Singular Value Decomposition for Feature Selection in Taxonomy Learning

Francesca Fallucchi and Fabio Massimo Zanzotto *
Via del Politecnico 00133 Rome, Italy
{*fallucchi,zanzotto*}@*info.uniroma2.it*

Abstract

In this paper, we propose a novel way to include unsupervised feature selection methods in probabilistic taxonomy learning models. We leverage on the computation of logistic regression to exploit unsupervised feature selection of singular value decomposition (SVD). Experiments show that this way of using SVD for feature selection positively affects performances.

1 Introduction

Taxonomies and, in general, networks of words connected with transitive relations are extremely important knowledge repositories for a variety of applications in natural language processing (NLP) and knowledge representation (KR). In NLP, taxonomies such as WordNet [17] are widely used in intermediate tasks such as word sense disambiguation (e.g. [1]) and selectional preference induction (e.g., [25]) as well as in final applications such as question answering (e.g., [4]) and textual entailment recognition (e.g. [5]). In KR, taxonomies as well as other word networks are the bulk of domain ontologies.

To be effectively used in NLP and KR applications, taxonomies and knowledge repositories have to be large or, at least, adapted to specific domains. Yet, even huge knowledge repositories such as WordNet [17] are extremely poor when used in specific domains such as the medical domain (see [29]). Automatically creating, adapting, or extending existing knowledge repositories using domain texts is, then, a very important and active area. A large variety of methods have been proposed: ontology learning methods [16, 3, 19] in KR as well as knowledge harvesting methods in NLP such as [13, 21]. These learning methods use variants of the distributional hypothesis [12] or exploit some induced lexical-syntactic patterns (originally used in [26]). The task is generally seen as a classification (e.g., [22, 27]) or a clustering (e.g., [3]) problem. This allows the use of machine learning models.

Yet, as any other machine learning problem, knowledge harvesting and ontology learning models exploit the above hypothesis to build feature spaces where instances, i.e., words as in [22] or word pairs as in [27], are represented. These feature spaces are used to determine whether or not new word pairs coming from the text collection have to be included in existing knowledge repositories. Decision models are learnt using existing knowledge repositories and then applied to new words or word pairs. Generally, these models use as features all the possible and relevant generalized contexts where words or word pairs can appear. For example, possible features in the word pair classification problem are *"is a"* and *"as well as"*. Given the nature of the problem, these feature spaces can then be huge as they include all potential relevant features for a particular relation among words. Relevant features are not known in advance. Yet, large feature spaces can have negative effects on machine learning models such as increasing the computational load and introducing redundant or noisy features. Feature selection is the solution (see [11]).

In this paper, we want to study how to improve performances of taxonomy learning methods by using feature selection. We focus on the probabilistic taxonomy learning model introduced by [27] as it uses existing taxonomies exploiting the transitivity of the *isa* relation. Leveraging on the particular model, we propose a novel way of using singular value decomposition (SVD) as unsupervised model for feature selection. In a nutshell, given the probabilistic model for taxonomy learning, we use SVD as a way to compute the pseudo-inverse matrix needed in logistic regression. We will analyze if our method for using unsupervised feature selection positively affect performances.

Before staring, in Sec. 2 we will shortly review methods for taxonomy learning and for feature selection. We motivate our choice of working within the probabilistic setting. In Sec. 3, as SVD is the core of our method, we will then introduce SVD as unsupervised feature selection model. In Sec. 4 we then describe how we introduced SVD as natural feature selector in the probabilistic taxonomy learning model introduced by [27]. To describe how we use SVD as natural feature selector, we will shortly review the logistic regression used to compute the taxonomy learning model. We will describe our experiments in Sec. 5. Finally, in Sec. 6, we will draw some conclusions and describe our future work.

2 Related work

Extracting knowledge bases from texts is one of the major goal of NLP and KR. These methods can give an important boost to knowledge-based systems. In this section we want to shortly analyze some of these methods in order to motivate our choice to work within an existing probabilistic model for learning taxonomies. We also review the more traditional models for super-

*DISP University Rome "Tor Vergata"

vised and unsupervised feature selection.

The models for automatically extracting structured knowledge, such as taxonomies, from texts use variants of the distributional hypothesis [12] exploit some induced lexical-syntactic patterns (originally used in [26]).

The distributional hypothesis is widely used in many approaches for taxonomy induction from texts. For example, it is used in [3] for populating lattices, i.e. graphs of a particular class, of formal concepts.

Lexical syntactic patterns are also a source of relevant information for deciding whether or not a particular relation holds between two words. This approach has been widely used for detecting hypernymy relations such as in [13, 18], for other ontological relations such as in [21], or for more generic relations such as in [24, 28]. These learning models generally use the hypothesis that two words are related according to a particular relation if these often appear in specific text fragments.

Despite the wide range of models for taxonomy learning, only very few exploit the structure of existing taxonomies. The task is seen as building taxonomies from scratch. In [3], for example, lattices and the related taxonomies are the target. Yet, existing taxonomies may be used to drive the process of building new taxonomies. In [19], WordNet [17] and WordNet glosses are used to drive the construction of domain specific ontologies. In [22], taxonomies are augmented exploiting their structure. Inserting a new word in the network is seen as a classification problem. The target classes are the nodes of the existing hierarchy. The distributional description of the word as well as the existing taxonomy structure is used to make the decision. This model is purely distributional. In [27], a probabilistic model exploiting existing taxonomies is introduced. This model is purely based on lexical-syntactical patterns. Also in this case, the insertion of a new word in the hierarchy is seen as a binary classification problem. Yet, the classification decision is taken over a pair of words, i.e., a word and its possible generalization. The probabilistic classifier should decide if this pair belongs or not to the taxonomy.

The probabilistic taxonomy learning models has at least two advantages with respect to the other models. The first advantage is that it coherently uses existing taxonomies in the expansion phase. Both existing and new information is modeled in the same probabilistic way. The second advantage is that classification problem is binary, i.e., a word pair belongs or not to the taxonomy. This allows to build a unique binary classifier. This is not the case for models such as the one of [22], where we need a multi-class classifier or a set of binary classifiers. For these two reasons, we are using the probabilistic taxonomy learning setting for our study.

Yet, in applications involving texts such as taxonomy learning, machine learning models are exposed to huge feature spaces. This has not always positive effects. A first important problem is that huge feature spaces require large computational and storage resources for applying machine learning models. A second problem is that more features not always result in better accuracies of learnt classification models. Many features can be noise. Feature selection, i.e., the reduction of the feature space offered to machine learners, is seen as a solution (see [11]).

There is a wide range of feature selection models that can be classified in two main families: *supervised* and *unsupervised*. Supervised models directly exploit the class of the instances for determining if a feature is relevant or not. The idea is to select features that are highly correlated with final target classes. Information theoretic ranking criteria such as mutual information and information gain are often used (see [8]). Unsupervised models are instead used when the information on classes of instances is not available at the training time or it is inapplicable such as in information retrieval. Straightforward and simple models for unsupervised feature selection can be derived from information retrieval weighting schemes, e.g., term frequency times inverse document frequency ($tf * idf$). In this case, relevant features are respectively those appearing more often or those more selective, i.e., appearing in fewer instances.

Feature selection models are also widely used in taxonomy learning. For example, attribute selection for building lattices of concepts in [3] is done applying specific thresholds on specific information measures on the attributes extracted from corpora. This models uses conditional probabilities, point-wise mutual information, and a selectional-preference-like measure as the one introduced in [25].

3 Unsupervised Feature Selection with SVD

A very important way of unsupervised feature selection is the application of the SVD. As this is the bulk of our methodology we will review how SVD can be used for this purpose. SVD has been largely used in information retrieval for reducing the dimension of the document vector space [7].

SVD, originally, is a decomposition of a rectangular matrix. Given a generic rectangular $n \times m$ matrix A, its singular value decomposition is $A = U\Sigma V^T$ where U is a matrix $n \times r$, V^T is a $r \times m$ and Σ is a diagonal matrix $r \times r$. The diagonal elements of the Σ are the *singular values* such as $\delta_1 \geq \delta_2 \geq ... \geq \delta_r > 0$ where r is the rank of the matrix A. For the decomposition, SVD exploits the linear combination of rows and columns of A.

There are different ways of using SVD as unsupervised feature reduction. An interesting way is to exploit its approximated computations, i.e. :

$$A \approx A_k = U_{m \times k} \Sigma_{k \times k} V^T_{k \times n} \quad (1)$$

where k is smaller than the rank r. The computation algorithm [10] allows to stop at a given k different from the real rank r. The property of the singular values, i.e., $\delta_1 \geq \delta_2 \geq ... \geq \delta_r > 0$, guarantees that the first k are bigger than the discarded ones. There is a direct relation between the informativeness of the i-th new dimension and the singular value δ_i. High singular values correspond to dimensions of the new space where examples have more variability whereas low singular values determine dimensions where examples have a smaller variability (see [15]). These latter dimensions can be then hardly used as efficient features in learning. The possibility of computing approximated versions of matrices gives a powerful method for feature selection and filtering as we can decide in advance how many features or, better, linear combination of original features we want to use.

4 Probabilistic Taxonomy Learning and SVD

In this section we will firstly introduce the probabilistic model (Sec. 4.1) and, then, we will describe how SVD is used as feature selector in the logistic regression that estimates the probabilities of the model (Sec. 4.2). To describe this part we need to go in depth into the definition of the logistic regression and some ways of computing it.

4.1 Probabilistic model

In the probabilistic formulation [27], the task of learning taxonomies from a corpus is seen as a maximum likelihood problem. The taxonomy is seen as a set T of assertions R over pairs $R_{i,j}$. If $R_{i,j}$ is in T, i is a concept and j is one of its generalization (i.e., the direct or the indirect generalization). For example, $R_{dog,animal} \in T$ describes that *dog* is an *animal* according to the taxonomy T.

The main probabilities are then: (1) the prior probability $P(R_{i,j} \in T)$ of an assertion $R_{i,j}$ to belong to the taxonomy T and (2) the posterior probability $P(R_{i,j} \in T|\overrightarrow{e}_{i,j})$ of an assertion $R_{i,j}$ to belong to the taxonomy T given a set of evidences $\overrightarrow{e}_{i,j}$ derived from the corpus. These evidences are derived from the contexts where the pair (i, j) is found in the corpus. The vector $\overrightarrow{e}_{i,j}$ is a feature vector associated with a pair (i, j). For example, a feature may describe how many times i and j are seen in patterns like *"i as j"* or *"i is a j"*. These among many other features are indicators of an is-a relation between i and j (see [13]).

Given a set of evidences E over all the relevant word pairs, the probabilistic taxonomy learning task is defined as the problem of finding a taxonomy $\widehat{T}$ that maximizes the probability of having the evidences E, i.e.:

$$\widehat{T} = \arg\max_{T} P(E|T)$$

In [27], this maximization problem is solved with a local search. What is maximized at each step is the ratio between the likelihood $P(E|T')$ and the likelihood $P(E|T)$ where $T' = T \cup N$ and N are the relations added at each step. This ratio is called multiplicative change $\Delta(N)$ and is defined as follows $\Delta(N) = P(E|T')/P(E|T)$.
The main innovation of the model in [27] is the possibility of adding at each step the best relation $N = \{R_{i,j}\}$ as well as $R_{i,j}$ with all the relations induced from $R_{i,j}$, i.e., $N = \{R_{i,j}\} \cup I(R_{i,j})$ where $I(R_{i,j})$ are the relations induced using the existing taxonomy and $R_{i,j}$. Given the taxonomy T and the relation $R_{i,j}$, the set $I(R_{i,j})$ contains $R_{i,k}$ if $R_{j,k}$ is in T and contains $R_{k,j}$ if $R_{k,i}$ is in T. [1]

We will experiment with our feature selection methodology in two different models:

flat: at each iteration step, a single relation is added, i.e. $\widehat{R}_{i,j} = \arg\max_{R_{i,j}} \Delta(R_{i,j})$

inductive: at each iteration step, a set of relations is added, i.e. $I(\widehat{R}_{i,j})$ where $\widehat{R}_{i,j} = \arg\max_{R_{i,j}} \Delta(I(R_{i,j}))$.

The last important fact is that it is possible to demonstrate that

$$\begin{aligned} \Delta(R_{i,j}) &= k \cdot \frac{P(R_{i,j} \in T|\overrightarrow{e}_{i,j})}{1 - P(R_{i,j} \in T|\overrightarrow{e}_{i,j})} = \\ &= k \cdot odds(R_{i,j}) \qquad (2) \end{aligned}$$

where k is a constant (see [27]) that will be neglected in the maximization process. This last equation gives the possibility of using the logistic regression as it is. In the next sections we will see how SVD and the related feature selection can be used to compute the odds.

4.2 Exploiting SVD in Logistic Regression

We here show that the $odds(R_{i,j})$ in eq. 2 can be computed with logistic regression (Sec. 4.2.1). We then describe how we can compute logistic regression using a particular pseudo-inverse matrix (Sec. 4.2.2). Finally, we show that approximated pseudo-inverse matrices can be computed using SVD (Sec. 4.2.3).

4.2.1 Logistic Regression

Logistic Regression [6] is a particular type of statistical model for relating responses Y to linear combinations of predictor variables X. It is a specific kind of Generalized Linear Model (see [20]) where its function is the *logit function* and the dependent variable Y is a *binary* or *dichotomic* variable which has a Bernoulli distribution. The dependent variable Y takes value 0 or 1. The probability that Y has value 1 is function of the regressors $x = (1, x_1, ..., x_k)$.

The probabilistic taxonomy learner model introduced in the previous section falls in the category of probabilistic models where the logistic regression can be applied as $R_{i,j} \in T$ is the binary dependent variable and $\overrightarrow{e}_{i,j}$ is the vector of its regressors. In the rest of the section we will see how the *odds*, i.e., the multiplicative change, can be computed.

We start from formally describing the Logistic Regression Model. Given the two stochastic variables Y and X, we can define as p the probability of Y to be 1 given that X=x, i.e.$p = P(Y = 1|X = x)$ The distribution of the variable Y is a Bernoulli distribution. Given the definition of the $logit(p)$ as $logit(p) = \ln\left(\frac{p}{1-p}\right)$ and given the fact that Y is a Bernoulli distribution,

[1] For example: given T and $R_{dog,animal}$ if $R_{animal,organism} \in T$ then $I(R_{dog,animal})$ contains $R_{dog,organism}$.
Moreover given T and $R_{bird,beast}$ if $R_{turkey,beast} \in T$ then $I(R_{bird,beast})$ contains $R_{turkey,bird}$.

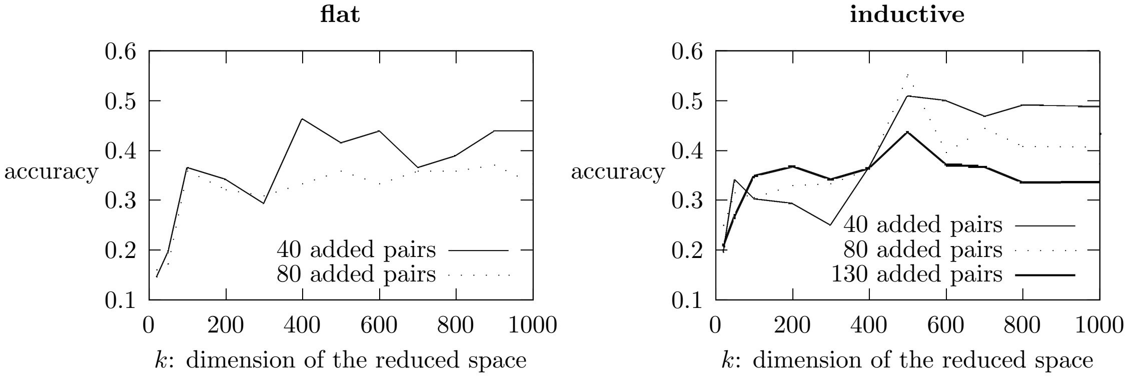

Fig. 1: *Accuracy over different cuts with SVD of the feature space*

the logistic regression foresees that the logit is a linear combination of the values of the regressors, i.e.,

$$logit(p) = \beta_0 + \beta_1 x_1 + ... + \beta_k x_k \tag{3}$$

where $\beta_0, \beta_1, ..., \beta_k$ are called *regression coefficients* of the variables $x_1, ..., x_k$ respectively.

It is obviously trivial to determine the $odds(R_{i,j})$ related to the multiplicative change of the probabilistic taxonomy model. The *odds*, the ratio between the positive and the negative event, can be determined as follows:

$$odds(R_{i,j}) = \frac{P(R_{i,j} \in T|\overrightarrow{e}_{i,j})}{1-P(R_{i,j} \in T|\overrightarrow{e}_{i,j})} = \exp(\beta_0 + \overrightarrow{e}^T_{i,j}\beta) \tag{4}$$

4.2.2 Estimating Coefficients with Pseudoinverse

The remaining problem is how to estimate the regression coefficients. This estimation is done using the maximal likelihood estimation to prepare a set of linear equations using the above *logit* definition and, then, solving a linear problem. This will give us the possibility of introducing the necessity of determining a pseudo-inverse matrix where we will use the singular value decomposition and its natural possibility of performing feature selection. Once we have the regression coefficients, we have the possibility of estimating a probability $P(R_{i,j} \in T|\overrightarrow{e}_{i,j})$ given any configuration of the values of the regressors $\overrightarrow{e}_{i,j}$, i.e., the observed values of the features. Let assume we have a multiset O of observations extracted from $Y \times E$ where $Y \in \{0, 1\}$ and we know that some of them are positive observations (i.e., $Y = 1$) and some of them are negative observations (i.e., $Y = 0$). For each pair, the relative configuration $\overrightarrow{e}_l \in E$ appears at least once in O and can be determined using the maximal likelihood estimation $P(Y = 1|\overrightarrow{e}_l)$. Then, from the equation of the logit (Eq. 3), we have a linear equation system, i.e.:

$$\overrightarrow{logit(p)} = Q\beta \tag{5}$$

where Q is a matrix that includes a constant column of 1, necessary for the β_0 of the linear combination of the values of the regression. Moreover it includes the set of evidences, i.e. $Q = (1, \overrightarrow{e}_1 ... \overrightarrow{e}_m)$.

The set of equations in Eq. 5 are a particular case multiple linear regression [2]. As Q is a rectangular and singular matrix, the system (Eq.5) has no solution. This problem can be solved by the **Moore-Penrose pseudoinverse** Q^+ [23]. Then, we determine the regressors as $\widehat{\beta} = Q^+\overrightarrow{logit(p)}$.

4.2.3 Computing Pseudoinverse with SVD

We finally reached the point where it is possible to explain our idea that is naturally using singular value decomposition (SVD) as feature selection in a probabilistic taxonomy learner. In previous sections we described how the probabilities of the taxonomy learner can be estimated using logistic regressions and we concluded that a way to determine the regression coefficients β is computing the **Moore-Penrose pseudoinverse** Q^+. It is possible to compute the **Moore-Penrose pseudoinverse** using the SVD in the following way [23]. Given an SVD decomposition of the matrix $Q = U\Sigma V^T$ the pseudo-inverse matrix is:

$$Q^+ = V\Sigma^+U^T \tag{6}$$

The diagonal matrix Σ^+ is a matrix $r \times r$ obtained calculating the reciprocals of the singular value of Σ.

We have now our opportunity of using SVD as natural feature selector as we can compute different approximations of the pseudo-inverse matrix. The algorithm for computing SVD is iterative (Sec. 3). The firstly derived dimensions are those with higher singular value. We can then decide how many dimensions we want to use. The first k dimensions are more informative than the $k + 1$. We can consider different k in order to obtain different SVD as approximations of the original matrix (Eq. 1). We can define different approximations of the inverse matrix Q^+ as Q^+_k, i.e.:

$$Q^+_k = V_{n\times k}\Sigma^+_{k\times k}U^T_{k\times m}$$

5 Experimental Evaluation

In this section, we want to empirically explore whether our use of SVD feature selection positively affects performances of the probabilistic taxonomy learner. The

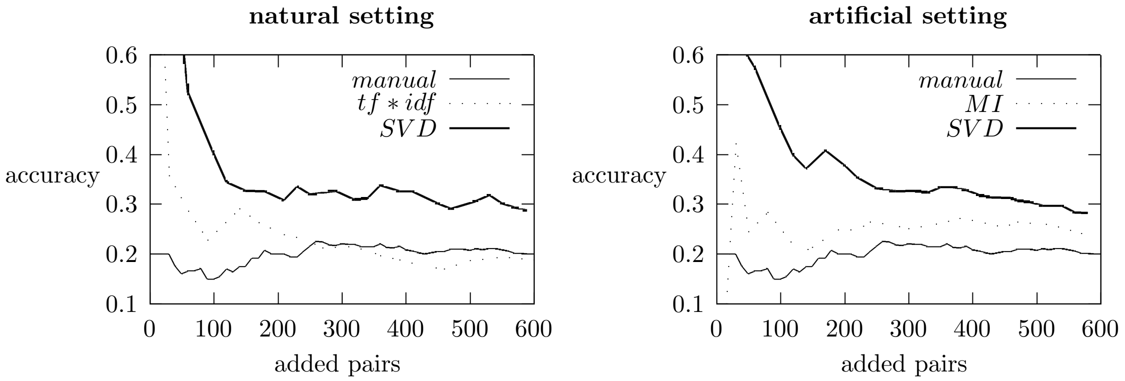

Fig. 2: *Comparison of different feature selection models*

best way of determining how a taxonomy learner is performing is to see if it can replicate an existing ”taxonomy“. We will experiment with the attempt of replicating a portion of WordNet [17]. In the experiments, we will address two issues: determining to what extent SVD feature selection affect performances of the taxonomy learner and determining if, for the probabilistic taxonomy learner, SVD is better than other simpler models for supervised and unsupervised feature selection. We will explore the effects on both the **flat** and the **inductive** probabilistic taxonomy learner.

In the rest of the section we will describe: the experimental set-up (Sec. 5.1) and the results of the experiments in term of performance (Sec. 5.2).

5.1 Experimental Set-up

To completely define the experiments we need to describe some issues: how we defined the taxonomy to replicate, which corpus we have used to extract evidences for pairs of words, which feature space we used, and, finally, the feature selection models we compared against. As target taxonomy we selected a portion of WordNet[2] [17]. Namely, we started from the 44 concrete nouns divided in 3 classes: animal, artifact, and vegetable. For each word w, we selected the synset s_w that is compliant with the class it belongs to. We then obtained a set S of synsets. We then expanded the set to S' adding the siblings (i.e., the coordinate terms) for each synset in S. The set S' contains 265 coordinate terms plus the 44 original concrete nouns. For each element in S we collected its hypernyms, obtaining the set H. We then removed from the set H the 4 topmosts: *entity*, *unit*, *object*, and *whole*. The set H contains 77 hypernyms. For the purpose of the experiments we both derived from the previous sets a taxonomy T and produced a set of negative examples $\overline{T}$. The two sets have been obtained as follows. The taxonomy T is the portion of WordNet implied by $O = H \cup S'$, i.e. T contains all the $(s, h) \in O \times O$ that are in WordNet and $\overline{T}$ contains all the $(s, h) \in O \times O$ that are not in WordNet. We have 5108 positive pairs in T and 52892 negative pairs in $\overline{T}$.

We then produced two experimental settings: a *natural* and an *artificial* one. In the *natural setting* we used only positive pairs in the training set. This is the natural situation when augmenting existing taxonomies. Only positive word pairs can be derived from existing taxonomies. Yet, negative pairs cannot. In the *artificial setting* we used both positive and negative examples.

To obtain the training and the testing sets, we randomly divided the set $T \cup \overline{T}$ in two parts T_{tr} and T_{ts}, respectively, of 70% and 30% of the original $T \cup \overline{T}$.

As corpus we used ukWaC [9]. This is a web extracted corpus of about 2700000 web pages containing more than 2 billion words. The corpus contains documents of different topics such as web, computers, education, public sphere, etc.. It has been largely demonstrated that the web documents are good models for natural language [14].

As the focus of the paper is the analysis of the effect of the SVD feature selection, we used as feature spaces both n-grams and bag-of-words. Out of the $T \cup \overline{T}$, we selected only those pairs that appeared at a distance of at most 3 tokens. Using this 3 tokens, we generated two spaces: (1) *bag-of-word* and (2) the bigram space that contains bigrams and monograms. For the purpose of this experiment, we used a reduced stop list as classical stop words as punctuation, parenthesis, the verb *to be* are very relevant in the context of features for learning a taxonomy.

Finally, we want define the feature selection models we compared against. As *unsupervised feature selection models* we used the term frequency times the inverse document frequency (*tf*idf*). Instances $\overrightarrow{e}$ have the role of the documents. As *supervised feature selection models* we used the mutual information (*mi*). For all the feature selection models, we selected the first k features. Finally, we used a manual feature selection model based on the Heart's patterns [13]. In this model that we call *manual*, we used as features only the classical Hearst's patterns.

5.2 Results

In the first set of experiments we want to focus on the issue whether or not performances of the proba-

[2] We used the version 3.0

bilistic taxonomy learner is positively affected by the proposed feature selection model based on the singular value decomposition. We then determined the performance with respect to different values of k. This latter represents the number of surviving dimensions where the pseudo-inverse is computed. The features of this experiment are unigrams derived from a 3-sized-window. Punctuation has been considered. Figures 1 plots the accuracy of the probabilistic learner with respect to the size of the feature set, i.e. the number k of single values considered for computing the pseudo-inverse matrix. To determine if the effect of the feature selection is preserved during the iteration of the local search algorithm, we report curves at different sizes of the set of added pairs. Curves are reported for both the *flat* model and the *inductive* model. The *flat* algorithm adds one pair at each iteration. Then, we reported curves for 40 and 80 added pairs. The curves show that accuracy doesn't increase after a dimension of k=400. For the *inductive* model we report the accuracies for around 40, 80, 130 added pairs. The optimal dimension of the feature space seems to be around 500 as after that value performances decrease or stay stable. SVD feature selection has then a positive effect for both the *flat* and the *inductive* probabilistic taxonomy learners. This has beneficial effects both on the performances and on the computation time.

In the second set of experiments we want to determine whether or not SVD feature selection for the probabilistic taxonomy learner behaves better than other feature selection models. We then fixed k to 600 both for the SVD selection model and for the other feature selection models. In this experiments, the original feature space is the bigram space. Figure 2 shows results. Curves report accuracies of the different models after n added pairs. In the *natural setting*, we compared our model against the $tf * idf$ and the *manual feature selection*. Our SVD model outperforms both models of feature selection. The same happened against mutual information (MI) in the *artificial setting*. Our SVD way of selecting features seems to be very effective.

6 Conclusions and Future Work

We presented a model to naturally introduce SVD feature selection in a probabilistic taxonomy learner. The method is effective as allows the designing of better probabilistic taxonomy learners. We still need to explore whether or not the positive effect of SVD feature selection is preserved in more complex feature spaces such as syntactic feature spaces as those used in [27].

References

[1] E. Agirre and G. Rigau. Word sense disambiguation using conceptual density. In *Proc. of the 16th COLING*, pages 16–22, Morristown, NJ, USA, 1996. ACL.

[2] D. Caron, W. Hospital, and P. N. Corey. Variance estimation of linear regression coefficients in complex sampling situation. *Sampling Error: Methodology, Software and Application*, pages 688–694, 1988.

[3] P. Cimiano, A. Hotho, and S. Staab. Learning concept hierarchies from text corpora using formal concept analysis. *JAIR*, 24:305–339, 2005.

[4] P. Clark, C. Fellbaum, and J. Hobbs. Using and extending wordnet to support question-answering. In *Proc. 4th GWC*, 2008.

[5] C. Corley and R. Mihalcea. Measuring the semantic similarity of texts. In *Proc. of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, pages 13–18, Ann Arbor, Michigan, June 2005. ACL.

[6] D. R. Cox. The regression analysis of binary sequences. *JRSS,B*, 20(2):215–242, 1958.

[7] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. L, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407, 1990.

[8] I. S. Dhillon, S. Mallela, I. Guyon, and A. Elisseeff. A divisive information-theoretic feature clustering algorithm for text classification. *Journal of Machine Learning Research*, 3:2003, 2003.

[9] A. Ferraresi, E. Zanchetta, M. Baroni, and S. Bernardini. Introducing and evaluating ukwac, a very large web-derived corpus of english. In *Proc. of the WAC4 Workshop at LREC 2008*, Marrakesh, Morocco, 2008.

[10] G. Golub and W. Kahan. Calculating the singular values and pseudo-inverse of a matrix. *J Soc Ind Appl Math B Num Anal*, 2(2):205–224, 1965.

[11] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *JMLR*, 3:1157–1182, March 2003.

[12] Z. Harris. Distributional structure. In J. J. Katz and J. A. Fodor, editors, *The Philosophy of Linguistics*, New York, 1964. Oxford University Press.

[13] M. A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proc. of the 15th CoLing*, Nantes, France, 1992.

[14] M. Lapata and F. Keller. The web as a baseline: Evaluating the performance of unsupervised web-based models for a range of nlp tasks. In *Proc. of the HLT-NAACL*, Boston, MA, 2004.

[15] B. Liu. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*. Data-Centric Systems and Applications. Springer, 2007.

[16] A. Medche. *Ontology Learning for the Semantic Web*, volume 665 of *Engineering and Computer Science*. Kluwer International, 2002.

[17] G. A. Miller. WordNet: A lexical database for English. *Communications of the ACM*, 38(11):39–41, Nov. 1995.

[18] E. Morin. *Extraction de liens sémantiques entre termes à partir de corpus de textes techniques*. PhD thesis, Univesité de Nantes, Faculté des Sciences et de Techniques, 1999.

[19] R. Navigli and P. Velardi. Learning domain ontologies from document warehouses and dedicated web sites. *Comput. Linguist.*, 30(2):151–179, 2004.

[20] J. A. Nelder and R. W. M. Wedderburn. Generalized linear models. *JRSS,A*, 135(3):370–384, 1972.

[21] P. Pantel and M. Pennacchiotti. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proc. of the 21st COLING and 44th Annual Meeting of the ACL*, pages 113–120, Sydney, Australia, July 2006. ACL.

[22] V. Pekar and S. Staab. Taxonomy learning: factoring the structure of a taxonomy into a semantic classification decision. *Proc. of the 19th COLING*, 2:786–792, 2002.

[23] R. Penrose. A generalized inverse for matrices. In *Proc. Cambridge Philosophical Society*, 1955.

[24] D. Ravichandran and E. Hovy. Learning surface text patterns for a question answering system. In *Proceedings of the 40th ACL Meeting*, Philadelphia, Pennsilvania, 2002.

[25] P. Resnik. *Selection and Information: A Class-Based Approach to Lexical Relationships*. PhD thesis, Dept. of Computer and Information Science, Univ. of Pennsylvania, 1993.

[26] H. R. Robison. Computer-detectable semantic structures. *Information Storage and Retrieval*, 6(3):273–288, 1970.

[27] R. Snow, D. Jurafsky, and A. Y. Ng. Semantic taxonomy induction from heterogenous evidence. In *ACL*, pages 801–808, 2006.

[28] I. Szpektor, H. Tanev, I. Dagan, and B. Coppola. Scaling web-based acquisition of entailment relations. In *Proc. of the EMNLP*, Barcellona, Spain, 2004.

[29] A. Toumouth, A. Lehireche, D. Widdows, and M. Malki. Adapting wordnet to the medical domain using lexicosyntactic patterns in the ohsumed corpus. In *Proc. of the AICCSA*, pages 1029–1036, Washington, DC, USA, 2006. IEEE Computer Society.