Geometrical Methods in Machine Learning 2019

Evaluating Graph Kernels

Graph kernels [7] allow to define similarity between graphs using different types of heuristics.

You are asked to review, find, apply and compare different graph kernels computed from different perspectives i.e. random walk kernels [12], topological kernels [4], perform classification and explorative analysis of obtained embeddings of standard graph datasets using Kernel PCA.

Evaluating Intrinsic Dimension Estimation Methods

You are asked to review, find, apply and compare methods from [2] and to repeat exprerimemts on model and real data.

Sparse Covariance Matrix Estimation

Sample covariance matrix estimation is the key step to solution to the PCA problem, although MLE estimator may be suboptimal in high-dimensional setting. On the applied side covariance and its' normed variant correlation matrix are arising in fMRI data analysis, where the similarity of the time series of BOLD signal measured on different parts of the brain is represented exactly as their correlation.

You are asked to review, find, apply and compare different sparse and robust estimators of covariance matrix and in some cases its' inverse and to evaluate how sparse/robust covariance matrix estimation affects the performance of the classification control and deseased groups of people based on their fMRI scans.

Geometric Approach to Robust PCA

Tangent space estimation is an important subproblem of LTSA and GSE manifold learning algorithms. By default it is addressed with local PCA approach, however PCA is sensitive to outliers, which may reduce quality of the embedding when applied to real data.

You are asked to implement Robust PCA [8] which sees every point inducing linear subspace of the ambient space and to apply it to data analysis and to tangent space estimation problem.

Betti Numbers Computation via Hodge Laplacian

K-th Betti number β_k of a topological space equals the number of k-dimensional holes of that space, for example for a graph β_0 is the number of connected components and β_1 is the number of cycles.

To compute k-th Betti numbers of general topological space one need to find the kernel and image of ∂_k and ∂_{k+1} boundary operators respectively, which relate k-dimensional simplex to its $k-1$-dimensional faces (for example edge of a graph is mapped to the vertices it is connecting). Algorithmically it is done by a variant of Gaussian elimination in $O(n^3)$ time of number of simplices. Alternative approach is to compute the kernel of k-th Hodge Laplacian [11], which generalizes the notion of Laplacian operator on graphs to higher-order entities like simplicial complexes. One of the cons that it could be done via iterative optimization algorithms.

You are asked to estimate of the Betti numbers of model and real point cloud data via extending a neighborhood graph to a simplicial complex [14] and solving Hodge Laplacian equation [13] exactly and/or approximately.

Simplicial Data Structures

Simplicial complex can be seen as a generalization of a graph. Indeed, graph is a simplicial complex of dimension 1, constituted of 0-dimensional (vertices) and 1-dimensional simplices (edges).

Commonly used data structures to represent graphs are an adjacency matrix and an edge list. Both have pros and cons in terms of memory usage and algorithmic complexity of insertion and query operations. Data structures used to represent simplicial complexes such as Hasse diagram, simplicial tree [1], and various variants of adjacency lists [3] also have similar trade-offs.

You are asked to describe data structures used to represent simplicial complexes and to evaluate their memory usage and performance of inserting and querying operations on both complexes triangulating data manifolds, and clique complexes of fMRI graphs.

Deep Learning on Manifold of SPD Matrices for Neurodata

Space of $d \times d$ symmetric positive definite (SPD) matrices (Sym_+^d) is a Riemannian manifold, concretely an open convex cone:

$$Sym_+^d = \bigcap_{\mathbf{x} \in \mathcal{R}^n} \{\mathbf{A} \in Sym^d : \mathbf{x}^T \mathbf{A} \mathbf{x} > 0\}$$

Considering a dataset $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ examples of SPD matrices are covariance matrix $\mathbf{S} = n^{-1}\mathbf{X}\mathbf{X}^T$, Gram matrix $\mathbf{G} = \mathbf{X}^T\mathbf{X}$, matrix of pairwise distances $\mathbf{A} = (a_{ij})$, such that $a_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|_p^2$ where $\| \cdot \|_p$ is the p-norm.

As it is not a vector space, the notion Euclidean inner product and distance is not giving the true distance on this space. Instead the geodesic distance or distance between projection to tangent spaces should be considered. The notion of distance on manifolds, for example log-Frobenius metric $d_F(\mathbf{A}, \mathbf{B}) = \|\log(\mathbf{A}) - \log(\mathbf{B})\|_F$ and its' respective Gaussian kernel allow to apply metric and kernel machine learning methods on such data.

Such methods for respecting natural manifold metric generally perform better that "naive" approaches, which do not take the SPD space structure into account. On the other side being shallow methods they perform not as good as "naive" deep learning models.

Recent studies [6, 10, 9] try to generalize deep learning to manifold-valued data. The task of this project is to study, implement and reproduce results of deep learning models for manifold-valued data of SPD matrices for neurodata.

Hyperbolic Embeddings for Functional Brain Networks

Hyperbolic embeddings have shown as an promising tool to model hierarchical data. Functional brain networks are shown to follow (truncated) power law, so hyperbolic embeddings for them may be fruitful.

You are asked to implement hyperbolic embeddings for graphs representing functional brain networks [5] and to evaluate the performance of classification algorithms on it.

References

[1] J.-D. Boissonnat and C. Maria. The simplex tree: An efficient data structure for general simplicial complexes. *Algorithmica*, 70(3):406–427, 2014.

[2] F. Camastra and A. Staiano. Intrinsic dimension estimation: Advances and open problems. *Inf. Sci.*, 328:26–41, 2016.

[3] D. Canino, L. D. Floriani, and K. Weiss. Ia*: An adjacency-based representation for non-manifold simplicial shapes in arbitrary dimensions. *Computers Graphics*, 35:747–753, 2011.

[4] M. Carriere, M. Cuturi, and S. Oudot. Sliced wasserstein kernel for persistence diagrams. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 664–673. JMLR. org, 2017.

[5] B. P. Chamberlain, J. R. Clough, and M. P. Deisenroth. Hybed: Hyperbolic neural graph embedding, 2018.

[6] Z. Gao, Y. Wu, X. Bu, and Y. Jia. Learning a Robust Representation via a Deep Network on Symmetric Positive Definite Manifolds. pages 1–11, 2017.

[7] S. Ghosh, N. Das, T. Gonçalves, P. Quaresma, and M. Kundu. The journey of graph kernels through two decades. *Computer Science Review*, 27:88–111, 2018.

[8] S. Hauberg, A. Feragen, and M. J. Black. Grassmann averages for scalable robust pca. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3810–3817, 2014.

[9] Z. Huang and L. Van Gool. A Riemannian Network for SPD Matrix Learning. 2016.

[10] Z. Huang, J. Wu, and L. Van Gool. Building Deep Networks on Grassmann Manifolds. (1), 2016.

[11] L.-H. Lim. Hodge laplacians on graphs. *CoRR*, abs/1507.05379, 2015.

[12] N. Shervashidze, P. Schweitzer, E. J. v. Leeuwen, K. Mehlhorn, and K. M. Borgwardt. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12(Sep):2539–2561, 2011.

[13] N. K. Vishnoi et al. Lx= b. *Foundations and Trends® in Theoretical Computer Science*, 8(1–2):1–141, 2013.

[14] A. Zomorodian. Fast construction of the vietoris-rips complex. *Computers Graphics*, 34:263–271, 06 2010.