Using Active Manifold Learning with Twitter Big Data

Catarina Silva, IEEE Member, Bernardete Ribeiro, IEEE Senior Member, Joana Costa, Mário Antunes

***Abstract*— Big data is currently a dazzling field with numerous applications. Current approaches to deal with big data usually include muscled infrastructures and frameworks that permit the parallelization of the defined tasks. Nevertheless, such solutions fall short when online scenarios are in place, since users expect swift feedback.**

Reduction techniques are commonly used in different machine learning problems to improve training and classification efficiency as required in online big data applications. There are two paths to exploit when reducing problems: (i) reduce the dimensionality by pruning or reformulating the features; and (ii) reducing the size of the sample by choosing the more relevant examples. Both approaches come with benefits, not only of time consumed to build a model, but eventually also performance-wise, usually by reducing overfitting and improving generalization capabilities.

In this paper we investigate reduction techniques that tackle both dimensionality and size of big data. We propose a framework that combines a manifold learning approach to reduce dimensionality and an active learning SVM-based strategy to reduce the size of labeled sample needed. Results on Twitter data show the potential of the proposed active manifold learning approach.

I. Introduction and Background

Big data is one of the major trends in research in the last years and is expected that science, business, industry, government, society, etc. will undergo a thorough change with the influence of big data [1]. Although one might argue that we have been in the presence of large data sets for a while and that this new term is just a hype, there are in fact tangible outcomes of this re-branding that are worth analysing, namely by the availability of specific (and free) frameworks [2] like Hadoop (`http://hadoop.apache.org/`) or Mahout (`http://mahout.apache.org/`).

Big data are a collection of dataset consisting of massive unstructured, semi-structured, and structured data [3]. Big data is being generated by everything around us at all times (`www.ibm.com/big-data/`). One of the major sources of data are the social networks, e.g. Twitter (`http://twitter.com/`), Facebook (`http://facebook.com/`) or Instagram (`http://instagram.com/`). In this social era, individuals and companies produce enormous amounts of data (*Volume*), extremely heterogeneous (*Variety*) and at alarming rates (*Velocity*). And thus with social networks we get the 3 V's that characterize big data scenarios. A fourth 'V', for *Veracity*, has been also considered and is in fact extremely important since it relates to the uncertainty in data and the trust one can or can not put on big data information. Specially when dealing with social networks big data, it can became crucial. Given this setup data scientists are in high demand and practical results are becoming extremely valuable research and business-wise. An example can be found in [4] where a distributed strategy with decision trees and Support Vector Machines (SVM) is proposed to predict the price trends of stock futures with large amounts of data. The focus was on the proposal of statistical features which where achieves using MapReduce.

Putting more emphasis on representation, in [3] a unified tensor model is proposed to represent the unstructured, semi-structured, and structured data where various types of data are represented as subtensors and then are merged to a unified tensor. To extract information an approach based on singular value decomposition (SVD) method is introduced showing competitive results in terms of time complexity, memory usage, and approximation accuracy.

Regarding dimensionality reduction, one can find in [5] an alternative to the usually greedy strategies, by using the Orthogonal Centroid (OC) as feature extraction method that is found very effective in classification problems. Another approach is presented in [6] where a two-step process is proposed to detect forged signatures, first by extracting features from biometric images using Discrete Cosine Transform and second using a GPU-based SVM classifier.

Nevertheless these cutting edge applications, challenges arise when using such robust frameworks in online scenarios. When searching information from an online source like Twitter, reducing size and dimension in supervised learning has gained interest in the machine learning community as a way to reduce time spent constructing learning models, but also as an effective way of improving performance by pruning extraneous data. In fact, dimensionality reduction has been considered as an essential data preprocessing technique for large-scale

Catarina Silva and Joana Costa are with the School of Technology and Management, Polytechnic Institute of Leiria, Portugal and with CISUC - Centre for Informatics and Systems of the University of Coimbra, Portugal (`{catarina, joana.x}@dei.uc.pt`).

Bernardete Ribeiro is with Department of Informatics Engineering, CISUC, University of Coimbra, Portugal (`bribeiro@dei.uc.pt`).

Mário Antunes is with School of Technology and Management, Polytechnic Institute of Leiria, Portugal and with CRACS - Center for Research in Advanced Computing Systems, INESC-TEC, Portugal (`mario.antunes@ipleiria.pt`).

and streaming data classification tasks [5]. This appeal is underpinned by the tremendous increase of digital information that often leads applications and learning algorithms to include a dimension/size reduction step.

High dimensionality has usually at least two angles. On one hand, the number of examples is massive and the difficulty to keep a representative training set of labeled instances is growing. On the other hand, the representation of each example can also reach high dimensions and make the decision space more complex in applications like text classification or gene expression.

In this paper we propose a framework to reduce size and dimension in Twitter Big Data. Size is reduced using a support vector machine active learning strategy that takes place after an Isomap-based nonlinear algorithm is put forward to reduce the initial huge dimensionality of a text classification problem.

Next two sections will introduce both reductions we are dealing with: dimensionality reduction on Section II and size reduction on Section III. Then, in Section IV we describe the manifold active learning approach and in Section V we show the results obtained along with the experimental setup. Finally, Section VI discusses conclusions and future work.

II. Dimensionality reduction - Manifold Learning

Initial dimensionality reduction is carried out in the feature space as a pre-processing step. Several supervised and unsupervised techniques can be applied. Manifold learning strategies, like Isomap (Isometric Mapping) [7], are effective for extracting nonlinear structures from high-dimensional data in pattern recognition [8]. Finding the structure behind the data may be important for a number of reasons in many applications. One possible application is data visualization. Graphical depiction of the document set can potentially be crucial, since it makes possible to quickly give large amounts of information to a human operator [9]. To this purpose it is appropriately assumed that the data lies on a statistical manifold, or a manifold of probabilistic generative models [10]. It can be regarded as a supervised learning method, where the training labels play a central role. In such a scenario, manifold learning can be used not only with the traditionally associated algorithms, such as K-Nearest Neighbors (K-NN), but also with state-of-the-art kernel-based machines like support vector machines (SVMs) [12].

Feature reduction methods aim at choosing from the available set of features a smaller set that more efficiently represents the data. Such reduction is not needed for all classification algorithms as some classifiers are capable of feature selection themselves. However for some other classifiers feature selection is mandatory, since a large number of irrelevant features can significantly weaken the classifier accuracy.

Many approaches have been proposed for dimensionality reduction, such as the well-known methods of principal component analysis (PCA) [18], independent component analysis (ICA) [19] and multidimensional scaling (MDS) [20]. All these methods are well understood and efficient and have thus been widely used in visualization and classification. Unfortunately, they share a common inherent limitation: they are all linear methods while the distributions of most real-world data are nonlinear. In [21] a survey on feature extraction foundations and applications can be found.

An emerging nonlinear dimension reduction technique is manifold learning [22], [23], which is the process of estimating a low-dimensional structure which underlies a collection of high-dimensional data. Manifold learning can be viewed as implicitly inverting a generative model for a given set of observations [24]. Let Y be a d dimensional domain contained in a Euclidean space $\mathbb{R}^d$. Let $f : Y \rightarrow \mathbb{R}^D$ be a smooth embedding for some $D > d$. The goal of manifold learning is to recover Y and f given N points in $\mathbb{R}^D$. Isomap [7] provides an implicit description of the mapping f (or f^{-1}). Given $X = \{\mathbf{x}_i \in \mathbb{R}^D | i = 1 \dots N\}$ find $Y = \{\mathbf{y}_i \in \mathbb{R}^d | i = 1 \dots N\}$ such that $\{\mathbf{x}_i = f(\mathbf{y}_i) | i = 1 \dots N\}$. Without imposing any restrictions of f, the problem is ill-posed. The simplest case is a linear isometry, i.e. f is a linear mapping from $\mathbb{R}^d \rightarrow \mathbb{R}^D$, where $D > d$.

In Isomap [7] the local neighborhood of each example is preserved, while trying to obtain highly nonlinear embeddings with manifold learning. For data lying on a nonlinear manifold, the *true distance* between two data points is the geodesic distance on the manifold, i.e. the distance along the surface of the manifold, rather than the straight-line Euclidean distance. The main purpose of Isomap is to find the intrinsic geometry of the data, as captured in the geodesic manifold distances between all pairs of data points. The approximation of geodesic distance is divided into two cases. In case of neighboring points, Euclidean distance in the input space provides a good approximation to geodesic distance. In case of faraway points, geodesic distance can be approximated by adding up a sequence of *short hops* between neighboring points. Isomap shares some advantages with PCA and MDS, such as computational efficiency and asymptotic convergence guarantees, but with more flexibility to learn a broad class of nonlinear manifolds. The Isomap algorithm takes as input the distances $d(\mathbf{x}_i, \mathbf{x}_j)$ between all pairs $\mathbf{x}_i$ and $\mathbf{x}_j$ from N data points in the high-dimensional input space. The algorithm outputs coordinate vectors $\mathbf{y}_i$ in a d-dimensional Euclidean space that best represent the intrinsic geometry of the data. Isomap is accomplished following these steps:

1) Construct neighborhood graph: Define the graph G over all data points by connecting points $\mathbf{x}_i$ and $\mathbf{x}_j$ if they are closer than a certain distance ε, or if $\mathbf{x}_i$ is one of

the K nearest neighbors of $\mathbf{x}_j$. Set edge lengths equal to $d(\mathbf{x}_i, \mathbf{x}_j)$.

2) Compute shortest paths: Initialize $d_G(\mathbf{x}_i, \mathbf{x}_j) = d(\mathbf{x}_i, \mathbf{x}_j)$ if $\mathbf{x}_i$ and $\mathbf{x}_j$ are linked by an edge; $d_G(\mathbf{x}_i, \mathbf{x}_j) = +\infty$ otherwise. Then for each value of $k = 1, 2, \ldots, N$ in turn, replace all entries $d_G(\mathbf{x}_i, \mathbf{x}_j)$ by $min\{d_G(\mathbf{x}_i, \mathbf{x}_j), d_G(\mathbf{x}_i, \mathbf{x}_k) + d_G(\mathbf{x}_k, \mathbf{x}_j)\}$. The matrix of final values $\mathbf{D}_G = \{d_G(\mathbf{x}_i, \mathbf{x}_j)\}$ will contain the shortest path distances between all pairs of points in G.

3) Apply MDS to the resulting geodesic distance matrix to find a d-dimensional embedding.

This is an unsupervised procedure and constitutes a preprocessing step for classification. Basically it performs a transformation from a high dimensional input data space into a lower dimensional feature space. Then a classifier, for instance, K-NN can be applied to the resulting data. However, the mapping function given by Isomap is only implicitly defined. Therefore, it should be learned by nonlinear interpolation techniques, such as generalized regression neural networks, which can then transform the new test data into the reduced feature space before prediction.

In the supervised version of Isomap [25], the information provided by the training class labels is used to guide the procedure of dimensionality reduction. The training labels are used to refine the distances between inputs. The rationale is that both classification and visualization can benefit when the inter-class dissimilarity is larger than the intra-class dissimilarity. However, this can also make the algorithm overfit the training set and can often make the neighborhood graph of the input data disconnected. The Euclidean distance $d(\mathbf{x}_i, \mathbf{x}_j)$ between two given observations $\mathbf{x}_i$ and $\mathbf{x}_j$, labeled y_i and y_j respectively, is replaced by a dissimilarity measure [25]:

$$D(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} \sqrt{1 - e^{\frac{-d^2(\mathbf{x}_i, \mathbf{x}_j)}{\beta}}} & y_i = y_j, \\ \sqrt{e^{\frac{-d^2(\mathbf{x}_i, \mathbf{x}_j)}{\beta}}} - \alpha & y_i \neq y_j. \end{cases} \quad (1)$$

Note that the Euclidean distance $d(\mathbf{x}_i, \mathbf{x}_j)$ is in the exponent and the parameter β is used to avoid that $D(\mathbf{x}_i, \mathbf{x}_j)$ increases too rapidly when $d(\mathbf{x}_i, \mathbf{x}_j)$ is relatively large. Hence, β depends on the *density* of the data set and is usually set to the average Euclidean distance between all pairs of data points. On the other hand, α gives a certain possibility to points in different classes to be *closer*, i.e. to be more similar, than those in the same class. This procedure allows a better determination of the relevant features and will definitely improve visualization.

III. Size reduction - Active Learning

To reduce the number of labeled training examples needed for a supervised learning algorithm, such as support vector machines (SVMs), there have been many studies employing unlabeled documents in the learning task, like, transductive learning [13], co-training [14] and active learning [15], [16], [17]. Usually, the training set is chosen to be a random sampling of instances. However, in many cases active learning can be employed. Here, the learner can actively choose the training data. It is hoped that allowing the learner this extra flexibility will reduce the learner's need for large quantities of labeled data and hence reduce training time [17]. Pool-based active learning for classification was introduced by Lewis and Gale [15]. The learner has access to a pool of unlabeled data and can request the true class label for a certain number of instances in the pool.

To achieve the best classification performance with a machine learning technique, one can face two problems: not enough data or too much data. Active learning mechanisms can be applied in both scenarios:

1) When there is not enough labeled data, but unlabeled data is readily available;
2) When there is too much labeled data and algorithms can benefit if a selection is carried out.

Any active learning algorithm selects of a pool of examples which should be used (usually after being classified) to create the learning model. Hence, to actively learn we aim at selecting those examples that, when labeled and incorporated into training, will minimize classification error over the distribution of future examples. The main issue with active learning is finding a way to choose good requests or queries from the pool. It is assumed that the instances $\mathbf{x}$ are independent and identically distributed (i.i.d.) according to some underlying distribution $F(\mathbf{x})$ and the labels are distributed with some conditional distribution [11].

In this work we propose an SVM-based active learning strategy. In SVMs, Support Vectors (SVs) and weights define the obtained model. SVs define the optimal separating hyperplane (OSH) [12]. According to this interpretation, the most informative unlabeled examples are potentially those closer to any of the existing SV in the model, since they can potential alter the OSH. To define these examples we propose a kernel-based approach, that defines a design matrix $\mathbf{\Psi}$, assessing the distances between the existing SV and the set of unlabeled examples available.

Given an initial SVM model, induced using input-output labeled training data $(\mathbf{x}_1, y_1), ...(\mathbf{x}_l, y_l) \in \mathbb{R}^M \times \{\pm 1\}$, resulting in a number s of SV, $\boldsymbol{\rho}$, $(\boldsymbol{\rho}_1, ..., \boldsymbol{\rho}_s) \in \mathbb{R}^M$. Given also unlabeled data $\mathbf{U}$, $(\mathbf{u}_1, ..., \mathbf{u}_h) \in \mathbb{R}^M$, the distance between an SV and an unlabeled document is defined as

$$\Psi_{ij} = k(\boldsymbol{\rho}_i, \mathbf{u}_j), \quad (2)$$

where k represents the kernel used to define a higher

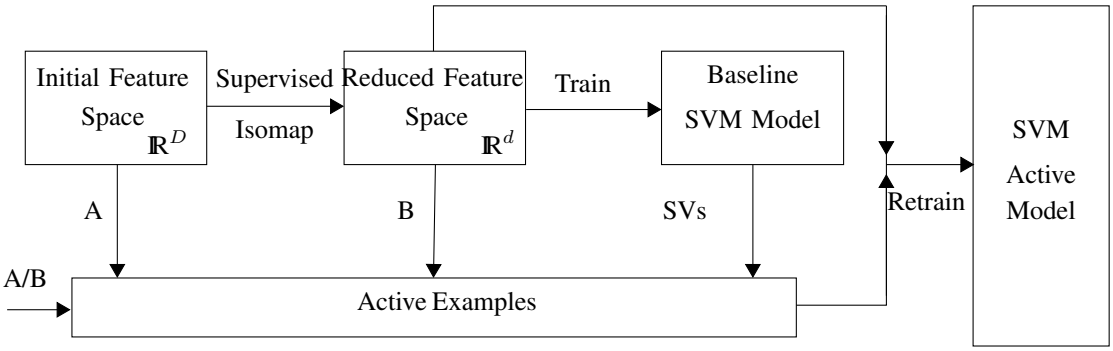


Fig. 1. Active learning strategy.

dimension space where points can be compared. For a generic kernel function Φ, Ψ_{ij} is the dot product

$$\Psi_{ij} = \langle \Phi(\boldsymbol{\rho}_i), \Phi(\mathbf{u}_j) \rangle. \tag{3}$$

Assuming a linear kernel, Ψ_{ij} is simplified

$$\Psi_{ij} = \langle \boldsymbol{\rho}_i, \mathbf{u}_j \rangle. \tag{4}$$

For a linear kernel the design matrix is simplified

$$\boldsymbol{\Psi}_{linear} = \boldsymbol{\rho} \,.\, \mathbf{U}'. \tag{5}$$

After this design matrix is constructed, it remains to be determined which unlabeled examples are potentially more informative, i.e. which ones are closer to any current SV. The procedure is easily implemented as follows. First, the closest SV to any given unlabeled document is determined taking the maximum value of each column of the design matrix (6). Second, a definable number of unlabeled examples with smaller minimum distance to an SV are chosen and added to the training set.

$$\left[\max(k(\boldsymbol{\rho}_i, \mathbf{u}_1)) \quad \ldots \quad \max(k(\boldsymbol{\rho}_i, \mathbf{u}_h))\right]. \tag{6}$$

Next section will detail the proposed approach that includes the above expalined active learning strategy and the previously introduced manifold reduction approach.

IV. Proposed approach

Our feature space reduction approach is a manifold learning strategy, underpinned by supervised Isomap [25]. Thus we use the training labels in the corpus to provide a better construction of features. We further apply the dissimilarity measure (1) to enhance the baseline Isomap Euclidean distance using label information, with α taking the value of 0.65 and β the average Euclidean distance between all pairs of text data points.

When a reduced space is reached, our aim is to learn a kernel-based model that can be applied in unseen examples. We propose an active learning support vector machine (SVM) with a linear kernel. For testing, however, Isomap does not provide an explicit mapping of documents. Therefore we can not generate the test set directly, since we would need to use the labels. Hence, we use a generalized regression neural network (GRNN) [26] to learn the mapping and apply it to each test document, before the SVM prediction phase.

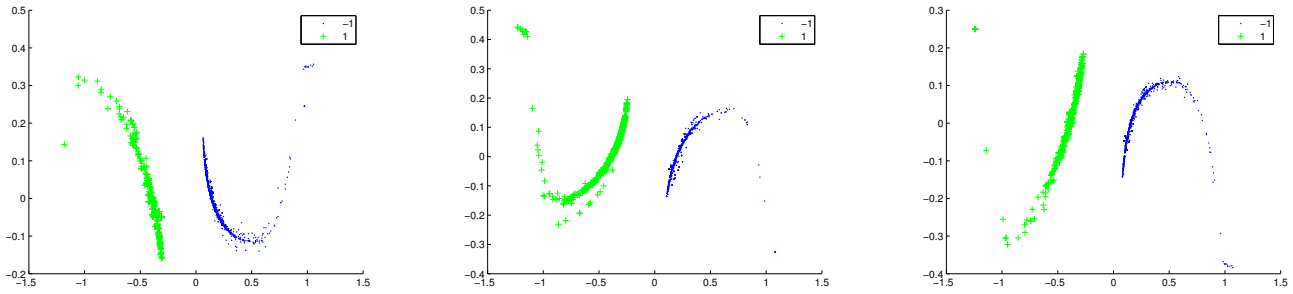
To apply the active learning strategy, we use the design matrix introduced in Section III to determine the active examples (which unlabeled examples are potentially more informative). The design matrix can be constructed in the original $\mathbb{R}^D$ space or in the reduced $\mathbb{R}^d$ space. As can be gleaned from Fig. 1, one can add active examples choosing from the initial feature space (A) or from the reduced feature space (B). However, strategy A (adding examples from $\mathbb{R}^D$) is computationally more intensive, while strategy B is straightforward. To choose active learning examples from the original feature space, first the baseline SV have to be remapped back into $\mathbb{R}^D$, then the design matrix is constructed and the active examples chosen. Before the final learning procedure can take place, a new Isomap feature reduction step with these new examples is carried out. On the other hand, choosing the examples directly from the reduced feature space includes a more complex initial Isomap step, with potential active examples, but does not include other overheads. Henceforth we will refer to the active learning strategy that uses the initial feature space as *Active A* and to the one that uses the reduced feature space as *Active B*, as represented in Fig. 1.

V. Experimental Setup

To test the proposed framework we will apply it to a real big data dataset retrieved form Twitter public stream

A. Problem Description: Twitter classification

Falta aqui a descrição do data set. Talvez possa ser muito semelhante ao do paper da joana, mas neste caso só com 3 classes: BJS: Bieber, Jobs, Syrisa



(a) #bieber (b) #jobs (c) #syrisa

Fig. 2. Separation of classes in the reduced dimension space.

B. Experimental Results and Discussion

Table I presents the performance test results for the 3 hashtags averaged over 10 runs using 70% of the examples for training and 30% for testing.

TABLE I

PERFORMANCES FOR THE THE HASHTAGS.

	Precision	Recall	F1	Accuracy
#bieber	76.49%	97.44%	87.96%	87.96%
#jobs	99.86%	36.36%	53.16%	78.61%
#syrisa	100.00%	79.05%	88.27%	92.93%
Average	92.12%	70.95%	75.47%	86.50%

AQUI FALTA A ANÁLISE DOS RESULTADOS.

An additional benefit from the use of manifold learning to reduce the feature space dimension is the possibility of providing a visual manifestation of the classification problem, not possible in the initial feature space. Fig. 2 shows the method's visualization capabilities.

ESTA ANÁLISE É DOS DADOS ANTERIORES DO REUTERS.

1) Statistical analysis: Tables II and III summarize the results in terms of statistical significance of the difference between three methods by means of t-test for both ModApte and Small Splits.

The significance level is set as 5%, so that the p-value less than 5% indicates that the two underlying methods are significantly different in the mean. As it may be observed in Table II, the method Active A is better than the Baseline approach in the ModApte split, thus we reject the Null Hypothesis with p-value = 0.019525 at a significance level of 5%.

As for the Small split the statistical results are highly significant. As illustrated in Table III method Active B significantly outperforms both the Baseline and Active A at a significance level of 1% in terms of F1-score. As for the Active A the t-test is significant as shown by the p-value 0.013506 which indicates that the method is better than the Baseline approach at 5% of significance level.

TABLE II

SIGNIFICANCE TESTS WITH STATISTICAL VARIABLE F1-SCORE FOR MODAPTE SPLIT: (A) BASELINE, (B) ACTIVE A , (C) ACTIVE B, SIGNIFICANCE LEVEL IS AT 5%.

	Active A	Active B
Baseline	0.019525∗	0.141347
Active A		0.399169

TABLE III

SIGNIFICANCE TESTS WITH STATISTICAL VARIABLE F1-SCORE FOR SMALL SPLIT: (A) BASELINE, (B) ACTIVE A (C) ACTIVE B, SIGNIFICANCE IS AT 5% LEVEL WITH P-VALUES INDICATED.

	Active A	Active B
Baseline	0.013506∗	0.000287**
Active A		0.002093**

VI. CONCLUSIONS AND FUTURE WORK

REFERENCES

[1] Zhi-Hua Zhou, N. V. Chawla, Yaochu Jin, G. J. Williams, "Big Data Opportunities and Challenges: Discussions from Data Analytics Perspectives", IEEE Computational Intelligence Magazine 9(4), pp. 62-74, 2014.

[2] A. Antoniades, C. C. Took, "A Google approach for computational intelligence in big data", IEEE International Joint Conference on Neural Networks (IJCNN), pp. 1050-1054, 2014.

[3] Liwei Kuang, Fei Hao, L. T. Yang, Man Lin, Changqing Luo, Geyong Min, "A Tensor-Based Approach for Big Data Representation and Dimensionality Reduction", IEEE Transactions on Emerging Topics in Computing 2(3), pp. 280 - 291, 2014.

[4] Dingxian Wang, Xiao Liu, Mengdi Wang, "A DT-SVM Strategy for Stock Futures Prediction with Big Data", 2013 IEEE 16th International Conference on Computational Science and Engineering (CSE), pp. 1005 - 1012, 2013.'

[5] A. C. Wilkerson, H. Chintakunta, H. Krim, "Computing persistent features in big data: A distributed dimension reduction approach", 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 11-15, 2014.

[6] B. Ribeiro, N.Lopes, J. Goncalves, "Signature identification via efficient feature selection and GPU-based SVM classifier", IEEE International Joint Conference on Neural Networks (IJCNN), pp. 1138-1145, 2014.

[7] J. B. Tenenbaum, V. de Silva and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction", Science, vol.290, no.5500, pp.2319-2323, 2000.

[8] H. Kim, H. Park and H. Zha, "Distance Preserving Dimension Reduction for Manifold Learning", Society for Industrial and Applied Mathematics Int. Conf. Data Mining, vol II, pp. 1147–1151, 2007.

[9] Daniel J. Navarro and Michael D. Lee, "Spatial Visualization of Document Similarity", Defence Human Factors. Special Interest Group Meeting, 2001.

[10] D. Zhang, X. Chen, and W. Lee, "Text Classification with Kernels on the Multinomial Manifold", SIGIR 2005, pp. 266-273, 2005.

[11] Hwanjo Yu, "SVM Selective Sampling for Ranking with Application to Data Retrieval", KDD 2005, pp. 354–363.

[12] V. Vapnik, "The Nature of Statistical Learning Theory", 2nd ed, Springer, 1999.

[13] T. Joachims, "Transductive Inference for Text Classification using Support Vector Machines", *Proceedings of the 16th ICML*, pp. 200-209, 1999.

[14] A. Blum, T. Mitchell, "Combining Labeled and Unlabeled Data with Co-Training", *Conference on Computational Learning Theory*, pp. 92-100, 1998.

[15] D. Lewis, W. Gale, "A sequential algorithm for training text classifiers", *ACM-SIGIR*, pp. 3-12, 1994.

[16] G. Schohn, D. Cohn, "Less is more: Active Learning with Support Vector Machines", *Proceedings of the Proceedings of the 17th International Conference on Machine Learning*, pp. 839-846, 2000.

[17] S. Tong, D. Koller, "Support vector machine active learning with applications to text classification", *JMLR*, vol. 2, pp. 45-66, 2001.

[18] I. T. Jolliffe, "Principal Component Analysis", New York: Springer, 1986.

[19] P. Comon, "Independent Component Analysis: a New Concept?", Signal Processing, vol.36(3), pp.287-314, 1994.

[20] T. Cox and M. Cox, "Multidimensional Scaling", London: Chapman & Hall, 1994.

[21] Isabelle Guyon and Steve Gunn and Masoud Nikravesh and Lofti Zadeh, "Series Studies in Fuzziness and Soft Computing", Physica-Verlag, Springer, 2006.

[22] Feiping Nie, Dong Xu, Tsang I.W.-H., Changshui Zhang, "Flexible Manifold Embedding: A Framework for Semi-Supervised and Unsupervised Dimension Reduction", IEEE Transactions on Image Processing, 19(7), pp. 1921-1932, 2010.

[23] Zhang, Zhenyue, Wang, Jing, Zha, Hongyuan, Zhejiang, "Adaptive Manifold Learning", IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(2), pp. 253 - 265, 2012.

[24] Ramani Duraiswami and Vikas C. Raykar, "The Manifolds of Spatial Hearing", ICASSP 2005, vol. III, pp. 285–288, 2005.

[25] X. Geng, De.n Zhan, and Z. Zhou, "Supervised Nonlinear Dimensionality Reduction for Visualization and Classification", IEEE Transactions Systems, Man, and Cybernetics - Part B, 35(6), pp. 1098-1107, 2005.

[26] Donald Specht, "A General Regression Neural Network". IEEE Transactions on Neural Networks, 2(6), pp. 568-576, 1991.

[27] F. Sebastiani,"Classification of Text, Automatic", *The Encyclopedia of Language and Linguistics*, In Keith Brown (ed.), Volume 14, 2nd Edition, Elsevier, 2006.

[28] S. Eyheramendy, A. Genkin, W. Ju, D. Lewis, D. Madigan, "Sparse Bayesian Classifiers for Text Classification", *Journal of Intell. Community R&D*, 2003.

[29] Y. Yang, J. Zhang, B. Kisiel, "A Scalability Analysis of Classifiers in Text Categorization", *SIGIR '03*, ACM Press, pp. 96-103, 2003.

[30] C. Apté, F. Damerau, S. Weiss, "Automated Learning of Decision Rules for Text Categorization", *ACM Trans. for Information Sys.*, vol. 12, pp. 233-251, 1994.

[31] S. Kaski and J. Nikkilä and M. Oja and J. Vennaand P. Törönen and E. Castrén, "Trustworthiness and Metrics in Visualizing Similarity of Gene Expression", BMC Bioinformatics, 4(48).

[32] J. Venna and S. Kaski, "Local Multidimensional Scaling with Controlled Tradeoff between Trustworthiness and Continuity", Workshop of Self-Organizing Maps, pp. 695–702, 2005.

[33] C. Silva and B. Ribeiro, "On Text-based Mining with Active Learning and Background Knowledge using SVM", Journal of Soft Computing, Springer, 11(6), pp. 519–530, 2007.

[34] C. van Rijsbergen, "Information Retrieval", Butterworths Ed., 1979.

[35] Miguel Ruiz, Padmini Srinivasan, "Automatic Text Categorization and Its Application to Text Retrieval", *IEEE Tran. Know. Data Eng.*, 11(6), pp. 865-879, 1999.