SVD Feature Selection for Probabilistic Taxonomy Learning

Fallucchi Francesca
Disp, University "Tor Vergata"
Rome, Italy
fallucchi@info.uniroma2.it

Fabio Massimo Zanzotto
Disp, University "Tor Vergata"
Rome, Italy
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 are extremely important knowledge repositories in a variety of applications for natural language processing and knowledge representation. Yet, manually built taxonomies such as WordNet (Miller, 1995) often lack in coverage when used in specific knowledge domains. Automatically creating or extending taxonomies for specific domains is then a very interesting area of research (O'Sullivan et al., 1995; Magnini and Speranza, 2001; Snow et al., 2006). Automatic methods for learning taxonomies from corpora often use distributional hypothesis (Harris, 1964) and exploit some induced lexical-syntactic patterns (Hearst, 1992; Pantel and Pennacchiotti, 2006). In these models, within a very large set, candidate word pairs are selected as new word pairs in hyperonymy and added to an existing taxonomy. Candidate pairs are represented in some feature space. Often, these feature spaces are huge and, then, models may take into consideration noisy features.

In machine learning, feature selection has been often used to reduce the dimensions in huge feature spaces. This has many advantages, e.g., reducing the computational cost and improving performances by removing noisy features (Guyon and Elisseeff, 2003).

In this paper, we propose a novel way to include unsupervised feature selection methods in probabilistic taxonomy learning models. Given the probabilistic taxonomy learning model introduced by (Snow et al., 2006), we leverage on the computation of logistic regression to exploit singular value decomposition (SVD) as unsupervised feature selection. SVD is used to compute the pseudo-inverse matrix needed in logistic regression.

To describe our idea, we firstly review how SVD can be used as unsupervised feature selection (Sec. 2). In Section 3 we then describe the probabilistic taxonomy learning model introduced by (Snow et al., 2006). We will then shortly review the logistic regression used to compute the taxonomy learning model to describe where SVD can be naturally used. We will describe our experiments in Sec. 4. Finally, we will draw some conclusions and describe our future work (Sec. 5).

2 Unsupervised feature selection with Singular Value Decomposition

Singular value decomposition (SVD) is one of the possible factorization of a rectangular matrix that has been largely used in information retrieval for reducing the dimension of the document vector space (Deerwester et al., 1990).

The decomposition can be defined as follows. 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 two matrices U and V are unitary, i.e., $U^TU = I$ and $V^TV = I$. 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.

A first trivial way of using SVD as unsupervised feature reduction is the following. Given E as set

of training examples represented in a feature space of n features, we can observe it as a matrix, i.e. a sequence of examples $E = (\overrightarrow{e_1}...\overrightarrow{e_m})$. With SVD, the $n \times m$ matrix E can be factorized as $E = U\Sigma V^T$. This factorization implies we can focus the learning problem on a new space using the transformation provided by the matrix U. This new space is represented by the matrix:

$$E' = U^T E = \Sigma V^T \quad (1)$$

where each example is represented with r new features. Each new feature is obtained as a linear combination of the original features, i.e. each feature vector $\overrightarrow{e_l}$ can be seen as a new feature vector $\overrightarrow{e_l}' = U^T\overrightarrow{e_l}$. When the target feature space is big whereas the cardinality of the training set is small, i.e., $n >> m$, the application of SVD results in a reduction of the original feature space as the rank r of the matrix E is $r \leq min(n, m)$.

A more interesting way of using SVD as unsupervised feature selection model is to exploit its approximated computations, i.e. :

$$A \approx A_k = U_{m\times k}\Sigma_{k\times k}V^T_{k\times n} \quad (2)$$

where k is smaller than the rank r. The computation algorithm (Golub and Kahan, 1965) is allowed 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 dimension and the value of the singular value. 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 (Liu, 2007)). These dimensions can not be used as discriminative features in learning algorithms. The possibility of computing the approximated version of the matrix 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.

As feature selection model, SVD is unsupervised in the sense that the feature selection is done without taking into account the final classes of the training examples. This is not always the case, feature selection models such as those based on Information Gain largely use the final classes of training examples. SVD as feature selection is independent from the classification problem.

3 Probabilistic Taxonomy Learning and SVD feature selection

Recently, Snow et al. (2006) introduced a probabilistic model for learning taxonomies form corpora. This probabilistic formulation exploits the two well known hypotheses: the distributional hypothesis (Harris, 1964) and the exploitation of the lexico-syntactic patterns as in (Robison, 1970; Hearst, 1992). Yet, in this formulation, we can positively and naturally introduce our use of SVD as feature selection model.

In the rest of this section we will firstly introduce the probabilistic model (Sec. 3.1) and, then, we will describe how SVD is used as feature selector in the logistic regression that estimates the probabilities of the model. To describe this part we need to go in depth into the definition of the logistic regression (Sec. 3.2) and the way of estimating the regression coefficients (Sec. 3.3). This will open the possibility of describing how we exploit SVD (Sec. 3.4)

3.1 Probabilistic model

In the probabilistic formulation (Snow et al., 2006), the task of learning taxonomies from a corpus is seen as a probability maximization 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 $animal$. The main innovation of this probabilistic method is the ability of taking into account in a single probability the information coming from the corpus and an existing 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. Evidences is a feature vector associated with a pair (i, j). For examples, 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 (Hearst, 1992)).

Given a set of evidences E over all the relevant word pairs, in (Snow et al., 2006), the probabilistic taxonomy learning task is defined as the problem of finding the taxonomy $\widehat{T}$ that maximizes the

probability of having the evidences E, i.e.:

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

In (Snow et al., 2006), this maximization problem is solved with a local search. What is maximized at each step is the increase of the probability $P(E|T)$ of the taxonomy when the taxonomy changes from T to $T' = T \cup N$ where N are the relations added at each step. This increase of probabilities is defined as multiplicative change $\Delta(N)$ as follows:

$$\Delta(N) = P(E|T')/P(E|T) \quad (3)$$

The main innovation of the model in (Snow et al., 2006) is the possibility of adding at each step the best relation $N = \{R_{i,j}\}$ as well as $N = I(R_{i,j})$ that is $R_{i,j}$ with all the relations by the existing taxonomy. We will then experiment with our feature selection methodology in the 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

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

where k is a constant (see (Snow et al., 2006)) 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.

3.2 Logistic Regression

Logistic Regression (Cox, 1958) 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 (Nelder and Wedderburn, 1972)) where its function is the *logit function* and the independent variable Y is a *binary* or *dicothomic* 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, i.e.:

$$Y \sim Bernoulli(p)$$

Given the definition of the $logit(p)$ as:

$$logit(p) = \ln\left(\frac{p}{1-p}\right) \quad (4)$$

and given the fact that Y is a Bernoulli distribution, the logistic regression foresees that the logit is a linear combination of the values of the regressors, i.e.,

$$logit(p) = \beta_0 + \beta_1 x_1 + ... + \beta_k x_k \quad (5)$$

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

Given the regression coefficients, it is possible to compute the probability of a given event where we observe the regressors x to be $Y = 1$ or in our case to belong to the taxonomy. This probability can be computed as follows:

$$p(x) = \frac{\exp(\beta_0 + \beta_1 x_1 + ... + \beta_k x_k)}{1 + \exp(\beta_0 + \beta_1 x_1 + ... + \beta_k x_k)}$$

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

$$odds(R_{i,j}) = \frac{P(R_{i,j} \in T|\overrightarrow{e}_{i,j})}{1 - P(R_{i,j} \in T|\overrightarrow{e}_{i,j})} \quad (6)$$

Then, it is strictly related with the logit, i.e.:

$$odds(R_{i,j}) = \exp(\beta_0 + \overrightarrow{e}^{T}_{i,j}\beta) \quad (7)$$

The relationship between the possible values of the probability, odds and logit is show in the Table 1.

Probability	*Odds*	*Logit*
$0 \leq p < 0.5$	$[0, 1)$	$(-\infty, 0]$
$0.5 < p \leq 1$	$[1, \infty)$	$[0, \infty)$

Table 1: Relationship between probability, odds and logit

3.3 Estimating Regression Coefficients

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 assigning 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. For sake of simplicity we will hereafter refer to $\overrightarrow{e}_{i,j}$ as $\overrightarrow{e}_l$.

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 pairs the relative configuration $\overrightarrow{e}_l \in E$ that appeared at least once in O, we can determine using the maximal likelihood estimation $P(Y = 1|\overrightarrow{e}_l)$. Then, from the equation of the logit (Eq. 5), we have a linear equation system, i.e.:

$$\overrightarrow{logit(p)} = Q\beta \tag{8}$$

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 transpose of the evidence matrix, i.e. $E = (\overrightarrow{e}_1 ... \overrightarrow{e}_m)$. Therefore the matrix will be:

$$Q = \begin{pmatrix} 1 & e_{11} & e_{12} & \cdots & e_{1n} \\ 1 & e_{21} & e_{22} & \cdots & e_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & e_{m1} & e_{m2} & \cdots & e_{mn} \end{pmatrix}$$

The set of equations in Eq. 8 can be solved using multiple linear regression.

In their general form, the equations of multiple linear regression may be written as (Caron et al., 1988):

$$y = X\beta + \varepsilon$$

where:

- y is a column vector $n \times 1$ that includes the observed values of the dependent variables $Y_1, ..., Y_k$;
- X is a matrix $n \times m$ of the values of the regressors that we have observed;
- β is a column vector $m \times 1$ of the regression coefficients;
- ε is a column vector including the stochastic components that have not been observed and that will not be considered later.

In the case X is a rectangular and singular matrix, the system $y = X\beta$ has not a solution. Yet, it is possible to use the principle of the Least Square Estimation. This principle determines the solution β that minimize the residual norm, i.e.:

$$\widehat{\beta} = \arg\min \|X\beta - y\|^2 \tag{9}$$

This problem can be solved by the **Moore-Penrose pseudoinverse** X^+ (Penrose, 1955). Then, the final equation to determine the β is

$$\widehat{\beta} = X^+ y$$

It is important to remark that if the inverse matrix exist $X^+ = X^{-1}$ and that X^+X and XX^+ are symmetric.

For our case, the following equation is valid:

$$\widehat{\beta} = Q^+ \overrightarrow{logit(p)}$$

3.4 Computing Pseudoinverse Matrix with SVD Analysis

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 the 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 (Penrose, 1955). Given an SVD decomposition of the

matrix $Q = U\Sigma V^T$ the pseudo-inverse matrix that minimizes the Eq. 9 is:

$$Q^+ = V\Sigma^+ U^T \quad (10)$$

The diagonal matrix Σ^+ is a matrix $r \times r$ obtained first transposing Σ and then calculating the reciprocals of the singular value of Σ. So the diagonal elements of the Σ^+ are $\frac{1}{\delta_1}, \frac{1}{\delta_2}, ..., \frac{1}{\delta_r}$.

We have now our opportunity of using SVD as natural feature selector as we can compute different approximations of the pseudo-inverse matrix. As we saw in Sec. 2, the algorithm for computing the singular value decomposition can be stopped a different dimensions. We called k the number of dimensions. As we can obtain different SVD as approximations of the original matrix (Eq. 2), we can define different approximations of :

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

In our experiments we will use different values of k to explore the benefits of SVD as feature selector.

4 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 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 (Miller, 1995). In the experiments, we will address two issues: 1) determining to what extent SVD feature selection affect performances of the taxonomy learner; 2) determining if SVD as unsupervised feature selection is better for the task than some simpler model for taxonomy learning. We will explore the effects on both the **flat** and the **inductive** probabilistic taxonomy learner.

The rest of the section is organized as follows. In Sec. 4.1 we will describe the experimental setup in terms of: how we selected the portion of WordNet, the description of the corpus used to extract evidences, a description of the feature space we used, and, finally, the description of a baseline models for taxonomy learning we have used. In Sec. 4.2 we will present the results of the experiments in term of performance.

4.1 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 baseline model we compared our feature selection model against.

As target taxonomy we selected a portion of WordNet[1] (Miller, 1995). Namely, we started from the 44 concrete nouns listed in (McRae et al., 2005) and divided in 3 classes: animal, artifact, and vegetable. For sake of comprehension, this set is described in Tab. 2. 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 (see Tab. 2). 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 hyperonym, obtaining the set H. We then removed from the set H the 4 topmosts: *entity*, *unit*, *object*, and *whole*. The set H contains 77 hyperonyms. 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. On the contrary, $\overline{T}$ contains all the $(s, h) \in O \times O$ that are not in WordNet. We then have 5108 positive pairs in T and 52892 negative pairs in $\overline{T}$.

We then split the set $T \cup \overline{T}$ in two parts, training and testing. As we want to see if it is possible to attach the set S' to the right hyperonym, the split has been done as follows. We randomly divided the set S' in two parts S_{tr} and S_{ts}, respectively, of 70% and 30% of the original S'. We then selected as training T_{tr} all the pairs in T containing a synset in S_{tr} and as testing set T_{ts} those pairs of T containing a synset of S_{ts}. For the probabilistic model, T_{tr} is the initial taxonomy whereas $T_{ts} \cup \overline{T}$ is the unknown set.

As corpus we used the *English Web as Corpus* (ukWaC) (Ferraresi et al., 2008). 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

[1] We used the version 3.0

	Concrete nouns	*Clas*	*Sense*		*Concrete nouns*	*Clas*	*Sense*
1	banana	Vegetable	1	23	boat	Artifact	0
2	bottle	Artifact	0	24	bowl	Artifact	0
3	car	Artifact	0	25	cat	Animal	0
4	cherry	Vegetable	2	26	chicken	Animal	1
5	chisel	Artifact	0	27	corn	Vegetable	2
6	cow	Animal	0	28	cup	Artifact	0
7	dog	Animal	0	29	duck	Animal	0
8	eagle	Animal	0	30	elephant	Animal	0
9	hammer	Artifact	1	31	helicopter	Artifact	0
10	kettle	Artifact	0	32	knife	Artifact	0
11	lettuce	Vegetable	2	33	lion	Animal	0
12	motorcycle	Artifact	0	34	mushroom	Vegetable	4
13	onion	Vegetable	2	35	owl	Animal	0
14	peacock	Animal	1	36	pear	Vegetable	0
15	pen	Artifact	0	37	pencil	Artifact	0
16	penguin	Animal	0	38	pig	Animal	0
17	pineapple	Vegetable	1	39	potato	Vegetable	2
18	rocket	Artifact	0	40	scissors	Artifact	0
19	screwdriver	Artifact	0	41	ship	Artifact	0
20	snail	Animal	0	42	spoon	Artifact	0
21	swan	Animal	0	43	telephone	Artifact	1
22	truck	Artifact	0	44	turtle	Animal	1

Table 2: Concrete nouns, Classes and senses selected in WordNet

are good models for natural language (Lapata and Keller, 2004).

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 these 3 tokens, we generated three spaces: (1) 1-gram that contains monograms, (2) 2-gram that contains monograms and bigrams, and (3) the 3-gram space that contains monograms, bigrams, and trigrams. 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 to describe our *baseline model* for taxonomy learning. This model only contains Heart's patterns (Hearst, 1992) as features. The feature value is the point-wise mutual information. These features are in some sense the best features for the task as these have been manually selected after a process of corpus analysis. These baseline features are included in our 3-gram model. We can then compare our best models with this baseline features in order to see if our SVD feature selection model outperforms manual feature selection.

4.2 Results

In the first set of experiments we want to focus on the issue whether or not performances of the probabilistic 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. Then, it represents the number of features the model adopts. We performed this first set of experiments in the 1-gram feature space. Punctuation has been considered. Figure 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

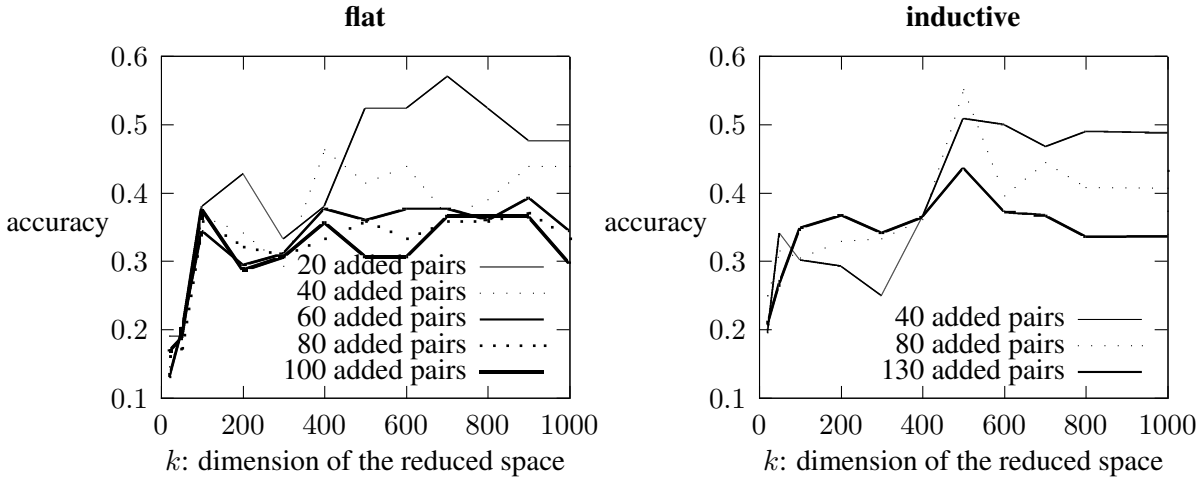

Figure 1: Accuracy over different cuts of the feature space

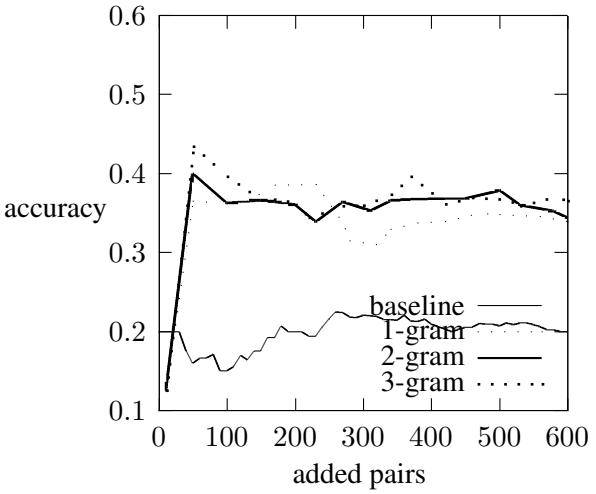

Figure 2: Comparison of different feature spaces with k=400

reported for both the *flat* model and the *inductive* model. The *flat* algorithm adds one pair at each iteration. Then, we reported curves for each 20 added pairs. Each curve shows that accuracy does not increase after a dimension of k=700. This size of the space is necessary only for the first 20 added pairs. Accuracy keeps increasing to k=700 and then decreases. When we add more pairs, the optimal size of the space is around k=200. For the *inductive* model we report the accuracies for around 40, 80, 130 added pairs. Here, at each iteration, more than one pair is added. 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 a reduced set of known features. We then fixed the dimension k to 400 and we compared the *baseline model* with different probabilistic models with different feature sets: 1-gram, 2-gram, and 3-gram. We can consider that the trigram model before the cut on its dimensions contains feature subsuming the *baseline model*. Figure 2 shows results. Curves report accuracy after n added pairs. All the probabilistic models outperform the baseline model. As what happened for the first series of experiments (see Fig. 1) more informative spaces such as 3-gram behaves better when the number of

added pairs is small. Performances of the three reduced pairs become similar after 100 added pairs. These experiments show that SVD feature selection has a positive effect on performances as resulting models are always better with respect to the baseline.

5 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 at least two issues. First, we need to determine 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 (Snow et al., 2006). Second, we need to compare the SVD feature selection with other unsupervised feature selection models to determine whether or not this is the best method to use in the case of probabilistic taxonomy learning.

References

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

D. R. Cox. 1958. The regression analysis of binary sequences. *Journal of the Royal Statistical Society. Series B (Methodological)*, 20(2):215–242.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. L, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407.

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

G. Golub and W. Kahan. 1965. Calculating the singular values and pseudo-inverse of a matrix. *Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis*, 2(2):205–224.

Isabelle Guyon and André Elisseeff. 2003. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, March.

Zellig Harris. 1964. Distributional structure. In Jerrold J. Katz and Jerry A. Fodor, editors, *The Philosophy of Linguistics*, New York. Oxford University Press.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 15th International Conference on Computational Linguistics (CoLing-92)*, Nantes, France.

Mirella Lapata and Frank Keller. 2004. The web as a baseline: Evaluating the performance of unsupervised web-based models for a range of nlp tasks. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, Boston, MA.

Bing Liu. 2007. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*. Data-Centric Systems and Applications. Springer.

Bernardo Magnini and Manuela Speranza. 2001. Integrating generic and specialized wordnets. In *In Proceedings of the Euroconference RANLP 2001*, Tzigov Chark, Bulgaria.

K. McRae, G.S. Cree, M.S. Seidenberg, and C. McNorgan. 2005. Semantic feature production norms for a large set of living and nonliving things. pages 547–559, Behavioral Research Methods, Instruments, and Computers.

George A. Miller. 1995. WordNet: A lexical database for English. *Communications of the ACM*, 38(11):39–41, November.

J. A. Nelder and R. W. M. Wedderburn. 1972. Generalized linear models. *Journal of the Royal Statistical Society. Series A (General)*, 135(3):370–384.

Donie O'Sullivan, A. McElligott, and Richard F. E. Sutcliffe. 1995. Augmenting the princeton wordnet with a domain specific ontology. In *Proceedings of the Workshop on Basic Issues in Knowledge Sharing at the 14th International Joint Conference on Artificial Intelligence*. Montreal, Canada.

Patrick Pantel and Marco Pennacchiotti. 2006. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 113–120, Sydney, Australia, July. Association for Computational Linguistics.

R. Penrose. 1955. A generalized inverse for matrices. In *Proc. Cambridge Philosophical Society*.

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

Rion Snow, Daniel Jurafsky, and A. Y. Ng. 2006. Semantic taxonomy induction from heterogenous evidence. In *In ACL*, pages 801–808.