RANKING USING TRANSITION PROBABILITIES LEARNED FROM MULTI-ATTRIBUTE DATA

Sigurd Løkse and Robert Jenssen

Machine Learning Group: site.uit.no/ml
Department of Physics and Technology, UiT - The Arctic University of Norway

ABSTRACT

In this paper, as a novel approach, we learn Markov chain transition probabilities for ranking of multi-attribute data from the inherent structures in the data itself. The procedure is inspired by consensus clustering and exploits a suitable form of the PageRank algorithm. This is very much in the spirit of the original PageRank utilizing the hyperlink structure to learn such probabilities. As opposed to existing approaches for ranking multi-attribute data, our method is not dependent on tuning of critical user-specified parameters. Experiments show the benefits of the proposed method.

Index Terms— Ranking, multi-attribute data, transition probabilities, similarity measure, parameter free

1. INTRODUCTION

In data analysis, ranking is a procedure where we seek a natural order of the data points. Ranking is relevant e.g. for web pages [1, 2], images [3], text documents [4], and general information networks [5]. Some methods for ranking multi-attribute data exist [6, 7]. However, these methods have severe shortcomings. They depend heavily on sensitive and user-specified parameters and are computing transition probabilities via similarities between multi-attribute data points to be ranked in a static and unflexible way.

In this paper, we take a completely different approach. Inspired by the vast success of ensemble learning [8, 9, 10, 11], we propose to truly *learn* Markov chain transition probabilities from the data itself, by examining the data in an iterative clustering procedure over a wide range of resolutions or scales. Learning similarities from the data itself is very much in the same spirit as the original formulation of the PageRank [1], where transition probabilities were effectively learned from the hyperlink structure of the internet. Our novel procedure requires no tuning of critical hyperparameters, and is shown in experiments to perform very well compared to alternatives on a range of problems.

We gratefully acknowledge the support of NVIDIA Corporation. This work was partially funded by the Norwegian Research Council FRIPRO grant no. 239844 on developing the *Next Generation Learning Machines*.

2. BACKGROUND ON THE PERSONALIZED PAGERANK

The Personalized PageRank (PPR) is a variant of the PageRank algorithm [1], which enables personalization to queries. Chung and Zhao derived a variant of the PPR to be used as a mathematical framework for studying relationships between the PPR and various graph invariants [12]. In this work, we exploit this formulation of the PPR since we recognize it as especially suitable for ranking of multi-attribute data given a symmetric similarity measure. Combined with our method for learning similarities between data points, and hence transition probabilities as explained below (see Sec. 3), this leads to a novel approach for ranking multi-attribute data.

Consider the difference equation

$$\mathbf{r}_{k+1}^T = (1-\alpha)\mathbf{r}_k^T\mathbf{P} + \alpha\mathbf{s}^T, \tag{1}$$

where $\mathbf{P}$ is a right stochastic matrix, $0 < \alpha < 1$ is the restart probability and $\mathbf{s} = \{s_i\}_{N\times 1},\ \sum_{i=1}^N s_i = 1$ is the seed distribution. This difference equation converges to the stationary distribution of the Markov chain associated with the transition probability matrix $\mathbf{P}' = (1-\alpha)\mathbf{P} + \alpha\mathbb{1}\mathbf{s}^T$. By defining $\mathbf{P} = \mathbf{D}^{-1}\mathbf{K}$, where $\mathbf{D} = \mathrm{diag}(d_i),\ d_i = \sum_{j=1}^N k_{ij}$ and $\mathbf{K} = \{k_{ij}\}_{N\times N}$ is a symmetric similarity matrix with positive elements representing a connected graph, one can show that the difference equation in (1) converges to

$$\boldsymbol{\pi}(\alpha, \mathbf{s}) = \beta\mathbf{D}\mathbf{G}_\beta\mathbf{s}. \tag{2}$$

Here, $\beta = \frac{\alpha}{1-\alpha}$ and $\mathbf{G}_\beta$ is the inverse of the β-adjusted Laplacian $\mathbf{L}_\beta = \beta\mathbf{D} + \mathbf{L}$, where $\mathbf{L} = \mathbf{D} - \mathbf{K}$.

3. RANKING USING TRANSITION PROBABILITIES LEARNED FROM DATA

A highly novel aspect of this paper is to *learn* the transition probabilities for ranking of *multi-attribute data* in a near fully automated way, without the selection of critical hyperparameters, such as the width when using the fixed RBF kernel [6, 13]. This reflects in a sense the original PageRank for ranking web-pages, where transition probabilities were effectively learned from the link structure.

Inspired by consensus clustering [14], we *learn* the similarity matrix $\mathbf{K}_L$ for building $\mathbf{P}_L$ such that the similarity measure adapts to the inherent structures in the data, both on local and global scales. This is achieved by fitting Gaussian mixture models (GMMs) to the data over a range of resolutions $g = 2, 3, \dots G$ (number of mixture components), providing both a local and a global view of the data. This is done for $q = 1, 2, \dots, Q$ initial conditions. Using the EM-algorithm [15], the posterior distribution $\boldsymbol{\gamma}_i(q, g)$ of data point $\mathbf{x}_i$ is computed. Then, the learned similarity matrix is defined as

$$K_L(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{Z} \sum_{q=1}^{Q} \sum_{g=2}^{G} \boldsymbol{\gamma}_i(q, g)^T \boldsymbol{\gamma}_j(q, g), \qquad (3)$$

where Z is a normalizing constant. This matrix is inspired by [16], where hard cluster memberships were used for semi-supervised learning, and is also referred to as the probabilistic cluster kernel (PCK) [17]. *Here, it is for the first time used for ranking.*

An important motivation for proposing such a similarity measure to learn transition probabilities for ranking is the novel interpretation we present next in this paper, which adds interpretability to the general framework.

3.1. Relating the learned similarity matrix to consensus clustering

Assume that the number of mixture components, $\mathcal{G}$, and the initial condition $\mathcal{Q}$ is independently drawn from the distributions $P(\mathcal{G})$ and $P(\mathcal{Q})$. Let $\mathcal{Y}_i = y$ if data point $\mathbf{x}_i$ is drawn from mixture component y. Then

$$\boldsymbol{\gamma}_i(q, g) = \begin{pmatrix} P_{\mathcal{Y}_i=1|q,g} & P_{\mathcal{Y}_i=2|q,g} & \cdots & P_{\mathcal{Y}_i=g-1|q,g} \end{pmatrix}^T,$$

where, $P_{\mathcal{Y}_i=y|q,g} = P(\mathcal{Y}_i = y|\mathcal{Q} = q, \mathcal{G} = g)$. This is justified since we implicitly condition on the initial condition and the number of mixture components when calculating the parameters in the GMM. Assuming that $\mathbf{x}_i$ and $\mathbf{x}_j$ are drawn independently from the mixture components[1], we get

$$\boldsymbol{\gamma}_i(q, g)^T \boldsymbol{\gamma}_j(q, g) = \sum_{y=1}^{g-1} P_{\mathcal{Y}_i=y|q,g} P_{\mathcal{Y}_j=y|q,g} = P_{\mathcal{Y}_i=\mathcal{Y}_j|q,g}.$$

Furthermore,

$$\begin{aligned} P(\mathcal{Y}_i = \mathcal{Y}_j) &= \sum_{q=1}^{Q} P_q P_{\mathcal{Y}_i=\mathcal{Y}_j|q} = \sum_{q=1}^{Q} \sum_{g=2}^{G} P_q P_g P_{\mathcal{Y}_i=\mathcal{Y}_j|q,g} \\ &= \sum_{q=1}^{Q} \sum_{g=2}^{G} P_q P_g \boldsymbol{\gamma}_i(q, g)^T \boldsymbol{\gamma}_j(q, g). \end{aligned}$$

[1]This assumption is satisfied on the off-diagonal elements of $\mathbf{K}_L$.

Assuming that $\mathcal{Q}$ and $\mathcal{G}$ are uniformly distributed such that $P_q P_g = \frac{1}{Q(G-1)}$ yields

$$\begin{aligned} P(\mathcal{Y}_i = \mathcal{Y}_j) &= \frac{1}{Q(G-1)} \sum_{q=1}^{Q} \sum_{g=2}^{G} \boldsymbol{\gamma}_i(q, g)^T \boldsymbol{\gamma}_j(q, g) \\ &= \mathbf{K}_L(\mathbf{x}_i, \mathbf{x}_j) \end{aligned} \qquad (4)$$

using the normalization constant $Z = \frac{1}{Q(G-1)}$. Thus, each element of $\mathbf{K}_L$ calculates the probability that two data points are drawn from the same mixture component, i.e. the probability that two data points belonging to the same cluster. From this, we define our learned transition probabilities as $\mathbf{P}_L = \mathbf{D}^{-1}\mathbf{K}_L$.

Calculating $\mathbf{P}_L$ includes estimating covariance matrices and inverting them. For high dimensional data ($d > N$), we recommend using an SVD to reduce the dimensionality of the data to avoid singular covariance matrices.

One should notice that each run of the EM-algorithm is calculated independently of the others. Thus, Eq. (3) can be computed in parallel. Note furthermore that the only parameters in this procedure are G and Q. *The exact choice of these parameters are however not critical for the performance* as long as sufficiently high values are used, since $\mathbf{K}_L$ adapts to the structures in the data set on both local scales (large G) and global scales (small G). In experiments not shown here due to space limitations, we have varied Q and G over a wide range of values, showing no significant difference for $Q, G >\approx 20$. Thus, in all experiments and for all different data sets used in this paper, we fix $Q = G = 20$.

3.2. Ranking algorithm

The algorithm for ranking multivariate data using $\mathbf{K}_L$ is as follows:

1. Construct the learned similarity matrix $\mathbf{K}_L$ using (3).
2. Sort the weights between pairwise nodes in descending order. Create a graph by connecting pairwise nodes from the sorted list successively until the graph is connected (see [6]). Connectivity in the graph can be verified by e.g. a depth-first search [18]. The main diagonal is set to zero.
3. Rank the data by using the PPR according to (2).

Similar approaches have been used for other similarity measures (see e.g. [6]).

4. EXPERIMENTS

In the following experiments, we validate the performance of ranking using the learned similarity matrix. We compute the Area Under the Curve (AUC) of the Receiver Operator Characteristics (ROC) curve for data with known labels, as done

in [6]. This is computed by using the scores for a given query as a probability of data points belonging to the positive class for a given query. The query is always sampled from the positive class. For data with group structures, but without known labels, we generate labels by cluster analysis.

We compare our new method with the state-of-the-art algorithm *ranking on data manifolds* [6], which uses an RBF as the similarity measure. The RBF is defined as

$$K_{\text{RBF}}(\mathbf{x}_i, \mathbf{x}_j) = e^{-\frac{1}{2\sigma^2}\|\mathbf{x}_i - \mathbf{x}_j\|^2}.$$

The RBF examines the data on one scale of resolution only, determined by σ, which significantly affects results. To show that our $\mathbf{K}_L$ is extremely robust, we fix $Q = G = 20$ for all experiments. Note that since ranking is an unsupervised learning problem, the width parameter σ of the RBF has to be pre-computed according to some criterion. In this paper we follow the widespread practice of setting σ equal to 15% of the median pairwise distances in the data set (see e.g. [19]). The restart probability is set to $\alpha = 0.15$, supposedly the same value used by Google for web page ranking [1].

4.1. Synthetic data

In our first experiment, the aim is to illustrate that the learned transition probabilities capture nonlinear and complex structure in the data. Towards this end, we study Jain's synthetic so-called two moon data set [8], as shown in Fig. 1a. The data set contains two classes, separable by a non-linear boundary.

Fig. 1b shows the ROC curves for $\mathbf{K}_L$ (blue) and the RBF (green) for a single query. The learned similarity matrix outperforms the RBF with an AUC of 1.0 and 0.60, respectively. A visualization of the ranking results is shown in Fig. 1c for $\mathbf{K}_L$ and Fig. 1d for the RBF. The size of the circles represents the ranking score for the data point. The query is represented by a red cross. From these plots, we see that ranking based on $\mathbf{K}_L$ is able to follow the structure of the data within the classes, with the result that every data point from the positive class have a larger ranking score than the data points from the negative class. The RBF however cannot capture such structure leading to data points close to the query but in the negative class having larger ranking scores than distant data points in the positive class. This illustrates the benefits of learning similarities from the inherent structures of the data, as opposed to the non-adaptive RBF.

4.2. Text document ranking

In this experiment, we rank documents from a subset of 1000 documents from the 20-newsgroups data set[2]. Fig. 2a shows the AUC obtained from 100 random queries in in this data set when using $\mathbf{K}_L$ versus an RBF. The gray line indicates equal performance. With a mean AUC of 0.70 for $\mathbf{K}_L$ and 0.53 for

[2]http://www.cs.nyu.edu/roweis/data/20news_w100.mat

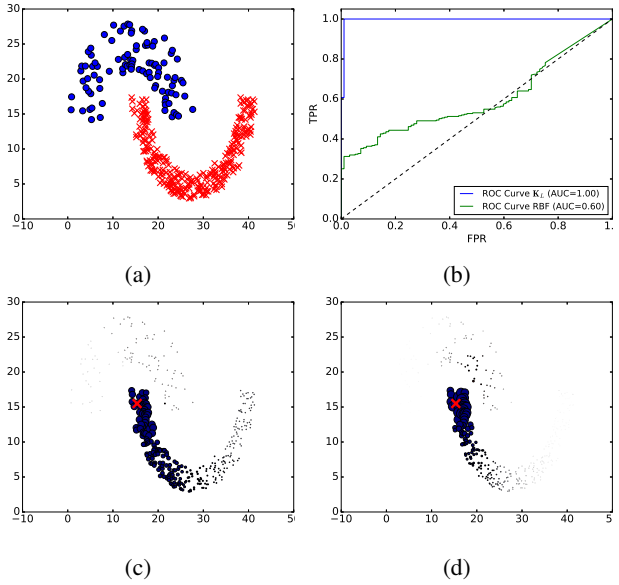


Fig. 1: **(a):** Jain's two moon data set. **(b):** ROC Curve when using $\mathbf{K}_L$ (blue) and an RBF (green). **(c)–(d)**: Plot of the two moon data set with ranking results for $\mathbf{K}_L$ (**(c)**) and an RBF (**(d)**). The size of the circles represent the score from the ranking. The red cross represents the query. We see that $\mathbf{K}_L$ is more capable of following the structure within the class of the query.

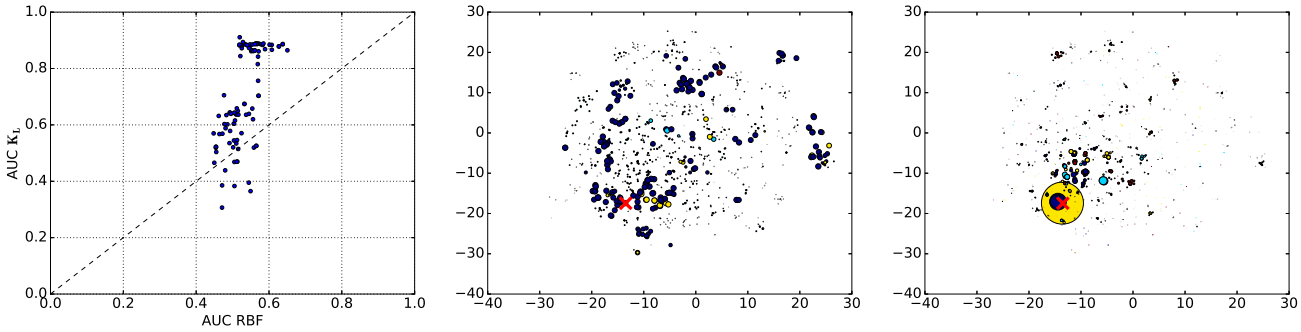
the RBF, $\mathbf{K}_L$ performs better overall than the RBF. This is easily seen in the figure, where most of the points are above the gray line.

Fig. 2b and Fig. 2c shows an embedding of the data using t-SNE [20]. The colors represents the ground truth classes, while the size of the dots represents the score for a query given by the red cross. In the plot for $\mathbf{K}_L$, we see that the main mass of high score documents comes from the dark blue class. In the RBF plot, we have a mix of classes in the highly ranked documents. The highest ranked document is from the yellow class. Further inspection shows that the query document is a member of the dark blue class, indicating that $\mathbf{K}_L$ is in fact able to encode similarities much better than the RBF, and in a meaningfull manner.

Tab. 1 shows the words in the top ranked documents from a query document with the words $\{$*data, display, email*$\}$. Blue words are from $\mathbf{K}_L$ (top), while red words are from an RBF (bottom). From this, it seems like $\mathbf{K}_L$ might be able to capture the *semantics* of the documents, while the RBF only considers distance between documents.

4.3. Image ranking

In this experiment, we rank 1000 images from the Frey Face data set. Although there are no ground truth labels for this data set, previous work have shown that there are group struc-

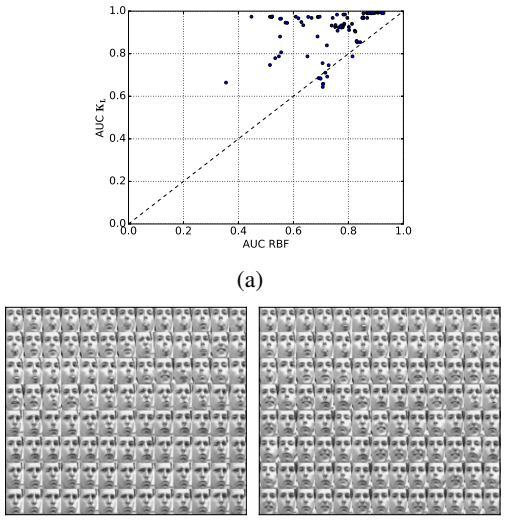


(a) (b) (c)

Fig. 2: Results for the 20 newsgroups data set. **(a)**: Scatterplot of the AUC for the newsgroups data set when using $\mathbf{K}_L$ versus an RBF with 100 random queries. The grey line represents equal performance for both. **(b)–(c)**: t-SNE of the data set for a random query with colors according to the ground truth. The red cross represents the query document, which is from the dark blue class.

Table 1: Words in the top ranked documents with the query document $\{$*data, display, email*$\}$. Blue words are from $\mathbf{K}_L$ (top), while red words are from an RBF (bottom).

1.	program, software, system, university, version, windows data, display
2.	data, files, memory, program, system, win, windows display, help
3.	system, win, windows display, problem
4.	disk, files, help, program, version display, help, windows
5.	help, problem, question, win, windows data



(a)

(b) (c)

Fig. 3: Results for the Frey Face experiment. **(a)**: Scatterplot of the AUC for the Frey Face data set. **(b)–(c)**: Example ranking for $\mathbf{K}_L$ (**(b)**) and an RBF (**(c)**).

tures in the data [19]. Thus, to have something to compare the algorithm with, we cluster the data using k-means with 3 clusters, the same number of clusters used in [19], and use the cluster labels as an estimate for ground truth. The results are shown in Fig. 3

Fig. 3a shows the AUC obtained when using $\mathbf{K}_L$ versus an RBF from 100 random queries with an average AUC of 0.92 and 0.76 for $\mathbf{K}_L$ and the RBF, respectively. We see that, except for a few queries, the AUC obtained when using $\mathbf{K}_L$ is larger than when using an RBF.

Fig. 3b and Fig. 3c shows the list for a random query for $\mathbf{K}_L$ and an RBF respectively. The query is shown as the top left image. In the beginning of the list, the two similarity measures seem to behave similarly. Later in the list, $\mathbf{K}_L$ is more consistent and seems to traverse along some manifold from one facial expression to another. With the RBF, the latter part of the list seems to contain a mix of different facial expressions.

5. CONCLUSION

In this paper, we have proposed a similarity measure for ranking multi-attribute data that is robust and does not need parameter tuning. When coupling this with ranking methods based on Markov chains, this similarity measure is effectively used to learn Markov chain transition probabilities from data. In the experiments, we have shown its robustness by letting all parameters be fixed over a range of data sets. The experiments have shown superior results, compared to a standard RBF.

Note that even though we have used the personalized PageRank in this paper, this similarity measure can be used with any ranking algorithm that assumes a symmetric similarity measure.

6. REFERENCES

[1] S. Brin and L. Page, "The anatomy of a large-scale hypertextual Web search engine," *Computer Networks and ISDN Systems*, vol. 30, no. 1-7, pp. 107–117, Apr. 1998.

[2] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *Journal of the ACM (JACM)*, vol. 46, no. 5, pp. 604–632, 1999.

[3] Y. Jing and S. Baluja, "VisualRank: applying PageRank to large-scale image search.," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 11, pp. 1877–90, Nov. 2008.

[4] J. Carbonell and J. Goldstein, "The use of mmr, diversity-based reranking for reordering documents and producing summaries," in *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1998, pp. 335–336.

[5] Y. Sun, J. Han, P. Zhao, Z. Yin, H. Cheng, and T. Wu, "RankClus: Integrating Clustering with Ranking for Heterogeneous Information Network Analysis," *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*, pp. 565–576, 2009.

[6] D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Schölkopf, "Ranking on Data Manifolds," *Advances in Neural Information Processing Systems*, vol. 16, pp. 169–176, 2004.

[7] L. Cao, A. Del Pozo, X. Jin, J. Luo, J. Han, and T. S. Huang, "RankCompete: Simultaneous Ranking and Clustering of Web Photos," *Proceedings of the 19th international conference on World wide web*, pp. 1071–1072, 2010.

[8] A. L. N. Fred and A. K. Jain, "Data clustering using evidence accumulation," in *Pattern Recognition, 2002. Proceedings. 16th International Conference on*. IEEE, 2002, vol. 4, pp. 276–280.

[9] A. Strehl and J. Ghosh, "Cluster Ensembles — a Knowledge Reuse Framework for Combining Multiple Partitions," *The Journal of Machine Learning Research*, vol. 3, pp. 583–617, 2003.

[10] P. Hore, L. O. Hall, and D. B. Goldgof, "A scalable framework for cluster ensembles," *Pattern recognition*, vol. 42, no. 5, pp. 676–688, 2009.

[11] H. Liu, J. Wu, T. Liu, D. Tao, and Y. Fu, "Spectral ensemble clustering via weighted k-means: Theoretical and practical evidence," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 5, pp. 1129–1143, 2017.

[12] F. Chung and W. Zhao, "Pagerank and random walks on graphs," *Fete of combinatorics and computer science*, pp. 1–16, 2010.

[13] H. J. Qiu and E. R. Hancock, "Clustering and embedding using commute times.," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 11, pp. 1873–90, Nov. 2007.

[14] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*, CRC Press, 2012.

[15] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society*, vol. 39, no. 1, pp. 1–38, 1977.

[16] J. Weston, C. Leslie, E. Ie, D. Zhou, A. Elisseeff, and W. S. Noble, "Semi-supervised protein classification using cluster kernels.," *Bioinformatics (Oxford, England)*, vol. 21, no. 15, pp. 3241–7, Aug. 2005.

[17] E. Izquierdo-Verdiguier, R. Jenssen, L. Gómez-Chova, and G. Camps-Valls, "Spectral clustering with the probabilistic cluster kernel," *Neurocomputing*, vol. 149, Part C, no. 0, pp. 1299–1304, 2015.

[18] R. Sendgewick, *Algorithms in C++ Part 5: Graph Algorithms*, Addison–Wesley Professional, 3th edition, 2002.

[19] R. Jenssen, "Kernel Entropy Component Analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 5, pp. 847–860, 2010.

[20] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.