Bayesian embedding of co-occurrence data for query-based visualization

Mohammad Khoshneshin
Department of Management Sciences
The University of Iowa
Iowa City, IA 52241 USA
mohammad-khoshneshin@uiowa.edu

W. Nick Street
Department of Management Sciences
The University of Iowa
Iowa City, IA 52241 USA
nick-street@uiowa.edu

Padmini Srinivasan
Computer Science Department
The University of Iowa
Iowa City, IA 52241 USA
padmini-srinivasan@uiowa.edu

***Abstract*—We propose a generative probabilistic model for visualizing co-occurrence data. In co-occurrence data, there are a number of entities and the data includes the frequency of two entities co-occurring. We propose a Bayesian approach to infer the latent variables. Given the intractability of inference for the posterior distribution, we use approximate inference via variational approaches. The proposed Bayesian approach enables accurate embedding in high-dimensional space which is not useful for visualization. Therefore, we propose a method to embed a filtered number of entities for a query—query-based visualization. Our experiments show that our proposed models outperform co-occurrence data embedding, the state-of-the-art model for visualizing co-occurrence data.**

I. Introduction

Visualization is one of the most important exploratory tools for data analysis and mining. In this paper we propose a fully generative probabilistic model—Bayesian co-occurrence data embedding—to embed co-occurrence data in a Euclidean space as an unsupervised learning approach that can be used for visualization. This model is a Bayesian extension to co-occurrence data embedding (CODE) [7].

Bayesian co-occurrence data embedding (Bayes-CODE) is a generative probabilistic model for embedding co-occurrence data in a Euclidean space. To explain different types of co-occurrence data, we use graph notation. Let $G = (U, E)$ represent an unweighted graph with multiple edges allowed to exist between any two nodes. Each edge depicts a token in the co-occurrence dataset. A token is a single observed occurrence. So if edge or token (i, k)—i and k indexes entities or nodes—occurred 3 times, then v_{ik} (the i^{th}, k^{th} element of co-occurrence matrix V) is 3, and there are 3 edges between nodes i and k in the equivalent graph. The reason we use graph notation is to consider some specific relationships that cannot be represented by the matrix notation. Based on the structure of the graph G, there are two types of structural attributes that describe co-occurrence data. First is whether the graph is directed or undirected. Second is whether the nodes U are homogeneous or heterogeneous.

When the co-occurrence graph is directed, it means that one type of entity is responsible for generating another type, therefore it is logical to define the generative model as a conditional probability. In the case of an undirected graph, the joint probability is more appropriate. In the case of heterogeneous nodes, nodes in U are divided into two groups, U_1 and U_2, and edges can only be defined between nodes of different types; this heterogeneous graph is a bipartite graph. In the case of homogeneous nodes, there is only one type of node.

Based on the categories for the co-occurrence data, four different classes can be defined. In hetero-directed, each token consists of two different entities and one type of entity is responsible for generating the other. The most popular example of this type is text data where a document is responsible for generating words. Therefore, the link direction is from document nodes to word nodes. In hetero-undirected, each token consists of two different entities and both entities are generated simultaneously from a joint distribution. An example of this type is the co-occurrence of image features and keywords. In, homo-directed, each token consists of two entities from the same type and one entity is responsible for generating the other. An example is co-citation data. Entities are scholars who cite other scholars in their research papers. In homo-undirected, each token consists of two entities from the same type and both entities are generated simultaneously from a joint distribution. An example is the co-occurrence between items from market basket data. Each token consists of two items that have been purchased together.

In this paper, we focus on the hetero-directed case. Deriving the model and inference algorithm for other cases will be straightforward based on the method proposed in this paper. Since inference in the proposed model is intractable, we use approximate inference via variational methods.

It is hard to capture the essence of real-world data in two dimensions due to high complexity. On the other hand, visualizing a large number of data is confusing rather than informative. Therefore, a way to present a filtered version of data is useful. To this end, we propose a query-based visualization method. Although we present this algorithm in the context of information retrieval, it can be applied to any query-answering problem for co-occurrence data.

The paper is organized as follows. In Section II, we review the related literature on visualizing co-occurrence data. In Section III, the Bayesian co-occurrence data embedding is presented. In Section IV, the approximate inference derivations are presented where we use variational Bayes approach to learn the posterior parameters. In Section V, we present query-based visualization. In Section VI, the experimental results are presented. We examined Bayes-CODE in the context of visualizing text data where it gives very competitive results. In the last section, we conclude with some comments and future directions.

II. Background

Visualizing co-occurrence data with heterogeneous nodes—generally heterogeneous data—has been rarely studied. Most of the literature concentrates on embedding only one type of data (e.g. multi-dimensional scaling [5]). In the context of text data—which is co-occurrence data with heterogeneous nodes—most visualization approaches usually embed only documents or words via embedding algorithms such as multidimensional scaling [10]. The state-of-the-art algorithm to visualize co-occurrence data with heterogeneous data is co-occurrence data embedding (CODE) [7].

In CODE, all entities are embedded in a unified Euclidean space in a way that closer entities are more correlated. Let X_i (a $1 \times D$ vector) represent the latent variable of entity i and Y_k (a $1 \times D$ vector) represent the latent variable of entity k in a D-dimensional space. In CODE, two approaches are used for fitting the model to the data. In the first approach, the joint distribution is modeled via:

$$P(i,k) = \frac{1}{Z}\tilde{P}(i)\tilde{P}(k)\exp(-\delta_{ik}^2), \tag{1}$$

where $\delta_{ik}^2 = (X_i - Y_k)(X_i - Y_k)^T$ is the squared Euclidean distance between latent variables of i and k and the normalizing factor is $Z = \sum_{ik}\tilde{P}(i)\tilde{P}(k)\exp(-\delta_{ik}^2)$. $\tilde{P}(i)$ and $\tilde{P}(k)$ are the empirical marginal probabilities which model the biasness of entities—some entities occur more frequently compared to others. This model is appropriate for the hetero-undirected case.

The second approach in CODE is modeling the conditional probability instead of the joint probability:

$$P(i|k) = \frac{1}{Z_k}\tilde{P}(i)\exp(-\delta_{ik}^2), \tag{2}$$

where $Z = \sum_i \tilde{P}(i)\exp(-\delta_{ik}^2)$, which is useful in hetero-directed. The latent variables can be found via minimizing the Kullback-Leibler divergence between empirical and model probabilities:

$$P^* = \arg\min D_{KL}[\tilde{P}|P].$$

The main intuition behind CODE is that if two points are related then they should be very close in the latent space. Therefore, the result of embedding can be presented as a visualization of entities. Although the embedding is based on the relationship between entities from different types, we expect the entities from the same type to be close due to the transitivity of distance.

Our proposed model, Bayes-CODE, is similar to CODE for the basic probability model. The main difference, which is our main contribution, is extending CODE to a fully generative probabilistic model for learning co-occurrence data. Instead of the maximum likelihood approach which is very vulnerable to overfitting, we use a Bayesian approach which has some classic advantages such as robustness. Although our generative model outperforms CODE even in a 2-dimensional space, the difference intensifies in higher dimensions. Nevertheless, it is not possible to interpret more than 3 dimensions visually. Our other contribution is proposing a query-based visualization to embed a filtered number of entities. Another contribution of our work is related to how the bias of entities is learned. Our algorithm uses a Bayesian approach, whereas bias parameters were estimated directly from marginal empirical probabilities in CODE.

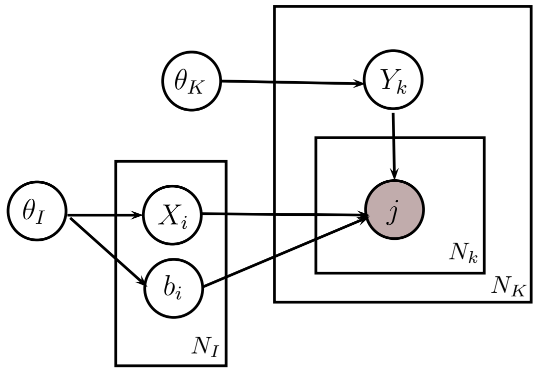

Fig. 1. The graphical model of Bayesian embedding of co-occurrence data.

III. Bayesian embedding of co-occurrence data

In Bayesian co-occurrence data embedding (Bayes-CODE), the relationship between entities is captured by embedding them in a latent variable space. Here, we present Bayes-CODE for the hetero-directed case and deriving the model for other cases is straightforward. Following the same notation from previous sections, i indexes the entities of the first type and k indexes the entities of the second type. Similar to other latent space models, the dimension D of the latent space is an algorithmic input and can be chosen in a Bayesian manner. Here, we treat D as a known parameter. The learned positions of entities in the latent space are denoted by X_i for the entities of the first type and Y_k for the entities of the second type. Furthermore, b_i and b_k represent the biases of entities. By biasness we refer to the situation in which some entities tend to occur more often, such as some words which have higher frequency than others.

In the current model, we assume Gaussian priors on all latent variables. This is an arbitrary choice of distribution and one may assume any other distribution. The relationship between entities is computed via the squared Euclidean distance. The probability model for this data is defined as a conditional probability:

$$P(i|k) = \frac{1}{Z_k}\exp(-\delta(X_i, Y_k) + b_i), \tag{3}$$

where $Z_k = \sum_i \exp(\delta(X_i, Y_k) + b_i)$. Note that we do not include a bias parameter for entity k, since we are conditioning on k and the bias of k has no effect. Even if b_k is inserted into (3), it will be canceled out.

The graphical model of the generative process is shown in Figure 1. First, latent variables are generated for the N_I entities of the first type from the prior parameters θ_I and for the N_K entities of the second type from the prior parameters θ_K. Then for each of the N_k tokens in entity k, an entity of the first type is chosen from N_I possible entities via a multinomial distribution with probabilities from (3). The generative process can be summarized as follows:

1) For each entity i:
   a) Choose entity latent variable $X \sim N(\mu_{0I}, \sigma_{0I}^2 \mathcal{I})$
   b) Choose entity bias variable $b \sim N(\beta_{0I}, \xi_{0I}^2)$
2) For each entity k:
   a) Choose entity latent variable $Y \sim N(\mu_{0K}, \sigma_{0K}^2 \mathcal{I})$
   b) For each token:
      i) Choose $j \sim Multinomial(P(.|k))$

where $\mathcal{I}$ denotes the identity matrix and P is computed from (3).

IV. Approximate Inference

The likelihood of the whole dataset given the assumption that the probability of each token is independent from other tokens given the hidden variables is as follows:

$$P(U|X,Y,b) = \prod_u P(i_u|k_u, X, Y, b) = \prod_{ik} P(i|k, X, Y, b)^{v_{ik}}, \tag{4}$$

where U is the set of all tokens (i.e. the whole dataset), and v_{ik} is the number of times the token (i,k) has occurred. Given (3) and (4), the log-likelihood is as follows:

$$\log P(U|X,Y,b) = -\sum_{ik} v_{ik}\delta(X_i, Y_k) + \sum_i v_{i.} b_i - \sum_k v_{.k} \log \sum_i \exp(-\delta(X_i, Y_k) + b_i), \tag{5}$$

where $v_{i.} = \sum_k v_{ik}$ and $v_{.k} = \sum_i v_{ik}$. To estimate the hidden variables X, Y and b, we can maximize the log-likelihood (5). However, the maximum likelihood approach has problems such as overfitting. A Bayesian approach will result in a more robust solution by giving the posterior distribution of hidden parameters. Given the Bayes rules, the posterior distribution of latent variables given the data is as follows:

$$P(X,Y,b|U) = \frac{P(U|X,Y,b)P(X)P(Y)P(b)}{\int P(U|X',Y',b')P(X')P(Y')P(b')dX'dY'db'}, \tag{6}$$

which is not computable analytically due to the intractability of the integral in the denominator. Therefore, we chose to use variation inference [8] which is a popular algorithm for approximate inference in graphical models.

In variational approximation, instead of finding the true posterior, we estimate a variational distribution for each latent variable. The main idea is minimizing the difference between the true posterior and the surrogate variational distribution so then we can use the variational distribution for making inference about latent variables. If $Q(X,Y,b)$ shows the variational distribution over latent variables, we are interested in minimizing the Kullback-Leibler divergence between the true posterior and its approximation: $KL(Q(X,Y,b)||P(X,Y,b|U)$. This is equivalent to maximizing a lower bound over the marginal probability of data [1]:

$$\log P(U) \geq E_Q[\log P(U|X,Y,b)] - KL(Q(X,Y,b)||P(X,Y,b)), \tag{7}$$

where $E_Q[.]$ is expectation with regard to variational distributions. Here we assume the variational distributions are independent Gaussians with the following parameters: $X_i \sim N(\mu_i, \sigma_i^2\mathcal{I})$, $Y_k \sim N(\mu_k, \sigma_k^2\mathcal{I})$, and $b_i \sim N(\beta_i, \xi_i^2)$, where $\mathcal{I}$ is the identity matrix with dimension D. Note that it is possible to use a covariance matrix instead of $\sigma^2\mathcal{I}$ and the only reason we chose to use independent coordinates is simplicity in optimization. Substituting $P(U|X,Y,b)$ with its value from (5), the lower bound on the probability of data in (7) can be written as:

$$\mathcal{L}(\mu,\beta,\sigma^2,\xi^2) = \sum_{ik} v_{ik} E_Q[-\delta(X_i,Y_k) + b_i] - \sum_k v_{.k} E_Q[\log \sum_i \exp(-\delta(X_i,Y_k) + b_i)] - KL(Q(X)||P(X)) - KL(Q(Y)||P(Y)) - KL(Q(b)||P(b)). \tag{8}$$

Given the Gaussian distribution for priors and variational distributions, all integrals for computing the expectations in (8) are analytically solvable except for the part $\sum_k v_{.k} E_Q[\log \sum_i \exp(\delta(X_i,Y_k) + b_i)]$ which is intractable because of the log-sum-exp format. It is possible to use the concavity of the log function to define an upper bound on the log-sum-exp function:

$$\log a \leq \phi a - \log \phi - 1, \tag{9}$$

where the equality holds iff $\phi = 1/a$. Such an approach has been used in several works in the variational inference context [4], [2], [3]. Therefore, the new bound is as follows:

$$\mathcal{L}(\mu,\beta,\sigma^2,\xi^2) = constant + \sum_{ik} v_{ik} E_Q[\delta(X_i,Y_k) + b_i] - \sum_k v_{.k}\phi_k \sum_i E_Q[\exp(\delta(X_i,Y_k) + b_i)] - KL(Q(X)||P(X)) - KL(Q(Y)||P(Y)) - KL(Q(b)||P(b)), \tag{10}$$

where the constant does not depend on decision variables and we set $\phi_k = [\sum_i E_Q(exp(-\delta(X_i,Y_k) + b_i))]^{-1}$ to tighten the lower bound with regard to the log-sum-exp part.

The only tricky part in (10) is deriving the integral $E_Q[\exp(-\delta^2(X_i,Y_k) + b_i)]$. Let $x_1 \sim N(\mu_1, \sigma_1)$ and $x_2 \sim N(\mu_2, \sigma_2)$, then it can be shown that the following equation holds:

$$E[exp(-(x_1 - x_2)^2)] = \frac{exp(-\frac{(\mu_1-\mu_2)^2}{1+2(\sigma_1^2+\sigma_2^2)})}{\sqrt{1+2(\sigma_1^2+\sigma_2^2))}} \qquad (11)$$

and therefore we have:

$$E_Q[e^{-\delta^2(X_i,Y_k)+b_k}] = \eta_{ik}^D e^{-\eta_{ik}^2\delta^2(\mu_i,\mu_k)+\beta_i+\xi_i^2/2}$$

where $\eta_{ik} = [1 + 2(\sigma_i^2 + \sigma_k^2))]^{-1/2}$. As a result, the lower bound in (10) can be written as follows:

$$\begin{aligned}
&\mathcal{L}(\mu, \beta, \sigma^2, \xi^2) = constant \\
&- \sum_{ik} v_{ik}\delta^2(\mu_i, \mu_k) - (\sum_i v_{i.}\sigma_i^2 + \sum_k v_{.k}\sigma_k^2)D + \sum_i v_{i.}\beta_i \\
&\quad - \sum_k v_{.k}\phi_k \sum_i \eta_{ik}^D \exp(-\eta_{ik}^2\delta^2(\mu_i,\mu_k) + \beta_i + \xi_i^2/2) \\
&\quad - \frac{1}{2}\sum_i[-D\log\sigma_i^2 + D\frac{\sigma_i^2}{\sigma_{0I}^2} + \frac{(\mu_i-\mu_{0I})(\mu_i-\mu_{0I})^T}{\sigma_{0I}^2}] \\
&- \frac{1}{2}\sum_k[-D\log\sigma_k^2 + D\frac{\sigma_k^2}{\sigma_{0K}^2} + \frac{(\mu_k-\mu_{0K})(\mu_k-\mu_{0K})^T}{\sigma_{0K}^2}] \\
&\qquad - \frac{1}{2}\sum_i[-\log\xi_i^2 + \frac{\xi_i^2}{\xi_0^2} + \frac{(\beta_i-\beta_0)^2}{\xi_0^2}]. \qquad (12)
\end{aligned}$$

To find variational values, we need to optimize (12). Since $\sigma^2 \geq 0$, to have an unconstrained optimization problem, we use an auxiliary variable χ and the exponential function in our experiments: $\sigma^2 = \exp(\chi)$. Any unconstrained optimization algorithm can be used to solve (12). In our experiments, we used gradient ascent with multiple random starts.

V. Query-based visualization

Here, we present query-based visualization for information retrieval; however, it can be applied to other dyadic data—see [9] for a similar approach in collaborative filtering.

In query-based visualization (QBV), documents, query words, and the query are embedded in a Euclidean space to help the user in identifying documents of interest. Unfortunately, 2 dimensions is barely enough to capture the complexity of a data, while higher dimensions cannot be interpreted visually. Additionally, representing all data to a user is beyond a person's processing ability. Therefore, visualizing only top-N (N can be specified by user) documents is of interest. These top-N documents can be chosen by an arbitrary retrieval method.

Therefore, a two-phase visualization can be used. First, the data will be embedded in a high-dimensional space and then a group of entities can be chosen via some filtering approach to be re-embedded in a 2-dimensional space by classic algorithms such as multidimensional scaling (MDS) [5]. The second embedding phase is straightforward, since we already have the distances from the first phase which satisfy all requirements for the Euclidean distance and MDS can be applied directly. In an information retrieval context, our proposed approach is embedding words, documents, and the query in a high-dimensional space, and then using the distances in that space, embedding selected objects in a 2-dimensional space via multidimensional scaling.

Another approach may be embedding the filtered data directly into a 2-dimensional space. However, such an approach is not desirable for two reasons. First, since there are few entities in the filtered data, generalization is expected to be poor which deteriorates the result. Second, embedding entities separately for each query is very time consuming and inefficient, especially given the high number of queries in the retrieval systems.

VI. Experiments

In our experiments, we compare CODE and Bayes-CODE in the context of information retrieval. Text data is considered as hetero-directed and so we use a conditional probability model.

We study the goodness of embedding using two evaluation metrics. First is the proximity of an embedded query to the relevant documents. This can be measured with average precision (AP). The definition of AP that we used is averaging the precision of all relevant documents at the point they are retrieved. Let $AP(S, R)$ be the average precision where R is the set of relevant documents and S is the score of a method for ranking all documents for retrieval. Then given an embedded query q, the average precision is $AP(-\delta_{q.}^2, R)$ where $\delta_{q.}^2$ shows the distance of the query to all documents. Second, the proximity of relevant documents is of interest. It is important whether all relevant documents are close to each other compared to other documents. To measure this, average relevant documents proximity (ARDP) is proposed:

$$ARDP = [\sum_c \frac{\sum_{k\in R_c} AP(-\delta_{k.}, R_c\backslash\{k\})}{|R_c| - 1}]/N_C, \qquad (13)$$

where c indexes categories or queries (any group of relevant documents), k indexes documents, R_c is the set of relevant documents in c, and N_C is the total number of categories or queries. ARDP is partially similar to doc-doc measure used in [7]. More precisely, we consider each relevant document as an embedded query and then we compute the AP of retrieving other relevant documents. This metric measures how well we embed data, since relevant documents are expected to be closer and the visualization is better as a result.

In Bayes-CODE, we set the prior mean to 0 for both words and documents, and prior variance to 1 for words and 2 for documents. No fine tunning was done for finding priors. We chose smaller prior variance for words due to the fact that the support for words is often lower compared to the support for documents. There are some words that occur only 2 or 3 times while the number of occurrences of documents (the number of words in them) is usually much higher.

Four datasets were used in our experiments. We used subsets of the TDT-2 and Reuters21578 datasets[1]. In these datasets, there are a number of categories and each document is related to one—the documents in the same category are related. In each dataset, we selected 5 categories so that the number of

[1]http://www.zjucadcg.cn/dengcai/Data/TextData.html

documents in categories is almost equal. Words that occurred fewer than 3 times were excluded. Our subset of TDT-2 includes 8,676 words and 1,584 documents and our subset of Reuters21578 includes 4,711 words and 1203 documents. These two datasets were only used for evaluating based on ARDP.

For the information retrieval task, we used datasets CRAN and MED[2]. The CRAN dataset includes 3,763 words, 1398 documents and 225 queries, and MEDLINE includes 7014 words, 1033 documents, and 30 queries. In both CODE and Bayes-CODE, queries were treated as new documents and mapped into the latent space. Then, the Euclidean distance between queries and documents was used to compute a score for retrieval. These data were used for evaluating based on both AP and ARDP.

Table I presents the result of ARDP for all datasets. We implemented CODE and Bayes-CODE in a 2-dimensional space and then computed the ARDP score. Bayes-CODE outperforms CODE in all 4 datasets (in CRAN the performance is close).

	TDT2	Reuters21579	CRAN	MEDLINE
CODE	.9023	.4582	.0637	.1561
Bayes-CODE	**.9490**	**.5568**	**.0646**	**.1797**

TABLE I
ARDP ON A 2-DIMENSIONAL SPACE (BEST RESULTS ARE BOLD)

For evaluating CODE versus Bayes-CODE in higher dimensions, we used CRAN and MEDLINE. Figures 2 and 3 shows the result for average precision and Figures 4 and 5 shows the result for ARDP, in CRAN and MEDLINE datasets respectively. As expected, the performance of CODE decreases as we increase the number of dimensions. Note that procedures such as tuning are not possible since CODE is an unsupervised algorithm.

	CRAN dataset		MEDLINE dataset	
	AP	ARDP	AP	ARDP
CODE-2	.1591	.1299	.4046	.4648
Bayes-CODE-QBV	**.3231**	**.3498**	**.6145**	**.6384**

TABLE II
QUERY-BASED VISUALIZATION + BAYES-CODE VERSUS CODE WITH 2 DIMENSIONS (BEST RESULTS ARE BOLD)

Finally, we explored query-based visualization using Bayes-CODE. First, we selected top-100 documents for each query using latent semantic indexing [6] which is a successful method in information retrieval. Note that it is possible to use Bayes-CODE for filtering documents directly but here we need a method for both CODE and Bayes-CODE for the sake of comparison. Then, we re-embedded all filtered documents, query words, and the query in a 2-dimensional space via MDS with distances from implementing Bayes-CODE in a 100-dimensional space. We compare the result to the CODE's result in a 2-dimensional space. Table II represents the result. The performance of query-based visualization is dramatically better than CODE.

Figures 6 and 7 represent a typical snapshot of visualizing a query using CODE and query-based visualization using Bayes-CODE respectively, for a specific query in the MEDLINE dataset. Note the distinction between relevant and irrelevant documents in query-based visualization while they are highly mixed in CODE. Additionally, the query is far away from other entities in CODE which makes interpretation difficult. Query words might help user to identify which area in the space is more relevant.

VII. CONCLUSION

In this paper, we developed a Bayesian model based on the state-of-the-art visualization model for co-occurrence—CODE. Our experimental studies reveal the superiority of the Bayesian approach. However, better embedding in higher dimensions is not useful for visualization. Therefore, we proposed a method to embed filtered data from a high-dimensional embedding for a query—query-based visualization which was successful in our experiments.

Query-based visualization can be a basis for an interactive user interface, in which a user receives her recommendations in a visual manner while she can have a general picture of relationships between her query, her keywords, and the top-N relevant documents. Then she can explore the visual space and mark documents accordingly. Also, she might have the option of asking for more documents close to a documents or more words close to a word, and after re-embedding she can have a more accurate picture. Also, using relevance feedback techniques, known relevant documents can be used to formulate a more accurate query, find more relevant documents, and re-construct the picture.

REFERENCES

[1] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics).* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[2] D. Blei and J. Lafferty. Correlated topic models. *Advances in Neural Information Processing Systems*, 18:147, 2006.

[3] D. M. Blei and J. D. Lafferty. Dynamic topic models. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 113–120, New York, NY, USA, 2006. ACM.

[4] G. Bouchard. Efficient bounds for the softmax function, applications to inference in hybrid models. *Advances in Neural Information Processing Systems*, 2007.

[5] M. Cox and T. Cox. Multidimensional scaling. *Handbook of Data Visualization*, pages 315–347, 2008.

[6] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American society for Information Science*, 41(6):391–407, 1990.

[7] A. Globerson, G. Chechik, F. Pereira, and N. Tishby. Euclidean embedding of co-occurrence data. *Journal of Machine Learning Research*, 8:2265–2295, 2007.

[8] M. Jordan, Z. Ghahramani, T. Jaakkola, and L. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, 1999.

[9] M. Khoshneshin and W. N. Street. Collaborative filtering via Euclidean embedding. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, pages 87–94, New York, NY, USA, 2010. ACM.

[10] J. Zhang. *Visualization for Information Retrieval.* Springer Verlag, 2008.

[2]http://web.eecs.utk.edu/research/lsi/

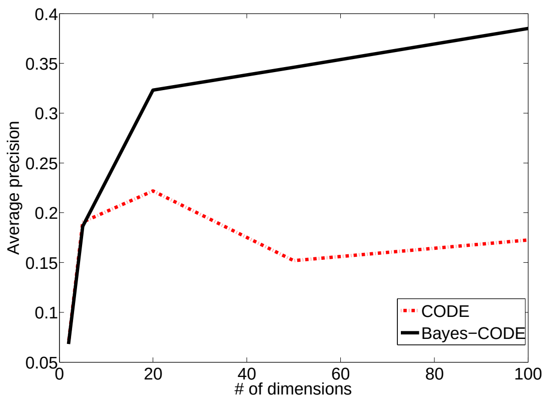

Fig. 2. The average precision results for CRAN dataset

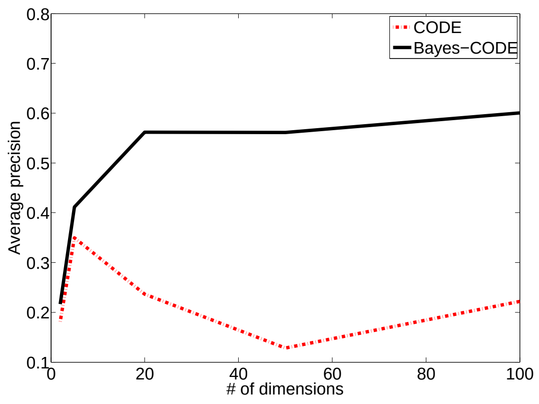

Fig. 3. The average precision results for MEDLINE dataset

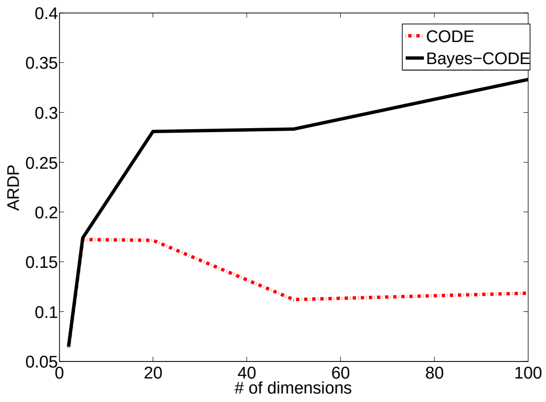

Fig. 4. The ARDP results for CRAN dataset

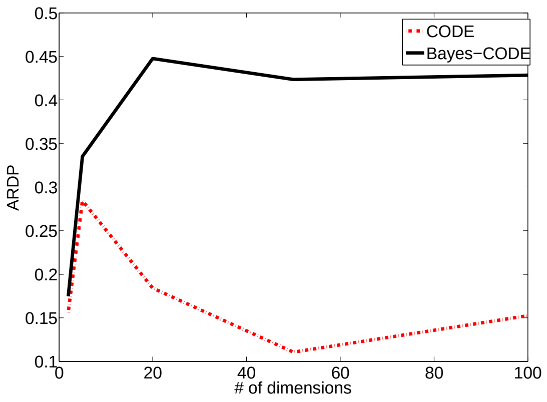

Fig. 5. The ARDP results for MEDLINE dataset

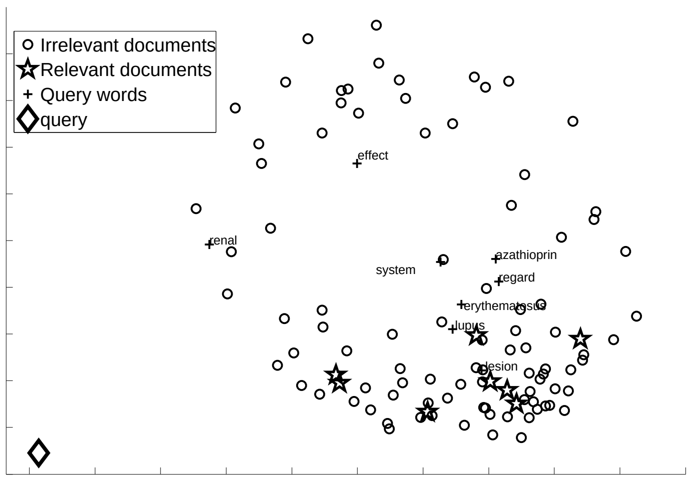

Fig. 6. Visualization of a typical query from MEDLINE dataset using CODE with 2 dimensions

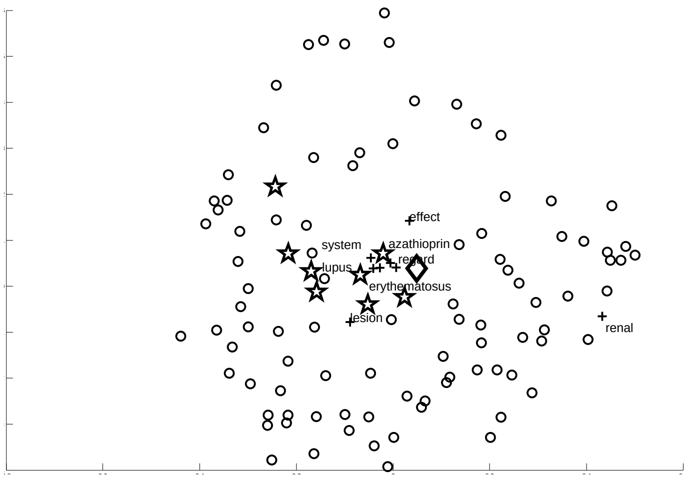

Fig. 7. Visualization of a typical query from MEDLINE dataset using query-based visualization + Bayes-CODE with 100 dimensions