. 1,704 companies can be found in total.

Thirdly, the relation data between companies, the ground truth data in our task, is crawled from Money Link website<sup>3</sup>, which gives upstream, downstream, and competitor relation information of companies. Table 2 shows some statistics about our ground truth data. Since it doesn't contain "no-relation" data, we do negative sampling to train our system, as described in the section Experiments. After that, we then construct triples in order to build a knowledge graph. Formal definition is given below.

Given two companies A and B, we construct the ordered triple as

- (A, B, upstream), if A is an upstream supplier of B,
- (A, B, downstream), if A is a downstream client of B,
- (A, B, competitor), if A is a competitor of B,
- (A, B, no-relation), if A has no relation with B.

Notice that (A, B, upstream) holds if and only if (B, A, downstream), (A, B, competitor) holds if and only if (B, A, competitor), and (A, B, no-relation) holds if and only if (B, A, no-relation). Consequently, we construct all possible triples as long as they hold. Fro example, if (B, A, downstream) exists in the dataset while (B, A, downstream) does not, we will add (B, A, downstream) to the dataset.

---

<sup>1</sup> <https://money.udn.com/money/index>.

<sup>2</sup> [http://www.twse.com.tw/en/page/products/stock\\_code.html](http://www.twse.com.tw/en/page/products/stock_code.html).

<sup>3</sup> <https://ww2.money-link.com.tw>.

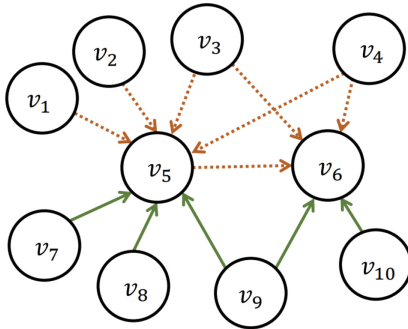
**Table 2.** The table shows the size of our ground truth data, the relations between two companies. Since upstream and downstream relations are conceptually inverse with each other, they have the same number of pairs. Unique company size in Money Link: 1608

Dataset	relation type			
	# of pairs	Competitor	Upstream	Downstream
Money Link	19,150	9,496	4,827	4,827

All information described in the above is then used to train the embeddings for companies. These embeddings can be put into our classifier for relation prediction between any two given companies. Notice that, although we conduct experiments on Taiwan markets, our system can indeed be applied on other markets as long as datasets are prepared. Details about how we exactly train the embeddings for company are elaborated in the next section.

## 4 System Overview

There are two main steps in our system: (1) embeddings training for companies, and (2) classifier training for companies' relation detection by leveraging embeddings trained by (1).



**Fig. 1.** Sample multi-relational directed graph. Directed arrows show directed links. The two relations in this graph are presented as solid and dashed lines, respectively. When predicting  $v_5$ , linked vertices  $v_1$  to  $v_4$  and  $v_7$  to  $v_9$  are used as the  $l$ 's in Eq. 1; when predicting  $v_6$ , linked vertices  $v_3$ ,  $v_4$ ,  $v_5$ ,  $v_9$  and  $v_{10}$  are used as the  $l$ 's. Embeddings will be generated for all vertices except  $v_6$ , whose links are all inlinks.

### 4.1 Embedding Training Stage

Because our datasets contain unstructured data (news) and structured data (relation information between companies), we develop a multi-relational graph

embedding to encode both kinds of information. We will first introduce how our multi-relational graph embedding works, followed by how we utilize this to build embeddings for companies.

**Multi-relational Graph Embedding.** To encode the graph structure by embeddings, we predict a vertex given its linked vertices, where different relation vertices are mapped to the same space through different relation transform matrices. Formally, given a graph with  $k$  multi-relations  $r_1, r_2, \dots, r_k$ , and the vertices to be predicted  $O$ , the objective function is to maximize the average probability

$$\frac{1}{|O|} \sum_{o_i \in O} \sum_{r_j \in R} \log p(o_i | l_{j,1}, l_{j,2}, \dots, l_{j,|l_j|}) \quad (1)$$

where  $R$  is the set of all relations, and  $l_{j,1}, l_{j,2}, \dots, l_{j,|l_j|}$  are the linked vertices of  $o_i$  with the relation  $r_j$ . As shown in Eq. 2, for each vertex  $o_i$ , we use one multiclass classifier with softmax to obtain the conditional probabilities, and it is repeated for each relation.

$$p(o_i | l_{j,1}, l_{j,2}, \dots, l_{j,|l_j|}) = \frac{e^{y_{o_i}}}{\sum_s e^{y_s}} \quad (2)$$

Each of  $y_s$  is the unnormalized probability to each output vertex  $o_s$ , computed as

$$y = Uh(l_{j,1}, l_{j,2}, \dots, l_{j,|l_j|}; E) + b \quad (3)$$

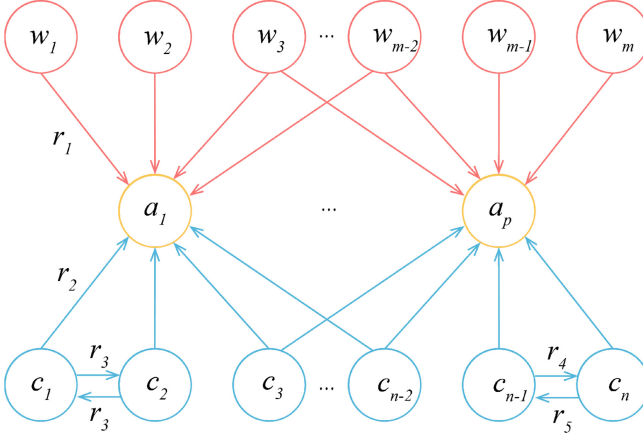
where  $U$  and  $b$  are the output layer weights and bias, respectively;  $h$  is the output of a hidden layer constructed by the transformation for relation  $r_j$  of the embeddings of  $l_j$  extracted from  $E$ , as shown in Eqs. 4 and 5:

$$h = t_j \cdot h_j \quad (4)$$

$$h_j = [v_{l_{j,1}}, v_{l_{j,2}}, \dots, v_{l_{j,|l_j|}}] \cdot E \quad (5)$$

where  $t_j$  transforms the extracted embeddings of each relation to the same space, and  $v_{l_{j,n}} \in \mathbb{R}^{1 \times |l_j|}$  is a one-hot vector used to retrieve the corresponding embeddings of  $l_j, n$ . Although Eq. 3 takes into account all vertices for prediction, in practice, we train  $E$ ,  $U$ , and  $b$  using each of the linked vertices in  $l_j$  as the sample to predict the vertex  $o_i$ . Therefore, unlike some intuitive methods which treat each relation as a separate graph and then concatenate the generated embeddings, we encoded through  $t_j$ , places all relations on the same graph for consideration when generating embeddings.

It is noticed that in this framework, only vertices in  $l$  (i.e., those who have outlinks) learn embeddings. Vertices with only inlinks are not embedded, since they are only predicted by other vertices. Figure 1 illustrates an example.



**Fig. 2.** Multi-relational graph: Assume  $m$  words,  $n$  companies, and  $p$  articles exist in the dataset. Moreover,  $c_1$  and  $c_2$  are competitor while  $c_{n-1}$  is  $c_n$ 's upstream supplier and  $c_n$  is  $c_{n-1}$ 's downstream retailer.

**Embedding Training for Companies.** To apply our multi-relational graph embedding to the relation classification task, we first construct a graph representing the entities and relations in the experimental datasets. Engaged entities include the articles, the article content words, and the companies exist in articles. The steps to construct this graph are as follows:

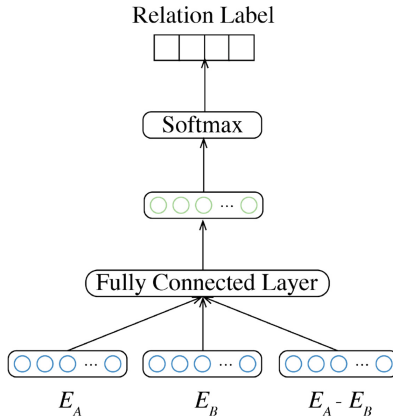
1. Entity: each article, each company, and all distinct words in the dataset are vertices.
2. Word-inclusion relation ( $r_1$ ): if a word belongs to the article, create a directed link from the word vertex to the article vertex.
3. Company-engagement relation ( $r_2$ ): if a company exists in the article, create a directed link from the company vertex to the article vertex.
4. Competitor relation ( $r_3$ ): if company A is company B's competitor, create a directed link from vertex company A to vertex company B.
5. Upstream relation ( $r_4$ ): if company A is company B's upstream supplier, create a directed link from vertex company A to vertex company B.
6. Downstream relation ( $r_5$ ): if company A is company B's downstream retailer, create a directed link from vertex company A to vertex company B.

Figure 2 is an illustration of the constructed graph. Note that even there is no link between words and companies, the words in this graph would still affect the embeddings of companies due to the universal weights  $U$  and bias  $b$  in Eq. 3. During the back-propagation in the training process, the weights  $U$  and bias  $b$  will be influenced by the words, thus influence the embedding of companies.

## 4.2 Classifier Training for Relation Detection

After getting the graph embeddings for companies, we then use them for relation classifier training. Figure 3 shows the architecture of the classifier.

The inputs of the classifier are the graph embeddings of two given companies A and B:  $E_A$  and  $E_B$ . We first concatenate  $E_A$ ,  $E_B$  and  $E_A - E_B$ . We do an element-wise subtraction because there might be some patterns between two embeddings in the latent space. For example, if company A is B’s upstream supplier and C is D’s upstream supplier,  $E_A - E_B$  might be similar to  $E_C - E_D$ . Thereafter, the concatenation is put into a fully connected layer, followed by a softmax layer that outputs the probability of each four options: Competitor, Upstream, Downstream, No-Relation.



**Fig. 3.** An illustration of classifier

After training, we can put any two companies into our system, and the system will predict the relation between the input company pair.

## 5 Experiments

To train this classifier, we use the ground truth from Money Link website. There are three types of relation contained in the dataset: upstream supplier, downstream retailer, and competitor. We also generate the negative samples, which means no relation between two companies. There are 9,496 competitors, 4,827 upstream, 4,827 downstream, and 6,432 negative samples.

The upstream and downstream are the reverse concept of each other. That is, (A, B, upstream) if and only if (B, A, downstream). Hence, to make our experiment more reliable, we require that both the upstream and its reverse downstream sample should be categorized into the same set when splitting data into train/val/test set. Moreover, because (A, B, Competitor) also means (B, A,

Competitor), as well as No-relation. As the result, the pairs should also be split into same train/val/test set. The split ratio is 0.8/0.1/0.1.

In order to know how well our graph embedding method utilizes the information from different datasets, we have three variation during the graph embeddings training stage.

1. Both Economic Daily News in Taiwan and the relation pairs between companies (upstream, downstream, competitor) in training and validation set from Money Link website are put into the graph embedding training.
2. Only Economic Daily News are put into the graph embedding training.
3. Only relation pairs in training and validation set are put into the graph embedding training.

Moreover, we use Glove from Pennington et al. [3] to train word and company embeddings on the Economic Daily News. Glove uses unsupervised learning algorithm to obtain vector representation for each token in the corpus. During Glove’s training process, it considers both the local context windows and global word-word co-occurrence probability from the corpus. As a result, it should be a good baseline to measure whether our model captures the semantic information in the news.

In the end, we generate randomly initialized embeddings for sanity check that both our graph embedding and Glove actually utilize the information from the news for classification task.

To train the classifier, we use one of the optimizer: Adadelta [4], Adam [5], RMSprop [6], SGD [7,8] with momentum [9], with cross entropy loss and apply dropout [10] and L2 regularization [11]. We do the grid search on the validation set to pick the best hyperparameters and optimizer. Thereafter, we put training and validation set together, train it with the best hyperparameters and the picked optimizer, and test the result on the test set. To better understand the effectiveness of embeddings training stage, we do experiment on both finetuning and not finetuning the embeddings. The finetuning means that we dynamically update the embeddings during training the classifier.

## 6 Results

We measure each of the setting’s precision, recall, and macro F1 score. The result is showed in Table 3. From Table 3, we can see that if we dynamically update the embeddings during the classifier training, the performance of all the settings except *Random* are pretty similar. However, even for the best setting, *News + Relation*, it only outperforms *Random* by 2.1% F1 score. This result is not expected because *Random* does not utilize any information in the embedding training stage. In only updates its company embeddings when training the classifier, but it could still achieve F1 score of 0.748.

To further investigate what the model actually learns during the training of embeddings, we do an experiment that do not allow the dynamically updating embeddings when we train the classifier. The performance of *Random* drops

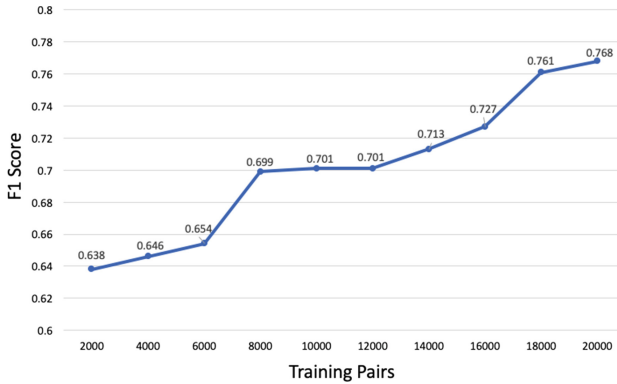


**Table 3.** The metric result for each class and the total performance. The scores in overall are calculated by macro averaging each measure of the listed four classes

		Finetune			No-Finetune		
		Precision	Recall	F1 score	Precision	Recall	F1 score
News + Relation	Competitor	0.876	0.831	0.853	0.684	0.828	0.749
	Upstream	0.778	0.714	0.745	0.701	0.578	0.634
	Downstream	0.739	0.729	0.734	0.733	0.581	0.649
	No-relation	0.697	0.800	0.745	0.619	0.584	0.601
	Overall	0.772	0.768	0.769	0.685	0.643	0.658
News	Competitor	0.870	0.830	0.849	0.546	0.633	0.586
	Upstream	0.723	0.727	0.725	0.606	0.470	0.530
	Downstream	0.727	0.711	0.719	0.598	0.485	0.535
	No-relation	0.706	0.764	0.734	0.396	0.413	0.404
	Overall	0.756	0.758	0.757	0.536	0.500	0.514
Relation	Competitor	0.837	0.842	0.839	0.766	0.828	0.796
	Upstream	0.771	0.712	0.741	0.706	0.558	0.623
	Downstream	0.743	0.720	0.732	0.680	0.599	0.637
	No-relation	0.713	0.761	0.736	0.639	0.707	0.671
	Overall	0.766	0.759	0.762	0.698	0.673	0.682
Glove	Competitor	0.837	0.865	0.851	0.545	0.654	0.594
	Upstream	0.733	0.730	0.731	0.637	0.536	0.582
	Downstream	0.745	0.727	0.736	0.543	0.520	0.531
	No-relation	0.744	0.720	0.732	0.388	0.321	0.351
	Overall	0.765	0.760	0.762	0.528	0.508	0.515
Random	Competitor	0.858	0.795	0.825	0.484	0.687	0.568
	Upstream	0.722	0.725	0.724	0.464	0.527	0.494
	Downstream	0.694	0.744	0.718	0.543	0.392	0.455
	No-relation	0.706	0.748	0.726	0.264	0.118	0.163
	Overall	0.745	0.753	0.748	0.439	0.431	0.420

dramatically to F1 score of 0.420, while others' performance drops some extent, but do much better than *Random*. In detail, *Relation* does the best, *News + Relation* second, followed by *Glove* and *News*. Previously, *News + Relation* outperforms the *Relation*, but here *Relation* outperforms the *News + Relation*. We guess this is because a large portion of information in the news is not related to the relation classification for companies, but those words that contain noise are still linked to the articles and being optimized when we train our graph embedding. If we do not finetune embeddings in the classifier, those noise cannot be filtered out, thus hurt the performance of the classifier. On the other hand, despite the noise contained in the news, the news still provides certain amount of information for the classifier because *Glove* and *News* outperform *Random* by almost 10%.

Based on the observation that (1) The F1 score of all the settings in Finetune are similar. (2) All the other settings' performance is much better than *Random* in No-Finetune. We can know that the news actually provides some information for the relation classification task. However, most the information extracted from the news are actually included in the relation pairs that we used to train the classifier.



**Fig. 4.** F1 score with respect to the number of training relation pairs used in the classifier.

To measure how the amount of training data used in the classifier influences the performance, we do an experiment on the embeddings from the best setting, Finetune *News+Relation*. We use different amount of data to train the classifier, and the result is showed in Fig. 4. We have two observation here: (1) The more the training data, the better the performance. (2) When we use only 2000 training relation pairs, which is 1/10 of all the training relation pairs, the performance does not drop a lot. This means that the effectiveness of embeddings will emerge when the training data is not a lot.

Lastly, by comparing *News* and *Glove*, our graph embedding has the competing ability to capture the semantic information from News data. Moreover, our model can utilize both the structured data(relation pairs) and unstructured data(News) while *Glove* embedding can only be trained on word corpus.

## 7 Conclusion

In this paper, we propose a system which can utilize the data from both News dataset and relation pairs dataset to detect relation between companies. In the meantime, we also propose a method to train multi-relational graph embedding, which can encode information from varies kinds of data as long as a graph is constructed. The graph embedding is used to capture the information from our datasets, thus helping the training of relation classifier. After training, our system can provide the relation between companies. This kind of information should be helpful for the current complex financial market. Although improvements can be made on our static method such as adding time into consideration to be dynamic prediction or by search for more related dataset, we believe this method has shown its possibility to help exposing more information hidden in industry business and it is a worth-trying direction. Furthermore, the current approach is not restricted to Taiwan market, which can be applied to any other markets as long as the related data is available.

## References

1. Hsieh, Y.L., Yang, D.-L., Wu, J.: Using data mining to study upstream and downstream causal relationship in stock market. *Computer* **1**, F02 (2005)
2. Ma, W.-Y., Chen, K.-J.: Introduction to CKIP Chinese word segmentation system for the first international Chinese Word Segmentation Bakeoff, pp. 168–171. Association for Computational Linguistics (2003)
3. Pennington, J., Socher, R., Manning, C.D.: GloVe: global vectors for word representation. In: *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543 (2014). <http://www.aclweb.org/anthology/D14-1162>
4. Zeiler, M.D.: ADADELTA: An Adaptive Learning Rate Method. *CoRR* abs/1212.5701 (2012). <http://arxiv.org/abs/1212.5701>
5. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. *CoRR* abs/1412.6980 <http://arxiv.org/abs/1412.6980> (2014)
6. Hinton, G.: Neural networks for machine learning - lecture 6a - overview of mini-batch gradient descent (2012)
7. Robbins, H., Monro, S.: A stochastic approximation method. *Ann. Math. Statist.* **22**(3), 400–407 (1951). <https://doi.org/10.1214/aoms/1177729586>. <https://projecteuclid.org/euclid.aoms/1177729586>
8. Kiefer, J., Wolfowitz, J.: Stochastic estimation of the maximum of a regression function. *Ann. Math. Statist.* **23**(3), 462–466 (1952). <https://doi.org/10.1214/aoms/1177729392>. <https://projecteuclid.org/euclid.aoms/1177729392>
9. Qian, N.: On the momentum term in gradient descent learning algorithms. *Neural Netw.* **12**(1), 145–151 (1999). <http://www.sciencedirect.com/science/article/pii/S0893608098001166>
10. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15**, 1929–1958 (2014). <http://jmlr.org/papers/v15/srivastava14a.html>
11. Ng, A.Y.: Feature selection, L1 vs. L2 regularization, and rotational invariance. In: *Proceedings of the Twenty-First International Conference on Machine Learning, ICML 2004, Banff, Alberta, Canada*, p. 78. ACM, New York (2004). <http://doi.acm.org/10.1145/1015330.1015435>