On the Computation of the Geodesic Distance with an Application to Dimensionality Reduction in a Neuro-Oncology Problem

Raúl Cruz-Barbosa[1], David Bautista-Villavicencio[1], and Alfredo Vellido[2]

[1] Universidad Tecnológica de la Mixteca, 69000, Huajuapan, Oaxaca, México
{rcruz,dbautista}@mixteco.utm.mx
[2] Universitat Politècnica de Catalunya, 08034, Barcelona, Spain
avellido@lsi.upc.edu

Abstract. Manifold learning models attempt to parsimoniously describe multivariate data through a low-dimensional manifold embedded in data space. Similarities between points along this manifold are often expressed as Euclidean distances. Previous research has shown that these similarities are better expressed as geodesic distances. Some problems concerning the computation of geodesic distances along the manifold have to do with time and storage restrictions related to the graph representation of the manifold. This paper provides different approaches to the computation of the geodesic distance and the implementation of Dijkstra's shortest path algorithm, comparing their performances. The optimized procedures are bundled into a software module that is embedded in a dimensionality reduction method, which is applied to MRS data from human brain tumours. The experimental results show that the proposed implementation explains a high proportion of the data variance with a very small number of extracted features, which should ease the medical interpretation of subsequent results obtained from the reduced datasets.

1 Introduction

The choice of a type of distance as a similarity measure is relevant in many supervised, unsupervised and semi-supervised machine learning tasks [1]. For real-valued data, the Euclidean distance is the most common choice due to its intuitive understanding and the simplicity of its computation. In manifold learning, though, the Euclidean distance has been shown not always to be the most adequate choice to measure the (dis)similarity between two data points [2,3,4]. This is most relevant when working with data that reside in a high-dimensional space of which we ignore the intrinsic geometry, a common situation in biomedicine or bioinformatics.

An alternative distance function that may alleviate the previously mentioned problem is the geodesic distance, since it measures similarity along the embedded manifold, instead of doing it through the embedding space. Unlike the Euclidean distance, the geodesic one follows the geometry of the manifold that models the

C. San Martin and S.-W. Kim (Eds.): CIARP 2011, LNCS 7042, pp. 483–490, 2011.

data. In this way, it may help to avoid some of the distortions (such as breaches of topology preservation) that the use of a Euclidean metric may introduce when learning the manifold (due to undesired manifold curvature effects).

Manifold learning methods that use geodesic distances can be categorized, according to their main task, as unsupervised [2,4,5] and semi-supervised. The first semi-supervised methods used for classification task were reported in [6] and [7]. These methods, as well as many others that involve the geodesic distance [8], are known as graph-based methods. Most of them compute the data point pairwise distance of a graph using the basic Dijkstra algorithm, as well as use a full data matrix representation for finding the shortest path between them. This may lead to computational time and storage problems. The current study provides different approaches to the computation of the geodesic distance and the implementation of Dijkstra's shortest path algorithm, comparing their performances.

The best performing methods are bundled in a software module that is inserted in a nonlinear dimensionality reduction (NLDR) method, namely ISOMAP [2], which is then applied to the analysis of magnetic resonance spectroscopy (MRS) data from human brain tumours. The performance of the proposed method is compared to that of the original ISOMAP implementation.

2 Geodesic Distances

The explicit calculation of geodesic distances can be computational impractical. This metric, though, can be approximated by graph distances [9], so that instead of finding the minimum arc-length between two data points lying on a manifold, we would set to find the shortest path between them, where such path is built by connecting the closest successive data points. In this paper, this is done using the K-rule, which allows connecting the K-nearest neighbours. A weighted graph is then constructed by using the data and the set of allowed connections. The data are the vertices, the allowed connections are the edges, and the edge labels are the Euclidean distances between the corresponding vertices. If the resulting graph is disconnected, some edges are added using a minimum spanning tree procedure in order to connect it. Finally, the distance matrix of the weighted undirected graph is obtained by repeatedly applying Dijkstra's algorithm [10], which computes the shortest path between all data samples. For illustration, this process is graphically represented in Fig. 1.

2.1 Computation of the Geodesic (Graph) Distance

There are different implementation alternatives for some of the stages involved in the geodesic distance computation (see Fig. 1). This computation is constrained by the type of graph representation of the dataset and by the chosen shortest path algorithm. Two alternatives for graph representation are the *adjacency matrix* and the *adjacency list*. The former consists in a n by n matrix structure, where n is the number of vertices in the graph. If there is an edge from a vertex

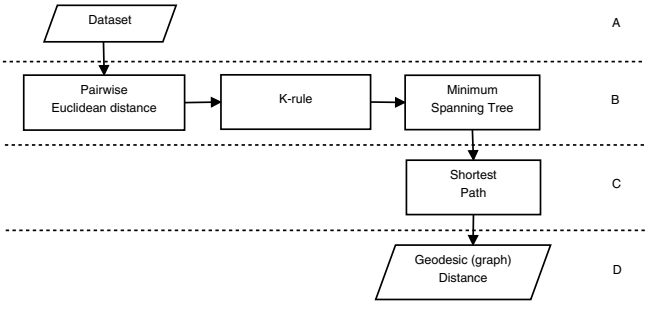

Fig. 1. Graph distance procedure scheme. Stage (A) represents the input data. Stage (B) is for building the weighted, undirected, connected graph. Stage (C) is for computing the geodesic (graph) distance, which is returned in Stage (D).

i to a vertex j, then the element a_{ij} is 1, otherwise it is 0. This kind of structure provides faster access for some applications but can consume huge amounts of memory. The latter considers that each vertex has a list of which vertices it is adjacent to. This structure is often preferred for sparse graphs as it has smaller memory requirements.

On the other hand, three options (of several) for the shortest path algorithm are: (basic) Dijkstra, Dijkstra using a Fibonacci heap (F-heap) and Floyd-Warshall. All of them assume that the graph is a weighted, connected graph. The time complexity of the simplest implementation of Dijkstra's algorithm is $O(|V|^2)$, using the Big-O notation. For some applications where the obtained graph is a sparse graph, Dijkstra's algorithm can save memory resources by storing the graph in the form of adjacency list and using an F-heap as a priority queue to implement extracting minimum efficiently. In this way, the time complexity of the algorithm can be improved to $O(|E| + |V|log|V|)$.

An F-heap is a binary tree with the property that, for every subtree, the root is the minimum item. This data structure is widely used as priority queue [11]. The priority queues are used to keep a dynamic list of different priorities jobs. An F-heap allows several operations as, for instance, ***Insert()***, which adds a new job to the queue and ***ExtractMin()***, which extracts the highest priority task.

Another approach for computing the shortest path is provided by the Floyd-Warshall algorithm, which is an example of dynamic programming. It finds the lengths of the shortest paths between all pairs of vertices. Unlike Dijkstra's algorithm which assumes that all weights are positive, this algorithm can deal with positive or negative edge weights. Its complexity is $O(|V|^3)$.

3 Experiments

The goal of the experiments herein reported is twofold. Firstly, we aim to assess which combination of graph representation and shortest path algorithm produces the best time performance for computing the geodesic distance for datasets with

increasing numbers of items. Secondly, the software implementation of the best found solution is inserted in the NLDR ISOMAP algorithm. Its performance is compared to that of the original Tenenbaum's implementation (basic and landmark versions) and to standard Principal Component Analysis (PCA), in terms of the amount of explained variance as a function of the number of new features extracted. We hypothesize that the connected graph built through the proposed procedure adds more geometric information to ISOMAP than the largest connected component found by the original version.

The experiments were carried out setting the K parameter to a value of 10, in order to get a connected graph when the K-rule is applied. After that, K was set to 1 for gauging the time performance of the geodesic distance computation when graph is sparse and unconnected. All experiments were performed using a dual-processor 2.3 Ghz BE-2400 desk PC with 2.7Gb RAM.

3.1 UCI Datasets and MRS Brain Tumour Database

Five datasets from the UCI machine learning repository [12], with increasing number of items, were used for the experiments. They are: *Ecoli* (336 7-dimensional points belonging to 8 classes representing protein location sites); *German* (1000 24-dimensional data points belonging to good or bad credit risks; *Segmentation* (2,310 19-dimensional items representing several measurements of image characteristics belonging to seven different classes); *Pageblocks* (5,473 items described by 10 attributes, concerning block measurements of distinct documents corresponding to five classes); and *Pendigits* (10,992 16-dimensional items corresponding to (x, y) tablet coordinate information measurements, which belong to ten digits).

We also experiment with MRS data acquired at different echo times (short -STE- and long -LTE-), as well as with a combination of both. Data belong to a multi-center, international database [13], and consist of: (1) 217 STE spectra, including 58 meningiomas (mm), 86 glioblastomas (gl), 38 metastases (me), 22 astrocytomas grade II (a2), 6 oligoastrocytomas grade II (oa), and 7 oligodendrogliomas grade II (od); (2) 195 LTE spectra, including 55 mm, 78 gl, 31 me, 20 a2, 6 oa, and 5 od. (3) 195 items built by combination (through direct concatenation) of the STE and LTE spectra for the same patients. Only the clinically relevant regions of the spectra were analyzed. They consist of 195 frequency intensity values (measured in parts per million (ppm), an adimensional unit of relative frequency position in the data vector), starting at 4.25 ppm. These frequencies become the observed data features.

3.2 Results and Discussion

The time performance results for computing geodesic (graph) distances, using $K = 10$, are shown in Table 1. Here, a combination of adjacency matrix for graph representation and basic Dijkstra as the choice for shortest path algorithm outperformed the other combinations, except for *Pageblocks*. This is due to the faster access to elements in an adjacency matrix when basic Dijkstra's

Table 1. Time performance results for the computation of geodesic (graph) distances (assuming a connected graph by setting $K = 10$) for several UCI datasets and different settings. The '–' symbol indicates that the memory limit was exceeded.

Dataset (# items)	**Shortest path**	**Representation**	**Time (s)**
Ecoli (336)	Dijkstra	Adjacency Matrix	0.43
	Dijkstra+F-heaps	Adjacency Matrix	1.19
	Floyd-Warshall	Adjacency Matrix	0.53
	Dijkstra	Adjacency List	0.67
	Dijkstra+F-heaps	Adjacency List	1.59
	Floyd-Warshall	Adjacency List	0.42
German (1000)	Dijkstra	Adjacency Matrix	12.43
	Dijkstra+F-heaps	Adjacency Matrix	25.03
	Floyd-Warshall	Adjacency Matrix	23.67
	Dijkstra	Adjacency List	16.18
	Dijkstra+F-heaps	Adjacency List	38.39
	Floyd-Warshall	Adjacency List	18.71
Segmentation (2310)	Dijkstra	Adjacency Matrix	185.57
	Dijkstra+F-heaps	Adjacency Matrix	297.31
	Floyd-Warshall	Adjacency Matrix	347.16
	Dijkstra	Adjacency List	229.83
	Dijkstra+F-heaps	Adjacency List	511.59
	Floyd-Warshall	Adjacency List	292.89
Pageblocks (5473)	Dijkstra	Adjacency Matrix	3621.90
	Dijkstra+F-heaps	Adjacency Matrix	4031.93
	Floyd-Warshall	Adjacency Matrix	18369.84
	Dijkstra	Adjacency List	3585.92
	Dijkstra+F-heaps	Adjacency List	8039.92
	Floyd-Warshall	Adjacency List	10409.90
Pendigits (10992)	Dijkstra	Adjacency Matrix	––
	Dijkstra+F-heaps	Adjacency Matrix	––
	Floyd-Warshall	Adjacency Matrix	––
	Dijkstra	Adjacency List	124363.18
	Dijkstra+F-heaps	Adjacency List	66105.34
	Floyd-Warshall	Adjacency List	204604.99

algorithm required them. It is worth noting how the time performance for the adjacency list representation and Dijkstra is better for larger datasets. This effect is pronounced for *Pendigits*, with which the matrix representation can not deal due to the storage restrictions of the operating system (it dedicates approximately 700 Mb for each process). In this case, the best combination is the adjacency list and Dijkstra using F-heaps. Now, using the matrix representation, and if time results are compared for Dijkstra and Dijkstra using F-heaps algorithms, we observe that the time proportion decreases when number of items increases; this difference is more pronounced for Dijkstra implemented with F-heaps. This tendency is not maintained for the list representation using small and medium datasets, but it is notably low for large datasets as *Pendigits*. Thus, it can be inferred that, for large datasets, the best time performance for computing geodesic distances would be provided by an adjacency list (or matrix, when storage restrictions are discarded) representation and Dijkstra using F-heaps. The opposite occurs for the Floyd-Warshall algorithm independently from the graph representation. Its performance is good only for small sets.

Now, the K parameter for the K-rule is set to 1, in order to show the time performance when the procedure is dealing with an unconnected and sparse graph (see Table 2). The pattern found in the results reported in Table 1 is

Table 2. Time performance results for the computation of geodesic (graph) distances (assuming an unconnected, sparse graph by setting $K = 1$) for several UCI datasets and different settings. The '–' symbol indicates that the memory limit was exceeded.

Dataset (# items)	**Shortest path**	**Representation**	**Time (s)**
Ecoli (336)	Dijkstra	Adjacency Matrix	0.47
	Dijkstra+F-heaps	Adjacency Matrix	1.21
	Floyd-Warshall	Adjacency Matrix	0.6
	Dijkstra	Adjacency List	0.67
	Dijkstra+F-heaps	Adjacency List	1.57
	Floyd-Warshall	Adjacency List	0.44
German (1000)	Dijkstra	Adjacency Matrix	12.85
	Dijkstra+F-heaps	Adjacency Matrix	25.72
	Floyd-Warshall	Adjacency Matrix	23.32
	Dijkstra	Adjacency List	16.18
	Dijkstra+F-heaps	Adjacency List	37.89
	Floyd-Warshall	Adjacency List	19.27
Segmentation (2310)	Dijkstra	Adjacency Matrix	186.55
	Dijkstra+F-heaps	Adjacency Matrix	294.22
	Floyd-Warshall	Adjacency Matrix	345.38
	Dijkstra	Adjacency List	228.47
	Dijkstra+F-heaps	Adjacency List	507.53
	Floyd-Warshall	Adjacency List	192.38
Pageblocks (5473)	Dijkstra	Adjacency Matrix	3483.08
	Dijkstra+F-heaps	Adjacency Matrix	3955.05
	Floyd-Warshall	Adjacency Matrix	10867.04
	Dijkstra	Adjacency List	5549.91
	Dijkstra+F-heaps	Adjacency List	7678.91
	Floyd-Warshall	Adjacency List	10179.90
Pendigits (10992)	Dijkstra	Adjacency Matrix	––
	Dijkstra+F-heaps	Adjacency Matrix	––
	Floyd-Warshall	Adjacency Matrix	––
	Dijkstra	Adjacency List	131085.17
	Dijkstra+F-heaps	Adjacency List	67312.69
	Floyd-Warshall	Adjacency List	193720.78

maintained. In general, it is observed that the modified minimum spanning tree procedure to connect the graph does influence the time results. The larger the dataset, the less affected the Dijkstra+F-heaps connection algorithm is.

Finally, the optimized geodesic distance calculation software module, developed in C++, was embedded in the NLDR ISOMAP algorithm, herein named ISOMAP gMod. Its performance was compared to that of Tenenbaum's ISOMAP implementation and PCA. The corresponding results are shown in Table 3. It can be observed that using ISOMAP gMod helps to explain a large percentage of the data variance with far fewer extracted features than the alternative implementations. For the LTE set (195 features corresponding to spectral frequencies), even just the first extracted feature explains 80% of the data variance. Moreover, for the high-dimensional SLTE set (390 features), two extracted features suffice to explain nearly 90% of the data variance. Overall, the ISOMAP gMod implementation outperforms all alternatives according to this evaluation measure. Further experiments were conducted with versions of the datasets reduced to 20 features through prior selection. Results are reported in Table 4 and they are consistent with those in Table 3.

Table 3. Explained variance as a function of the number of extracted features. ISOMAP variants: Standard, Landmark (Land) and with the proposed optimized module (gMod). NEF stands for *number of extracted features*.

Dataset *(itemXdim)*	**DR method**	**% of variance explained by NEF** 1	2	3	4	5	6	7	8	9	10	**#Var** > 80%	% (#Var)
LTE (195 × 195)	PCA	57.82	9.89	8.32	5.36	4.97	3.54	3.25	2.61	2.16	2.09	4	81.39
	ISOMAP	58.31	12.08	9.88	4.52	3.96	2.72	2.45	2.18	2.05	1.85	3	80.28
	ISOMAP Land	58.82	10.49	7.35	4.46	4.11	3.61	3.21	3.00	2.62	2.33	4	81.11
	ISOMAP gMod	80.50	9.06	3.50	2.25	1.19	1.02	0.76	0.66	0.59	0.46	1	80.50
STE (217 × 195)	PCA	66.88	7.68	6.58	5.74	3.71	2.64	2.18	1.80	1.41	1.38	3	81.14
	ISOMAP	67.05	8.38	7.86	4.70	3.12	2.30	2.00	1.65	1.55	1.39	3	83.29
	ISOMAP Land	66.42	7.42	6.58	4.26	3.16	2.92	2.70	2.45	2.18	1.92	3	80.42
	ISOMAP gMod	78.15	8.10	3.75	3.06	2.14	1.35	1.04	0.90	0.81	0.70	2	86.24
SLTE (195 × 390)	PCA	61.61	8.28	7.10	6.02	4.16	3.40	2.77	2.58	2.14	1.94	4	83.01
	ISOMAP	65.26	9.73	7.01	3.97	3.0	2.83	2.55	2.09	1.88	1.67	3	82.00
	ISOMAP Land	66.27	9.48	4.48	4.26	3.51	3.22	2.57	2.40	1.98	1.85	3	80.23
	ISOMAP gMod	75.28	13.22	4.53	1.76	1.32	1.00	0.88	0.77	0.68	0.55	2	88.50

Table 4. Summary of the explained variance as a function of the first 20 extracted features. Legend as in Table 3

Dataset *(item×dim)*	**DR method**	**#Var**> 80%	% (#Var)
LTE (195 × 195)	PCA	6	80.85
	ISOMAP	6	80.84
	ISOMAP Land	8	81.73
	ISOMAP gMod	2	87.23
STE (217 × 195)	PCA	4	81.52
	ISOMAP	4	80.20
	ISOMAP Land	6	80.78
	ISOMAP gMod	2	83.19
SLTE (195 × 390)	PCA	6	80.64
	ISOMAP	6	82.17
	ISOMAP Land	6	80.27
	ISOMAP gMod	2	85.71

4 Conclusion

The use of the geodesic metric has been shown to be relevant in NLDR manifold learning models. Its implementation, though, is not trivial and usually requires graph approximations. The characteristics of the software implementation of such approximations may have a considerably impact on the computational requirements, but also on the final results. Experimental results have shown that the combined use of an adjacency matrix and Dijkstra algoritm is recommendable for computing geodesic distances in small and medium datasets. For larger datasets, though, the use of an adjacency list representation becomes crucial.

The NLDR ISOMAP algorithm was implemented using the proposed optimized procedures and it was used to analyze a data set of small size but high dimensionality of MRS spectra corresponding to human brain tumours. In problems concerning the diagnosis and prognosis of such tumours, the interpretability of the results is paramount. Such interpretability can be helped by dimensionality reduction procedures. The ISOMAP gMod implementation has been shown to outperform several alternatives in terms of explaining a large percentage of

the variance of these data through an extremely reduced number of features. Future research will investigate the use of this data reduction results in brain tumour diagnostic classification tasks. A comparison of ISOMAP variants with the original Euclidean model of them, metric MDS, should also be included.

Acknowledgments. Partial funding for this research was provided by the Mexican SEP PROMEP/103.5/10/5058 and the Spanish MICINN TIN2009-13895-C02-01 research projects. Authors gratefully acknowledge the former INTERPRET European project partners. Data providers: Dr. C. Majós (IDI), Dr. À. Moreno-Torres (CDP), Dr. F.A. Howe and Prof. J. Griffiths (SGUL), Prof. A. Heerschap (RU), Prof. L. Stefanczyk and Dr J. Fortuniak (MUL) and Dr. J. Calvar (FLENI); data curators: Dr. M. Julià-Sapé, Dr. A.P. Candiota, Dr. I. Olier, Ms. T. Delgado, Ms. J. Martín and Mr. A. Pérez (all from GABRMN-UAB). GABRMN coordinator: Prof. C. Arús.

References

1. Cruz-Barbosa, R., Vellido, A.: Semi-supervised analysis of human brain tumours from partially labeled MRS information, using manifold learning models. International Journal of Neural Systems 21, 17–29 (2011)
2. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science 290, 2319–2323 (2000)
3. de Silva, V., Tenenbaum, J.: Global versus local methods in nonlinear dimensionality reduction. In: Becker, S., Thrun, S., Obermayer, K. (eds.) Advances in Neural Information Processing Systems, vol. 15. The MIT Press (2003)
4. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation 15(6), 1373–1396 (2003)
5. Roweis, S.T., Lawrence, K.S.: Nonlinear dimensionality reduction by locally linear embedding. Science (290), 2323–2326 (2000)
6. Zhu, X., Ghahramani, Z.: Learning from labeled and unlabeled data with label propagation. Technical report, CMU-CALD-02-107, Carnegie Mellon University (2002)
7. Belkin, M., Niyogi, P.: Using manifold structure for partially labelled classification. In: Advances in Neural Information Processing Systems (NIPS), vol. 15. MIT Press (2003)
8. Cruz-Barbosa, R., Vellido, A.: Semi-supervised geodesic generative topographic mapping. Pattern Recognition Letters 31(3), 202–209 (2010)
9. Bernstein, M., de Silva, V., Langford, J.C., Tenenbaum, J.B.: Graph approximations to geodesics on embedded manifolds. Technical report, Stanford University, CA, USA (2000)
10. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numerische Mathematik 1, 269–271 (1959)
11. Fredman, M.L., Tarjan, R.E.: Fibonacci heaps and their uses in improved network optimization algorithms. J. ACM 34(3), 596–615 (1987)
12. Asuncion, A., Newman, D.: UCI machine learning repository, University of California, Irvine, School of Information and Computer Sciences (2007), `http://www.ics.uci.edu/~mlearn/MLRepository.html`
13. Julià-Sapé, M., et al.: A multi-centre, web-accessible and quality control-checked database of *in vivo* MR spectra of brain tumour patients. Magn. Reson. Mater. Phys. MAGMA 19, 22–33 (2006)