EXTRACTING LATENT STRUCTURES IN NUMERICAL CLASSIFICATION: AN INVESTIGATION USING TWO FACTOR MODELS

*Arindam Choudhury, YewSoon Ong & Andy J. Keane**

Computational Engineering and Design Center
University of Southampton, U.K, S017 1BJ

ABSTRACT

We investigate the use of SVD based two factor models for numerical data classification. Motivations for such a study include the widespread success of such models (e.g, LSI) in textual information retrieval, emerging connections with well established statistical techniques and the increasing occurrence of mixed mode (text–and–numeric) data. A direct extension as well as an efficient modification of the LSI model applied to numerical data problems are presented and the associated problems and likely remedies discussed. The techniques under investigation are shown to perform competitively with respect to popular existing numerical classification techniques on a range of synthetic and real world benchmark data. In particular, we show that the modified LSI proposed in this work avoids confronting the optimal subspace selection problem yet generalizes well and remains computationally efficient for large data.

1. INTRODUCTION

Over the last 10 years, the latent semantic indexing technique (LSI) has been effectively used for information retrieval in the text domain in a variety of tasks, see for example [1, 2, 3]. LSI presumes the existence of a latent structure in the textual data and treats the unreliability in observed term-document association as a statistical problem to uncover this structure. Specifically it uses a truncated singular value decomposition (SVD) to project terms, documents and queries into the same multidimensional real valued derived feature space. Each query is represented as a pseudo-document formed from a weighted combination of terms. Since SVD is essentially a proximity based model, a subsequent use of a dot product based similarity metric retrieves documents related to the query.

The empirically observed success of LSI has often been attributed to its capability of filtering out noise and generation of a clean set of orthogonal basis vectors which help it project terms, documents and queries consistently into the same space. In recent years, the connections of LSI with other well known techniques have also brought forth theoretical evidence in its favor. We mention some of the more notable connections below.

The technique of LSI is closely related to that of Multi Dimensional Scaling (MDS) [4], a technique which seeks to construct a reduced order configuration for projecting input data while preserving a prespecified similarity measure in the least square sense. In fact, the document representation found using LSI is an optimal special case of MDS[5], when the inner product is used as the similarity measure.

The equivalence relationship between factor analysis techniques and multiple regression models first explored by Malinvaud et. al. [6] has also provided valuable insights into the effectiveness of the latent semantic model. Specifically, it has been shown that LSI reduces the magnitude of the specification error[1] which arises because of absence of certain keywords in the query. A conventional keyword search algorithm would assign zero(0) weights to such absent keywords. This might result in improper handling of important missing variables. LSI solves this problem by assigning weights given words in a query so that they can serve as proxies for synonyms and closely related terms not present in the query[7].

The success of LSI in information retrieval and its interesting relationships with existing data modeling techniques naturally leads one to contemplate on its application to numeric or non text based(mixed) domains. This paper investigates the use of latent semantic indexing as a tool for numerical data classification based on such a motivation. In particular, we investigate two techniques aimed at uncovering the latent association between numeric data attributes and the corresponding class labels. Typically, the resulting predictive models are similarity based models operating in a transformed feature space. A further common feature of the techniques discussed in this paper is their ability to handle mixed (continuous and/or nominal) attribute data.

The rest of the paper is organized as follows: section 2

* e–mail: {A.Choudhury, Y.S.Ong, Andy.Keane}@soton.ac.uk

[1] Specification error refers to the estimation errors arising out of the improper inclusion or exclusion of key explanatory variables from a model.

discusses a straightforward application of conventional LSI/SVD (as used in the text domain) to numeric data while section 3 presents a more efficient reformulation of the same. Related computational issues are discussed in section 4 while experimental investigations are presented in section 5. The paper concludes with a summary of the current research and pointers to future directions of work in section 6.

2. CONVENTIONAL LSI FOR NUMERIC DATA

As mentioned earlier, LSI in the text domain captures the uncertainty in association between terms and documents. This is achieved by computing and factorizing a term document co–occurrence frequency matrix. A straightforward extension to the numerical domain would therefore involve constructing an attribute instance co–occurrence frequency matrix. Since the attributes may themselves be continuous in nature, a discretization technique needs to be employed.

2.1. Entropy-based discretization

In the present setup, a recursive minimum entropy partitioning (RMEP) technique [8] has been employed for discretizing the input space of the continuous attributes. RMEP is a supervised heuristic discretization algorithm based on information theory which uses the *class information entropy* measure to determine bin boundaries for discretization. Given a dataset S, a feature a and a partition boundary T, the *class information entropy* of the partition induced by T is defined as follows:

$$E(a,T;S) = \frac{|S_1|}{|S|}Ent(S_1) + \frac{|S_2|}{|S|}Ent(S_2) \quad (1)$$

where S_1 and S_2 denote the two partition subsets (out of S), $|\cdot|$ represents the cardinality operation on a set, and $Ent(S_1)$ and $Ent(S_2)$ denote the respective class entropies. The binary discretization boundary T is chosen each time to minimize the entropy function over all possible partitions. The method proceeds by recursively partitioning the sets induced by T till a stopping criterion is satisfied. Recursive partitioning within a set of instances S stops if [8]:

$$Gain(a,T;S) < \frac{\log_2(N-1)}{N} + \frac{\Delta(a,T;S)}{N} \quad (2)$$

where N is the number of unique instances in set S and,

$$\begin{aligned} Gain(a,T;S) &= Ent(S) - E(a,T;S) & (3)\\ \Delta(a,T;S) &= \log_2(3^k-2) - [k.Ent(S) - & (4)\\ & \quad -k_1 Ent(S_1) - k_2 Ent(S_2)] & \end{aligned}$$

where k_1, k_2 denote the number of class labels in sets S_1 and S_2 (out of S), respectively. The reader is referred to [8] for a detailed exposition of the technique.

The attribute–instance frequency matrix is then constructed using the bin–widths generated by the above technique. A subsequent SVD factorization generates the predictive model encapsulating the latent structure in the data. Let n_a denote the total number of bins generated after discretization of attributes and N be the total number of instances assigned to m classes. Then represent the (discretized) attribute–instance frequency matrix by X and its truncated SVD representation by $X \sim U_k \sum_k V_k^T$, where $U_k \in \mathbb{R}^{n_a \times k}$ and $V_k \in \mathbb{R}^{N \times k}$ contain the k left and right singular vectors (of X), respectively. The columns (singular vectors) in U_k and V_k themselves correspond to the k largest singular values (of X) contained in $\sum_k = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_k), \in \mathbb{R}^{k \times k}, \sigma_1 > \sigma_2 > \ldots > \sigma_k$.

Given a novel data instance, we compute its discretized version $x \in \mathbb{R}^{n_a}$, and project it into the k dimensional reduced LSI subspace by computing $\hat{x} = x^T U_k \sum^{-1}$. A subsequent nearness score is computed using $s = \{\hat{x} \sum_k V_k^T\}$ and the class of the nearest neighbor document assigned to the new instance. Note that we shall henceforth refer to this scheme by the acronym LSI_{td}, where the subscript emphasizes its similarity with term document type LSI, commonly used in the information retrieval literature.

2.2. Discussion

While the utility of LSI as filtering technique is well understood, a few important questions/concerns remain unresolved. The most pertinent of these is the issue of model selection which in this case amounts to optimal subspace selection i.e., the optimal number (say k) of singular vectors(values) to be retained for best generalization on unseen data. To the best of our knowledge, *a priori* estimation of k from a direct inspection of the matrix X is still an open question[9, 10]. We mention that in its absence, all the results for LSI_{td} reported in this paper (see Tables 1,2) are the optimal model performance values obtained through exhaustive computation of the SVD and an *aposteriori* determination of the size of the optimal subspace. Since the computational complexity of SVD of $X \in \mathbb{R}^{n_a \times N}$ amounts to $\mathcal{O}(n_a N^2 + N^3)$, where N is the size of data, such an exhaustive computation may not be possible for large N.

However, once the SVD is available, optimal subspace selection can be done using any of the numerous model selection criteria available in the literature such as cross validation, minimum descriptive length (MDL) or Akaike's information criterion (AIC) [11]. Under reasonable assumptions on the distribution of examples, it can be shown [10] that the loglikelihood of the LSI model can be expressed (approximately) as a sum of squares of its singular values. Hence, one may as well track the loglikelihood of the model to approximately determine the optimal size of the reduced subspace. An interesting possibility is to compute a for-

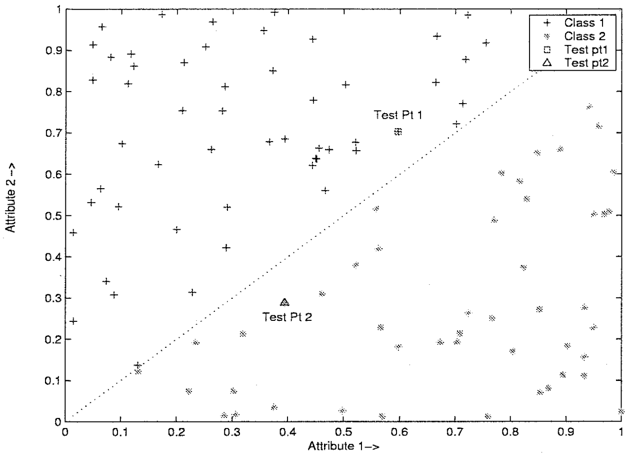

Figure 1: A synthetic 2D linearly separable toy problem.

ward stagewise incremental SVD update starting from an exact SVD factorization for a modestly sized random sample from the data. We have more to say on this approach in section 4.

3. A MODIFIED LSI

As mentioned in the last section, the conventional approach to LSI is faced with the twin issues of optimal subspace selection and rising computational costs. In what follows, we show that by a slight reformulation of the LSI approach, both these issues can be resolved. In particular, in the setting presented below, the question of optimal subspace selection has a trivial solution and the scalability of the modified scheme has no direct relationship with data size.

3.1. Attribute class co–occurrence

The emphasis in the current section (as opposed to conventional LSI, c.f section 2) is on directly modeling the uncertainty in the relationship between (discretized) attributes and the class labels. For this purpose, we construct the attribute class co-occurrence frequency matrix, $Z \in \mathbb{R}^{n_a \times m}$, where n_a, m carry over their usual meanings from the previous section. For clarity and ease of presentation, we describe the new approach in terms of the attribute instance frequency matrix, $X \in \mathbb{R}^{n_a \times N}$, introduced in the last section. Further, without any loss of generality we restrict our attention to the two–class case ($m = 2$); the extension to the multiclass case ($m > 2$) is trivial and follows naturally from the presentation below.

We begin by partitioning X into two blocks as $X = [X_1 \quad X_2]$, where $X_1 \in \mathbb{R}^{n_a \times N_1}$ and $X_2 \in \mathbb{R}^{n_a \times N_2}$, respectively denote instance attribute frequency matrices for classes 1 and 2 ($N = N_1 + N_2$). Then the attribute class frequency matrix $Z \in \mathbb{R}^{n_a \times 2}$ can be obtained by summing up all the columns in the respective classes. More compactly, in matrix notation, one can write:

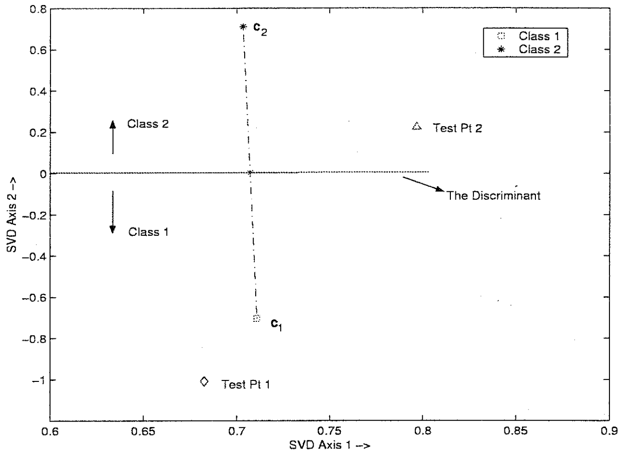

Figure 2: Test points in the projected "class space".

$$Z = X\Gamma \quad \text{where } \Gamma = \begin{pmatrix} \mathbf{1}^{N_1} & 0 \\ 0 & \mathbf{1}^{N_2} \end{pmatrix} \in \mathbb{R}^{N \times 2} \qquad (5)$$

and $\mathbf{1}^{N_1(N_2)} \in \mathbb{R}^{N_1(N_2)}$ is a column vector of $N_1(N_2)$ consecutive ones. Compute the SVD factorization $Z = A\Omega C$, where $A \in \mathbb{R}^{n_a \times 2}$ and $C \in \mathbb{C}^{2 \times 2}$ contain the left and right singular vectors of Z and $\Omega \in \mathbb{R}^{2 \times 2}$, is a diagonal matrix containing the corresponding singular values. As before, given a novel data instance, we compute its discretized version $x \in \mathbb{R}^{n_a}$, and project it into the reduced 2(i.e., m) dimensional LSI subspace by computing $\hat{z} = x^T A \Omega^{-1}$. A subsequent nearness score s can now be computed by treating the columns of $C = [c_1 \quad c_2]$ as representative "class vectors", i.e. $s = [\hat{z}c_1 \quad \hat{z}c_2]$ and the appropriate nearest class assigned to the new instance. We shall refer to the scheme introduced here as LSI_{ac}, where the subscript indicates the attribute class nature of Z directly. Since, in general, $C \in \mathbb{R}^{m \times m}$, and $C^T C = I$, the columns of C i.e., the "class vectors" fully characterize an m–dimensional space, which we shall refer to as the "class space". We mention that this interpretation is consistent with the proximity based probabilistic model proposed in [10], where the columns of C perform the role of the so called characteristic vectors.

Fig.1 shows a toy problem and the ideal separating plane. Also shown are two test points which are projected into the "class space" defined above. Fig. 2 shows geometrically the effect of projecting the test points into the space where the "class vectors" reside (as discussed above). The perpendicular bisector of the line joining the two "class vectors" $\mathbf{c}_1$ and $\mathbf{c}_2$ has been used to decide the class of the new query. As shown in the figure, test point 1 lies on the same side as the class vector $\mathbf{c}_1$ and hence is labelled as belonging to the class C_1. Similarly, test point 2 lies on the same side as the class vector $\mathbf{c}_2$ and hence is labelled as belonging to the class $\mathcal{C}_2$. Since the ascribed class labels of the test points match the true ones (see Fig. 1), the LSI-based classifier has been able to correctly classify them.

3.2. Discussion

Notice that the simple modification suggested in Equation (5) offers us two distinct advantages when compared with the traditional LSI approach (ref. section 2). Firstly, we note that computing an SVD factorization of the attribute-class co-occurrence frequency matrix, $Z \in \mathbb{R}^{n_a \times m}$ involves expending $\mathcal{O}(n_a m^2 + m^3)$ computations[12]. Therefore, computational cost scales linearly with number of discrete regions, n_a, and has a cubic dependence on the number of classes, m, present in the dataset. Since, in practice, $m \ll N$ and is usually a constant for a problem, the effective computational complexity is $\mathcal{O}(n_a)$, i.e., it scales linearly with number of discretized bins (as opposed $\mathcal{O}(N^3)$ for LSI_{td}). Since n_a has no (explicit) relationship with N, the data size, this scaling is extremely attractive. Secondly, since $\rho = \text{rank}(Z) \sim \min(n_a, m)$, the size ($k$) of the optimal LSI subspace will be less than or equal to m. When $\rho = m$, the m columns of the C matrix can be effectively used as class representatives and prediction done as above. Finally, the case of $\rho < m$ i.e., when some singular values are exceptionally small or zero, may be used to indicate (i) the likely presence of *spurious* class(es) i.e., when the underlying structure in the data does not bear out the distinction between two (or more) class labels, and/or (ii) the possible ill–posedness of problem or insufficient data.

4. FURTHER COMPUTATIONAL ISSUES

An interesting situation arises when one needs to frequently update the SVD factorization obtained in the LSI schemes. One can show that for large data (i.e., $N \gg (n_a, m)$), the LSI_{ac} model can be updated atleast $N^2/(n_a m)$[15] times faster than LSI_{td}, per update. An alternative approach is computing a semi-discrete decomposition (SDD)[14] of the co-occurrence frequency matrix, Z(or X). SDD requires only about one-twentieth of the storage space required by SVD and about half the query time. Furthermore, updating the SDD is much faster than updating the SVD, although constructing the initial SDD representation may take five times as long as constructing the corresponding SVD. However, since this is only a one time expense, it does not increase the cost of the update process. Finally, deleting terms from the SDD representation is a trivial operation while it is quite complicated for SVD, due to orthogonality requirements.

However, since both SVD and SDD update techniques utilize the initial approximation structures to varying degrees, an inevitable performance degradation occurs with the addition of a significant number of new elements due to loss of orthogonality in the updated system.

Method	Noise →						
	0%	5%	10%	15%	20%	25%	30%
LSI_{td}	6.1	9.8	15.2	16.8	20.6	20.5	35.7
(k)	(5)	(5)	(5)	(4)	(4)	(4)	(4)
LSI_{ac}	3.5	9.5	15.2	23.6	27.7	29.8	39.5
$(k = 2)$							

Table 1: Performance evaluation on test function at various noise levels; k = optimal subspace size.

5. EXPERIMENTAL INVESTIGATIONS

The techniques investigated in this paper were compared with each other as well as with discretized versions of other popular machine learning techniques viz., naïve Bayesian classifier and C4.5 trees. A standard CV-10 run was performed for all the techniques considered to compute the mean percentage accuracy and the variability about the mean.

Table (1) illustrates a performance comparison of LSI_{td} and LSI_{ac} on synthetic 2D two class noisy data. For this purpose, we corrupted randomly generated noise free data sets obeying the decision $sign(f)$, where $f(\cdot,\cdot)$ is a known cubic function of the input variables, with varying degrees of output noise intensities. For each noise level, the data was randomly split into a training and a testing set of 1000 instances each. The results reported in Table (1) are averaged over 20 such splits.

We observe that at relatively low noise levels, LSI_{ac} performs as well as LSI_{td}, albeit with a (fixed size) smaller optimal subspace. Understandably, performance deteriorates as the noise level (hardness) is increased. Notice that the disadvantage of smaller, fixed subspace size (for LSI_{ac}) becomes more clear at high noise levels, when more effective filtering is required.

Table (2) summarizes the 10–CV performance comparison for 8 standard UCI machine learning datasets. For LSI_{td}, the optimal model size is also indicated. We observe that in general, the SVD based techniques perform as well as the standard machine learning techniques for numeric data. In particular, we note the performance of the LSI_{ac} technique, which is remarkable given the fact that it is a one–shot process and does not involve a model selection stage (as opposed to C4.5 and LSI_{td}). This fact along with its better computational features (both offline and online) are important outcomes of this research. Here, we also point out the exceptional accuracy shown by LSI_{td} on the vehicle dataset; we believe this outcome is a combination of high noise in the data and the availability of a relatively large set of basis vectors for enhanced filtering.

Dataset	Methods			
Name	Discrete	Discrete	Discrete	Discrete
(#Class)	C4.5	NB	LSI_{td} ($\frac{\mathrm{k}}{\#\mathrm{SV}}$)	LSI_{ac}
Breast(2)	94.5±2.1	97.0±1.6	95.9±2.1 ($\frac{5}{21}$)	97.3±1.8
Diabetes(2)	74.5±4.6	75.1±3.6	72.3±5.2 ($\frac{9}{10}$)	74.6±7.1
German(2)	72.9±5.8	72.6±5.3	66.3±5.1 ($\frac{7}{7}$)	68.4±4.1
Glass(7)	71.0±8.3	71.5±6.8	65.9±10.6 ($\frac{11}{14}$)	69.0±9.1
Glass2(2)	79.0±8.6	80.3±6.3	77.6±13.2 ($\frac{7}{8}$)	85.6±8.4
Heart(2)	82.2±10.1	82.2±5.5	80.7±7.5 ($\frac{10}{10}$)	84.1±6.9
Iris(3)	95.3±4.5	92.7±5.8	95.4±4.9 ($\frac{4}{8}$)	94.7±4.2
Vehicle(4)	69.1±5.2	61.1±5.4	72.4±5.1 ($\frac{41}{51}$)	62.3±4.9

Table 2: Percentage accuracy comparison for standard UCI Datasets; k = optimal subspace size based on test data, # sv = total number of singular values.

6. CONCLUDING REMARKS

In this article, we investigated the application of SVD based models to numerical data classification. Traditionally, such techniques (e.g., LSI) have been used effectively for information retrieval in the text domain. A straightforward extension of the popular LSI technique to numeric data performs well (in terms of accuracy) but is fraught with problems of model selection and escalating computational costs with datasize. We propose a modified LSI which directly models the uncertainty relationship between class labels and the input feature space. The resulting scheme is more efficient than the conventional one and also circumvents the optimal subspace selection problem faced in the conventional approach. Performance comparisons on synthetic and real–world benchmark data demonstrate the competitiveness of the current approach w.r.t popular existing machine learning techniques. Various computational issues and bottlenecks are highlighted and possible remedies suggested.

As mentioned earlier, updating the classifier without recomputing the SVD (or the SDD) would inevitably lead to a degradation in performance. This may be traced to the loss of orthogonality in the updated system. An interesting enhancement to the present work would be to use the extent of this loss to derive a threshold to determine when to stop and recompute the SVD (or the SDD) all over again. Work in this direction is currently in progress although much remains to be done. An alternative direction of work is the development of fast stable SVD updates, while minimizing the loss of orthogonality of the system [15].

Finally, it is important to reiterate that while a small number of discriminative directions can often provide fairly good generalization for low noise situations, such an approach maybe counter productive for high noise scenarios. In such cases, one may actually need to generate more discriminative directions (basis vectors) to enable effective filtering, although the associated costs to be incurred in model selection need to be appreciated as well.

7. REFERENCES

[1] P. W. Foltz and S. T. Dumais., Personalized information delivery: An analysis of information filtering methods, *Communications of the ACM*, Vol.35(12), pp. 51–60, 1992

[2] S. T. Dumais, T. A. Letsche, M. L. Littman and T. K. Landaeur *Automatic cross-language retrieval using latent semantic indexing* in *Proceedings of the AAAI Spring Symposium on Cross-language Text and Speech Retrieval, March 1997*

[3] S. T. Dumais and J. Nielsen, Automating the assignment of submitted manuscripts to reviewers, in *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM Press, pp 233-244.

[4] I. Borge and J. Lingoes, *Multidimensional similarity structure analysis,* Springer Verlag, New York, 1987.

[5] B. T. Bartell, G. W. Cottrell, and R. K. Belew, Latent semantic indexing is an optimal special case of multidimensional scaling, In *Proceedings of the SIGIR '92*, pp 161-167, 1992.

[6] E. Malinvaud, Statistical Methods of Econometrics, In Henri Theil (Ed.), *Studies in Mathematical and Managerial Economics*, Vol. 6, pp 374-411, American Elsevier Publishing Co., New York, NY, 1970.

[7] R. E. Story, An explanation of the effectiveness of latent semantic indexing by means of a bayesian regression model. *Information Processing & Management*, 32(3), pp.329–344, 1996.

[8] U. M. Fayyad and K. B. Irani, Multi-interval discretization of continuous-valued attributes for classification learning, In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, Morgan Kaufmann, pp. 1022-1027.

[9] S.T.Dumais, Using LSI for information retrieval: TREC-3 experiments. D. Harman(Ed.), Overview of TREC-3, National Institute of Standards and Technology special publication, Tech report 500–335, 1995.

[10] C.H.Q. Ding, A Similarity-based probability model for latent semantic indexing In *Proceedings of the 22nd ACM SIGIR Conference, pp. 59–65, 1999*

[11] M.H. Hansen & B. Yu, Model selection and the principle of minimum description length, *Journal of the American Statistical Association*, Vol. 96(454), pp. 746–774, 2001.

[12] G. Golub and C. V. Loan, *Matrix Computations,* John Hopkins University, Baltimore, MD, Second edition, 1989.

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

[14] T. G. Kolda and D. P. O'Leary, A semi-discrete matrix decomposition for latent semantic indexing in information retrieval, *ACM Transactions on Information Systems*,vol 16(4), 1998.

[15] M. Gu and S. C. Eisenstat, A stable and fast algorithm for updating the singular value decomposition, Technical Report YALEU/DCS/RR-966, Yale University, New Haven, C-T, 1994.