Feature Graph Learning for 3D Point Cloud Denoising

Wei Hu

2019.8.22

Outline

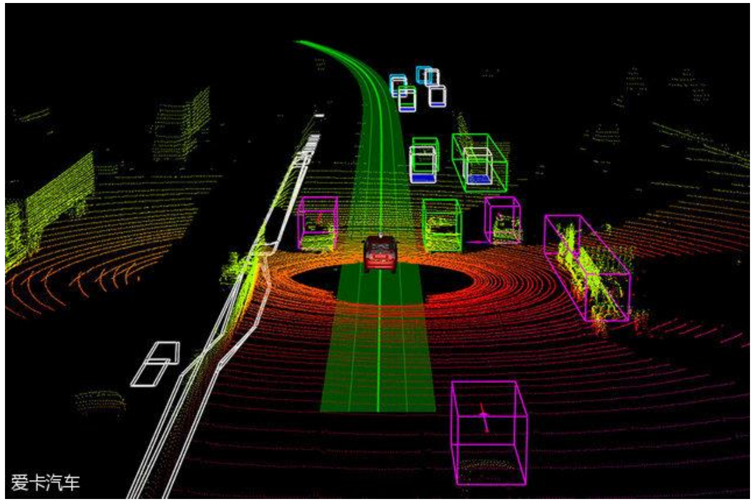
- Introduction to Graph Learning
- Related Concepts in Graph Spectral Theory
- Proposed Feature Metric Learning
- Application in 3D Point Cloud Denoising
- Conclusion

Introduction to Graph Learning

Graphs are flexible mathematical structures modeling pair-wise relations between data entities.

(a) Brain network

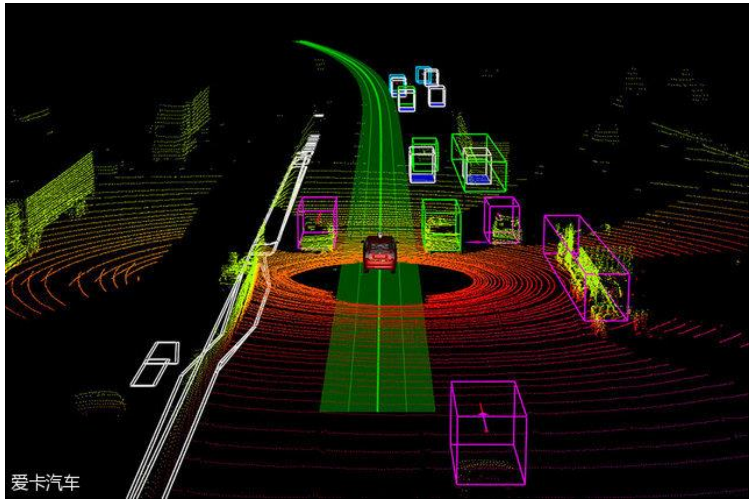
(b) Social network



(c) 3D point clouds

Introduction to Graph Learning

In many cases, the underlying graph is unknown or partially unknown.



Graph learning is critical!

Previous Works on Graph Learning

- Statistical methods
  - Graphical models: nodes represent r. v. , edges encode conditional dependencies
  - Learn the inverse covariance matrix (i.e., precision matrix)
  - Assumption: sparsity
  - Algorithms: Graphical lasso and its variants
  - Cons: assume many observations of a graphical model

J. Friedman, T. Hastie, and R. Tibshirani, “Sparse inverse covariance estimation with the graphical lasso,” *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.

Previous Works on Graph Learning

- Graph spectral methods
  - Enforce low frequency representation of observed signals & constraints for a valid graph Laplacian matrix
  - Assumption: smoothness prior of graph signals
  - Compared to statistical methods, assume fewer number of signal observations

➢ Our problem setting: the availability of a relevant feature vector per node.

X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, “Learning Laplacian matrix in smooth graph signal representations,” *IEEE Transactions on Signal Processing*, vol. 64, no. 23, pp. 6160–6173, 2016.

H. E. Egilmez, E. Pavez, and A. Ortega, “Graph learning from data under structural and Laplacian constraints,” *arXiv preprint arXiv:1611.05181*, 2016.

Related Concepts in Graph Spectral Theory

- An undirected graph: $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{W}\}$
- Edge weight $w_{i,j} \geq 0$: the degree of pairwise similarity
- Degree matrix $\mathbf{D}$: $d_{i,i} = \sum_{j=1}^{N} w_{i,j}$
- Combinatorial graph Laplacian: $\mathbf{L} := \mathbf{D} - \mathbf{W}$

Graph Laplacian Regularizer (GLR)

- Graph signal refers to data that resides on the nodes of a graph
- A graph signal $\mathbf{z} \in \mathbb{R}^N$ is smooth w.r.t. the underlying graph if

$$\mathbf{z}^\top \mathbf{L} \mathbf{z} = \sum_{i=1}^{N} \sum_{j=1}^{N} w_{i,j} (z_i - z_j)^2 < \epsilon$$

A function of features in our case

Proposed Feature Metric Learning

- Edge weight: $w_{i,j} = \exp\{-\delta_{i,j}\}$

  ↖ feature distance

- Mahalanobis distance:

$$\delta_{i,j} = (\mathbf{f}_i - \mathbf{f}_j)^\top \mathbf{M} (\mathbf{f}_i - \mathbf{f}_j)$$

capture correlation among features

↖ Positive definite

- Problem formulation

$$\min_{\mathbf{M}} \sum_{\{i,j\}} \exp\left\{-(\mathbf{f}_i - \mathbf{f}_j)^\top \mathbf{M} (\mathbf{f}_i - \mathbf{f}_j)\right\} d_{i,j}$$

$$\text{s.t.} \quad \mathbf{M} \succ 0.$$

↖ $d_{i,j} = (z_i - z_j)^2$

Proposed Feature Metric Learning

- Edge weight: $w_{i,j} = \exp\{-\delta_{i,j}\}$

  ↖ feature distance

- Mahalanobis distance:

$$\delta_{i,j} = (\mathbf{f}_i - \mathbf{f}_j)^\top \mathbf{M}(\mathbf{f}_i - \mathbf{f}_j)$$

↖ Positive definite

capture correlation among features

- Problem formulation

$$\min_{\mathbf{M}} \sum_{\{i,j\}} \exp\left\{-(\mathbf{f}_i - \mathbf{f}_j)^\top \mathbf{M}(\mathbf{f}_i - \mathbf{f}_j)\right\} d_{i,j}$$

$$\text{s.t.} \quad \mathbf{M} \succ 0; \quad \mathrm{tr}(\mathbf{M}) \leq C.$$

↖ $d_{i,j} = (z_i - z_j)^2$

SVD? No!

Optimization of Diagonal Entries

$$\min_{\mathbf{M}} \sum_{\{i,j\}} \exp\left\{-(\mathbf{f}_i-\mathbf{f}_j)^\top \mathbf{M}(\mathbf{f}_i-\mathbf{f}_j)\right\} d_{i,j}$$

$$\text{s.t.} \quad \mathbf{M} \succ 0; \quad \mathrm{tr}(\mathbf{M}) \leq C.$$

⇩

Off-diagonal entries

$$\min_{\{m_{i,i}\}} \sum_{\{i,j\}} \exp\left\{-\mathbf{g}_{i,j}^\top \left(\mathbf{M}' + \mathrm{diag}(\mathbf{M})\right) \mathbf{g}_{i,j}\right\} d_{i,j}$$

$$\mathbf{g}_{i,j} = \mathbf{f}_i - \mathbf{f}_j$$

$$\text{s.t.} \quad \mathbf{M} \succ 0; \quad \sum_i m_{i,i} \leq C.$$

- Gershgorin-based reformulation (mitigate full matrix eigen-decomposition)
- Proximal Gradient algorithm

Optimization of Off-diagonal Entries

$$\min_{\mathbf{M}} \sum_{\{i,j\}} \exp\left\{-(\mathbf{f}_i - \mathbf{f}_j)^\top \mathbf{M}(\mathbf{f}_i - \mathbf{f}_j)\right\} d_{i,j}$$

$$\text{s.t.} \quad \mathbf{M} \succ 0; \quad \mathrm{tr}(\mathbf{M}) \leq C.$$

⇩ Block Coordinate Iteration

$$\mathbf{M} = \begin{bmatrix} m_{1,1} & \mathbf{M}_{1,2} \\ \mathbf{M}_{2,1} & \mathbf{M}_{2,2} \end{bmatrix}$$

$$\min_{\mathbf{M}_{2,1}} \sum_{\{i,j\}} \exp\left\{-(\mathbf{f}_i - \mathbf{f}_j)^\top \mathbf{M}(\mathbf{f}_i - \mathbf{f}_j)\right\} d_{i,j}$$

$$\text{s.t.} \quad m_{1,1} - \mathbf{M}_{2,1}^\top \mathbf{M}_{2,2}^{-1} \mathbf{M}_{2,1} > 0,$$

← Haynsworth inertia additivity

$$m_{1,1} \leq C - \mathrm{tr}(\mathbf{M}_{2,2}).$$

- $\lambda_{\max}$ - bounded reformulation (mitigate large matrix inverse)
- Proximal Gradient algorithm

Application in 3D Point Cloud Denoising

- Patch-based formulation

Desired point cloud — Noisy point cloud — patches

$$\min_{\mathbf{Y},\mathbf{M}} \|\mathbf{Y}-\mathbf{P}\|_F^2 + \gamma \operatorname{tr}\left((\mathcal{T}\mathbf{Y}-\mathbf{C})^\top \mathbf{L}(\mathbf{M})(\mathcal{T}\mathbf{Y}-\mathbf{C})\right),$$

$$\text{s.t.} \quad \mathbf{M} \succ 0; \quad \operatorname{tr}(\mathbf{M}) \leq C.$$

- Features: 3D coordinates & surface normal

- Alternately optimize the point cloud and the Mahalanobis distance matrix

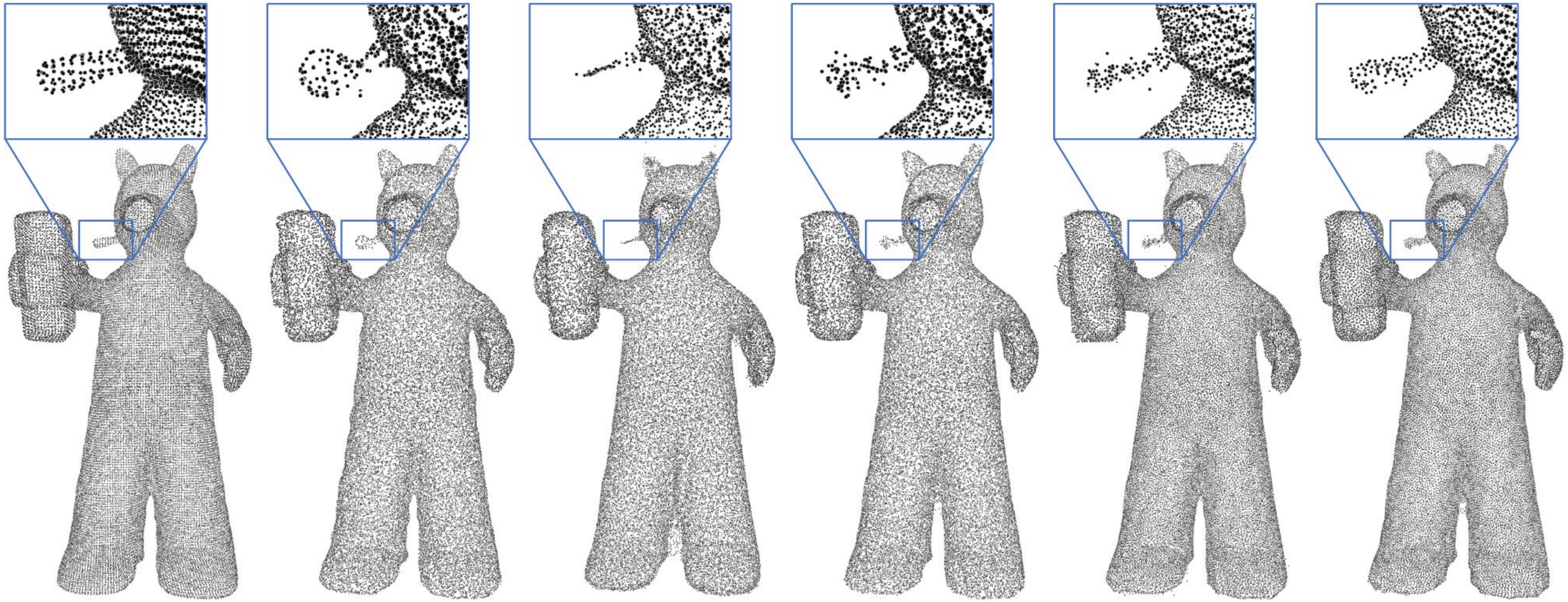
Application in 3D Point Cloud Denoising

TABLE I
MSE COMPARISON FOR DIFFERENT MODELS IN BENCHMARK WITH GAUSSIAN NOISE.

Model	Noisy	APSS	RIMLS	AWLOP	NLD	MRPCA	LR	GLR	Baseline	Ours
$\sigma = 0.02$										
Anchor	0.259	0.208	0.212	0.237	0.231	0.202	0.228	**0.189**	0.197	0.194
Daratech	0.245	0.203	0.209	0.228	0.222	0.225	0.213	0.197	0.196	**0.192**
DC	0.237	0.186	0.198	0.211	0.206	0.189	0.206	0.177	0.180	**0.177**
Gargoyle	0.257	0.208	0.217	0.230	0.230	0.215	0.240	0.202	0.204	**0.200**
Quasimoto	0.224	0.171	0.183	0.196	0.190	0.171	0.180	0.162	0.163	**0.161**
Average	0.244	0.195	0.203	0.220	0.215	0.200	0.213	0.185	0.188	**0.184**
$\sigma = 0.03$										
Anchor	0.322	0.239	0.244	0.259	0.265	0.230	0.246	**0.217**	0.227	0.221
Daratech	0.303	0.242	0.258	0.298	0.258	0.259	0.252	0.238	0.244	**0.236**
DC	0.293	0.210	0.226	0.257	0.235	0.211	0.221	0.203	0.207	**0.200**
Gargoyle	0.318	0.239	0.252	0.294	0.262	0.241	0.257	0.233	0.237	**0.228**
Quasimoto	0.274	0.188	0.203	0.226	0.217	0.187	0.193	0.176	0.181	**0.175**
Average	0.302	0.223	0.236	0.266	0.247	0.225	0.233	0.213	0.219	**0.212**
$\sigma = 0.04$										
Anchor	0.372	0.254	0.263	0.306	0.297	0.242	0.259	**0.228**	0.243	0.232
Daratech	0.348	0.282	0.308	0.286	0.295	0.288	0.283	0.276	0.286	**0.274**
DC	0.338	0.227	0.254	0.270	0.269	0.223	0.234	0.228	0.225	**0.215**
Gargoyle	0.368	0.262	0.277	0.297	0.294	0.257	0.269	0.257	0.259	**0.245**
Quasimoto	0.318	0.201	0.219	0.218	0.252	0.199	0.204	0.187	0.195	**0.182**
Average	0.348	0.245	0.264	0.275	0.281	0.241	0.249	0.235	0.242	**0.229**

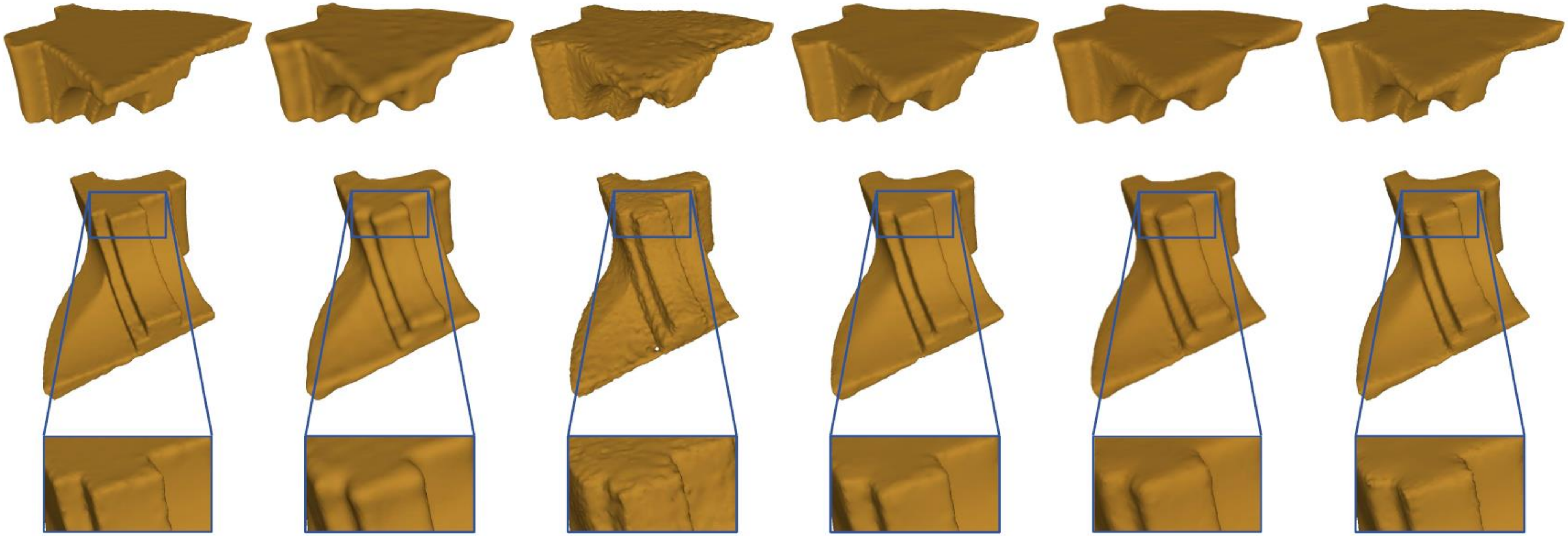
Baseline: randomly set edge weights in range [0,1]

Application in 3D Point Cloud Denoising



(a) Ground-truth (b) APSS (c) AWLOP (d) MRPCA (e) GLR (f) Ours

Application in 3D Point Cloud Denoising



(a) Ground-truth (b) APSS (c) AWLOP (d) MRPCA (e) GLR (f) Ours

Conclusion

- Propose feature graph learning given features of a single/partial signal observation

- Formulate as minimization of Graph Laplacian Regularizer with the Mahalanobis distance matrix $\mathbf{M}$ as variable

- Develop a fast algorithm to alternately optimize diagonal & off-diagonal entries of $\mathbf{M}$, while keeping $\mathbf{M}$ positive definite

- Validate the effectiveness by applying to 3D point cloud denoising

Thank you!

forhuwei@pku.edu.cn
http://www.icst.pku.edu.cn/huwei/

Reference to this talk: https://arxiv.org/abs/1907.09138