Feature representation and Laplacian embedding

Shen Zhengwei

July 13, 2020

Definition of feature selection

- In order to applying data mining and machine learning techniques to automatically discover knowledge from data of various sorts. We need feature selection, or dimensionality reduction.
- Feature selection(a.k.s. variable selection, factor analysis, attribute selection or variable subset selection)—selecting a subset of features from the high-dimensional data with many irrelevant features for a compact(low dimensional) and accurate(without noisy features) data representation.
  1. simplification of models to make them easier to interpret by researchers/users;
  2. reducing computational costs;
  3. to avoid the *curse of dimensionality*;
  4. preventing overfitting using less features;
  5. building better generalization models.

Definition of feature extraction

- Difference to Feature extraction———Feature extraction projects the original high-dimensional features to a new feature space with low dimensionality. The newly constructed feature space is usually a linear or nonlinear transformation of the original features. Feature selection, on the other hand, directly selects a subset of relevant features for model construction.
- Both feature extraction and feature selection have the advantages of improving learning performance, increasing computational efficiency, decreasing memory storage, and building better generalization models. Hence, they are both regarded as effective *dimensionality reduction techniques.*

Two types feature selection algorithms

- Supervised learning——usually select features by evaluating feature's correlation to labels(which class) of the training data. Generally, designed for classification or regression problems.
- Unsupervised learning for feature learning——-exploits data variance and separability(not the label information like SL) to evaluate feature relevance without labels, thus is more difficult than supervised learning methods. Generally designed for clustering problems

Dimensionality reduction

- Given a high dimensional data set $X = \{\mathbf{x}_1^T; \mathbf{x}_2^T; \dots; \mathbf{x}_n^T\} \in R^{n \times d}$ where the row vector $\mathbf{x}_i^T \in R^d$ is a instance with d-dimensional patterns. How to compute n corresponding output k-dimensional($k << d$) patterns(features) $f_j^\top \in R^k$ that provides a faithful low dimensional representation to X? By faithful, nearby inputs are mapped to nearby outputs, while faraway inputs are mapped to faraway outputs.
- The spectral methods for dimensionality reduction, where the low dimensional representations are derived from the top spectral methods or bottom eigenvectors of specially constructed matrices, e.g. covariance matrix $C = n^{-1}XX^\top$ for PCA, Gram matrix $G = X^\top X$ for MDS, Laplacian matrix in this part, including linear DR(PCA, MDS) and nonlinear DR(This part)

Dimensionality reduction-cont'd

- PCA the projected low-dimensional subspace have maximum variance, and metric multidimensional scaling(MDS) preserve the inner product.
- However, when the input patterns lie on or near a low dimensional submanifold of the input space. In this case, the structure of the data set may be highly nonlinear, and linear methods are bound to fail.
- Graph-based methods have recently emerged as a powerful tool for analyzing high dimensional data that has been sampled from a low dimensional submanifold

Some basic concepts of graph

- The points(data) in Graph does not like the points in Euclidean space R^d having fixed neighborhood and distance, thus have to represents graph using some matrices: adjacency and graph Laplacian and its variants.

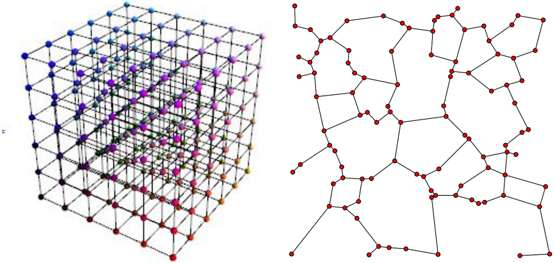
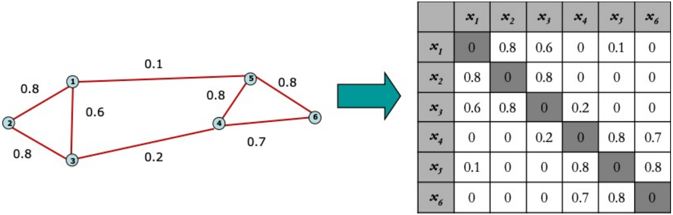


Figure: Grid and Graph

- Thus the properties of graphs is studied(*spectral graph theory*) via the eigenvalues and eigenvectors of these associated graph matrices: *Adjacency and Laplacian matrix.*

Adjecency matrix

- By taking n data instances $\{\mathbf{x}_1^T; \mathbf{x}_2^T; \ldots; \mathbf{x}_n^T\}$ as vertex set of an *weighted undirected graph* $G = (\mathcal{V}, \mathcal{E})$, and denote its adjacency matrix $W = (w_{ij}) \in R^{n \times n} > 0$ if there is an edge between the ith and jth vertices.



	x_1	x_2	x_3	x_4	x_5	x_6
x_1	0	0.8	0.6	0	0.1	0
x_2	0.8	0	0.8	0	0	0
x_3	0.6	0.8	0	0.2	0	0
x_4	0	0	0.2	0	0.8	0.7
x_5	0.1	0	0	0.8	0	0.8
x_6	0	0	0	0.7	0.8	0

Figure: weighted undirected graph and adjacency matrix

Adjacency matrix cont'd

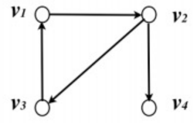
1. Adjacency matrix W is symmetric;
2. Eigenvalues are real;
3. The corresponding eigenvectors could span *orthonormal basis* to R^n space;
4. W can be viewed as a linear map(*diffusion operator/weighted average*), for any vector $f = \{f_1, f_2, ...f_n\}^T$,

$$(Wf)_i := \sum_{i \sim j} w_{ij} f_j$$

that is, the value of Wf at f_i is the weighted sum of the values of vectors f at the jth nodes adjacent to i th node.

Laplacian matrix(operator)

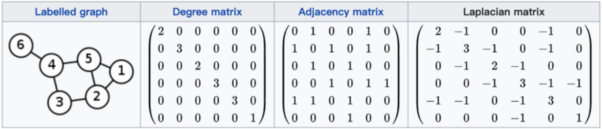
- How to characterize the *difference* between vertex and its connected neighborhood vertices in an undirected graph G?
- Denote $(\nabla \boldsymbol{f})(e_{ij}) = f(v_j) - f(v_i)$ if v_i and v_j are connected via edge e_{ij}.
- $$\begin{pmatrix} f(v2)-f(v1) \\ f(v1)-f(v3) \\ f(v3)-f(v2) \\ f(v4)-f(v2) \end{pmatrix} = \underbrace{\begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & -1 & 0 & +1 \end{bmatrix}}_{\nabla \text{ incidence matrix}} \begin{pmatrix} f(v1) \\ f(v2) \\ f(v3) \\ f(v4) \end{pmatrix}$$



- Denote $L = \nabla^{\top}\nabla$ by the Laplacian operator.
- Then $(Lf)(x_i) = \sum_{x_i \sim x_j} (f(x_i) - f(x_j))$ is used to characterize the sum of difference between x_i and its connected neighborhood vertices x_j.

Laplacian matrix(operator) cont'd

- Degree matrix $D = W\mathbf{1}_n = diag(d_i)$, $d_i = \sum_{k=1}^{n} w_{ik}$, where d_i can be interpreted as an estimation of the density around $\mathbf{x}_i$, since the more data points that are close to $\mathbf{x}_i$, larger the d_i.
- Laplacian matrix $L = D - W$ and the normalized Laplacian matrix $\mathcal{L} = D^{-1/2}LD^{-1/2} = I - D^{-1/2}WD^{-1/2}$.



Labelled graph	Degree matrix	Adjacency matrix	Laplacian matrix
	$\begin{pmatrix} 2&0&0&0&0&0\\ 0&3&0&0&0&0\\ 0&0&2&0&0&0\\ 0&0&0&3&0&0\\ 0&0&0&0&3&0\\ 0&0&0&0&0&1 \end{pmatrix}$	$\begin{pmatrix} 0&1&0&0&1&0\\ 1&0&1&0&1&0\\ 0&1&0&1&0&0\\ 0&0&1&0&1&1\\ 1&1&0&1&0&0\\ 0&0&0&1&0&0 \end{pmatrix}$	$\begin{pmatrix} 2&-1&0&0&-1&0\\ -1&3&-1&0&-1&0\\ 0&-1&2&-1&0&0\\ 0&0&-1&3&-1&-1\\ -1&-1&0&-1&3&0\\ 0&0&0&-1&0&1 \end{pmatrix}$

Figure: Undirected graph and adjacency matrix

Some properties of—Laplacian matrix(operator)

1. L is symmetric, thus having *real* eigenvalues and eigenvectors;
2. The undirected weighted graph Laplacian can be viewed as a linear map, for all vector $f \in R^n$, we have

$$Lf = \sum_{i \sim j} w_{ij}(f_i - f_j)$$

3. Laplacian defines natural *quadratic form* of graphs(aka, the *Dirichlet sum* of graph)

$$f^T Lf = \frac{1}{2} \sum_{i \sim j} w_{ij}(f_i - f_j)^2 \tag{1}$$

which indicates L is positive-semidefinite, thus, $0 = \lambda_1 \leq \lambda_2 \leq ..... \leq \lambda_n$(spectrum of the Laplacian).

Proof to (??):

$$
\begin{aligned}
f^T L f = f^T(D-W)f &= f^T D f - f^T W f \\
&= \sum_{i=1}^{n} d_i f_i^2 - \sum_{i,j=1}^{n} w_{i,j} f_i f_j \\
&= \frac{1}{2}\left(\sum_{i=1}^{n} d_i f_i^2 + \sum_{j=1}^{n} d_j f_j^2 - 2\sum_{i,j=1}^{n} w_{i,j} f_i f_j \right) \\
&= \frac{1}{2}\left(\sum_{i=1}^{n} \left(\sum_{j=1}^{n} w_{ij} \right) f_i^2 + \sum_{j=1}^{n} \left(\sum_{i=1}^{n} w_{ij} \right) f_j^2 - 2\sum_{i,j=1}^{n} w_{i,j} f_i f_j \right)
\end{aligned}
$$

Some properties of—Laplacian matrix(operator) cont'd

Connected graph Laplacians

1. $L\mathbf{1}_n = 0$, where $\mathbf{1}_n = \{1, 1, ..., 1\}^T$, which indicates the graph is connected; and $\mathbf{1}_n$ is the eigenvector of the the smallest eigenvalue 0.

$$L\mathbf{1}_n = \begin{bmatrix} d_1 - \sum_j w_{1j} \\ d_2 - \sum_j w_{2j} \\ \dots \\ d_n - \sum_j w_{nj} \end{bmatrix} = 0$$

2. If any two vertices are connected by a path, then $\boldsymbol{u} = (u(1), \dots, u(n))$ needs to be constant at all vertices such that the quadratic form vanishes, i.e. satisfying

$$0 = \boldsymbol{u}^\top L\boldsymbol{u} = \sum_{i,j=1}^{n} w_{ij}(u(i) - u(j))^2.$$

Graph with one connected component has the constant vector $\boldsymbol{u}_1 = \mathbf{1}_n$ as the only eigenvector to eigenvalue 0.

Some properties of—Laplacian matrix(operator) cont'd

Connected graph Laplacians

1. Let $u_1 = \mathbf{1}_n$ be the eigenvector of the the smallest eigenvalue 0. u_2 be the eigenvector is the first nonzero eigenvalue, then,

$$u_2^T u_1 = u_2^T \mathbf{1}_n = 0$$

2. In fact, for any eigenvector $u_i, 2 \leq i \leq n$,

$$u_i^T \mathbf{1}_n = 0$$

   That is

$$\sum_{j=1}^{n} u_i \left(v_j\right) = 0$$

   Each component is bounded by:

$$-1 < u_i \left(v_j\right) < 1$$

Some properties of—Laplacian matrix(operator) cont'd

A graph with $k > 1$ connected components

1. Each connected component has an associated Laplacian. Therefore, we can write matrix L as a block diagonal matrix:

$$L = \begin{bmatrix} L_1 & & \\ & \ddots & \\ & & L_k \end{bmatrix}$$

2. The spectrum of L is given by the union of the spectra of L_i;
3. Each block corresponds to a connected component, hence each matrix L_i has an eigenvalue 0 with multiplicity 1;
4. Thus the eigenvalue $\lambda_1 = 0$ of L has multiplicity k.
5. In general, the multiplicity of 0 as a Laplacian eigenvalue is the number of connected components of a graph.

Some properties of—Laplacian matrix(operator) cont'd

A graph with $k > 1$ connected components

1. The eigenspace corresponding to $\lambda_1 = \ldots = \lambda_k = 0$ is spanned by the k mutually orthogonal vectors:

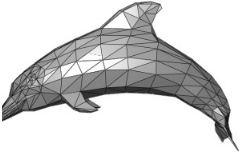
$$\begin{aligned} \boldsymbol{u}_1 &= \mathbf{1}_{L_1} \\ &\ldots \\ \boldsymbol{u}_k &= \mathbf{1}_{L_k} \end{aligned}$$

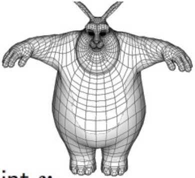
2. $\mathbf{1}_{L_i} = (0000111110000)^\top \in \mathbb{R}^n$
3. These vectors are the indicator vectors of the graph's connected components.
4. Notice that $\mathbf{1}_{L_1} + \ldots + \mathbf{1}_{L_k} = \mathbf{1}_n$

Applications of Laplacian operator

- Laplacian allows a natural link between discrete representation, such as graphs; and continuous representations, such as *manifolds* and vector spaces. Some applications of Laplacian
  1. Graph partioning problem(spectral clustering);
  2. Graph matching(spectral matching)
  3. Image segmentation(Spectral partitioning)
  4. Document classification based on semantic association of words, collaborative recommendation;
  5. Feature selection(Laplacian embedding);
  6. Manifold analysis: Manifold embedding, manifold learning, mesh segmentation;

Example of Laplacians—Discrete surface Laplacians(3D Meshes)





- A graph vertex v_i is associated with a 3D point $\boldsymbol{v}_i$.
- The weight of an edge e_{ij} is defined by the Gaussian kernel:

$$w_{ij} = \exp\left(-\|\boldsymbol{v}_i - \boldsymbol{v}_j\|^2/\sigma^2\right)$$

- $0 \leq w_{\min} \leq w_{ij} \leq w_{\max} \leq 1$
- Hence, the geometric structure of the mesh is encoded in the weights.
- Other weighting functions were proposed in the literature.

Extremal eigenvalues of symmetric matrices

- The eigenvalues and eigenvectors of symmetric matrices have many characterizations. The ones that will be most useful to us will come from optimization problems. In particular, they arise when maximizing or minimizing the Rayleigh quotient with respect to a matrix M.
- Rayleigh quotient of a vector $\mathbf{x} \neq 0$ with respect to a symmetric matrix M is the ratio

$$R(\mathbf{x}) := \frac{\boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}}$$

- Rayleigh quotient does not change multiplying $\mathbf{x}$ by a nonzero constant. So, it suffices to consider vectors $\mathbf{x}$ of unit norm i.e. $\mathbf{x}^T\mathbf{x} = 1$, which is a compact set.
- Taking $\mathbf{x} = \mathbf{u}$ be an eigenvector of M of eigenvalue λ, then

$$\frac{\boldsymbol{u}^T \boldsymbol{M} \boldsymbol{u}}{\boldsymbol{u}^T \boldsymbol{u}} = \frac{\boldsymbol{u}^T \lambda \boldsymbol{u}}{\boldsymbol{u}^T \boldsymbol{u}} = \frac{\lambda \boldsymbol{u}^T \boldsymbol{u}}{\boldsymbol{u}^T \boldsymbol{u}} = \lambda$$

Extremal eigenvalues of symmetric matrices cont'd

- Then
$$\lambda_{\text{min}} = \min_{\boldsymbol{x} \neq 0} R(\boldsymbol{x}), \quad \lambda_{\text{max}} = \max_{\boldsymbol{x} \neq 0} R(\boldsymbol{x})$$
and the extremal values are attained precisely on the corresponding eigenvectors.
- **Proof:** Let $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots \boldsymbol{u}_n$ be an orthonormal basis of eigenvectors corresponding to eigenvalues $\lambda_{\text{min}} = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n = \lambda_{\text{max}}$. Then we can write any vector $\boldsymbol{x}$ in this basis as $\boldsymbol{x} = \sum \left(\boldsymbol{u}_i^T \boldsymbol{x}\right) \boldsymbol{u}_i = \sum c_i \boldsymbol{u}_i$, and $M\boldsymbol{x} = \sum \left(\boldsymbol{u}_i^T \boldsymbol{x}\right) \lambda_i \boldsymbol{u}_i = \sum c_i \lambda_i \boldsymbol{u}_i$. Then,
$$R(\mathbf{x}) = \frac{\boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}} = \frac{\sum \lambda_i |c_i|^2}{\sum |c_i|^2}$$
Suppose $\mathbf{x}$ to be unit norm, then $\sum |c_i|^2 = 1$ indicates that
$$\lambda_{min} = \lambda_{min} \sum |c_i|^2 \leq R(\mathbf{x}) = \sum \lambda_i |c_i|^2 \leq \lambda_{max} \sum |c_i|^2 \leq \lambda_{max}$$
Hence $\mathbf{u}_{min} = arg \min_{\boldsymbol{x} \neq 0} R(\boldsymbol{x})$ and $\mathbf{u}_{max} = arg \max_{\boldsymbol{x} \neq 0} R(\boldsymbol{x})$

Extremal eigenvalues of symmetric matrices cont'd

- In fact, we have the following generalized results

$$\lambda_i = \min_{x \perp \mathbf{u}_1, \ldots, \mathbf{u}_{i-1}} \frac{x^T \mathbf{M} x}{x^T x}$$

  and

$$\mathbf{u}_i = arg \min_{x \perp \mathbf{u}_1, \ldots, \mathbf{u}_{i-1}} \frac{x^T \mathbf{M} x}{x^T x}$$

Eigenvalues of Laplacian matrix L

- Suppose the eigenvalues of L to be $\lambda_{\min} = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n = \lambda_{\max}$, here n denotes the number of vertices of the graph(the number of data). For the purposes of relating eigenvalues to *graph invariants*, the most important eigenvalues will turn out to be the extremal ones, as well as λ_2.
- If graph is connected, eigenvalue $\lambda_{\min} = \lambda_1 = 0$ of L has multiplicity 1 and constant vector $\mathbf{1}_n$ is the eigenvectors. This constant vector $\mathbf{u} = \mathbf{1}_n$ make the quadratic form of Laplacian L vanishing, i.e. $0 = \boldsymbol{u}^\top L\boldsymbol{u} = \sum_{i,j=1}^n w_{ij}(u(i) - u(j))^2$.
- In other words,

$$0 = \min_{\boldsymbol{f} \in R^n} \boldsymbol{f}^\top L\boldsymbol{f}$$

and

$$\mathbf{1}_n = arg \min_{\boldsymbol{f} \in R^n} \boldsymbol{f}^\top L\boldsymbol{f}$$

are constant at each vertex $\mathbf{u}(v_i) = 1$.

Laplacian embedding

- Remember the data set $X = \left\{\mathbf{x}_1^T; \mathbf{x}_2^T; \dots; \mathbf{x}_n^T\right\}$ represented by a graph G, each vertex has a data $\mathbf{x}_i \in R^d$.
- Laplacian embedding is to find an encoding map $\mathbf{f} : R^d \to R^k, (k < d)$ to encodes each vertex $\mathbf{x}_i$ into a general low-dimensional space vector $\mathbf{x_i} \to f_i(\mathbf{x}_i) \in R^k$ to preserve the data structure information(manifold structure or semantic information) accurately. Then $\mathbf{f}(X) = \{f_1(\mathbf{x}_1)^T; f_2(\mathbf{x}_2)^T; ....; f_n(\mathbf{x}_n)^T\} \in R^{n\times k}$
- As we know, it the graph is connected, $\mathbf{1}_n$ is the only eigenvector for $\lambda_1 = 0$. To eliminate this trivial solution which collapses all vertices of G onto the constant number 1, we usually put an constraints of orthogonality to the solution $\mathbf{f}$, that is

$$\mathbf{f}^* = arg \min_{\boldsymbol{f} \in R^n, \ \mathbf{f}^T \mathbf{1}_n = 0} \boldsymbol{f}^\top L \boldsymbol{f} \tag{2}$$

Laplacian embedding—1-D embedding

- $\mathbf{f}(X) = \{f_1(\mathbf{x}_1); f_2(\mathbf{x}_2); \ldots.; f_n(\mathbf{x}_n)\} \in R^{n\times 1}$
- Map a weighted graph onto a line(only one vector) such that connected(closed) nodes stay as close as possible, i.e., minimize

  $$\sum_{i,j=1}^{n} w_{ij}\left(f_i\left(\mathbf{x}_i\right) - f_i\left(\mathbf{x}_j\right)\right)^2$$

  under appropriate constraints, w_{ij} incurs a heavy penalty if neighboring point $\mathbf{x}_i, \mathbf{x}_j$ are mapped far apart. Thus,

  $$\arg\min_{\boldsymbol{f}} \boldsymbol{f}^\top L\boldsymbol{f} \text{ subject to } \boldsymbol{f}^\top\boldsymbol{f} = 1,\ \boldsymbol{f}^\top\mathbf{1}_n = 0 \qquad (3)$$

  where $\boldsymbol{f}^\top\boldsymbol{f} = 1$ is the unit norm constraint, $\boldsymbol{f}^\top\mathbf{1}_n = 0$ avoiding trivial solution. *Thus, solving* (**??**) *equals to project vertex information of graph onto the unit sphere.*
- $\Longleftrightarrow$

  $$\arg\min_{\boldsymbol{f}^\top\mathbf{1}_n=0} \frac{\boldsymbol{f}^\top L\boldsymbol{f}}{\boldsymbol{f}^\top\boldsymbol{f}} \qquad (4)$$

Laplacian embedding—1-D embedding cont'd

- Solution to (**??**) or (**??**) is the eigenvector $\mathbf{u}_2$ associated with the smallest nonzero eigenvalues λ_2 of L.
- **Algorithm:**
  1. **Step1**: Eigen-decomposition to $L = \mathbf{U}\Lambda\mathbf{U}^\top$;
  2. **Step2**: $\frac{\boldsymbol{f}^\top L\boldsymbol{f}}{\boldsymbol{f}^\top\boldsymbol{f}} = \frac{\boldsymbol{f}^\top\mathbf{U}\Lambda\mathbf{U}^\top\boldsymbol{f}}{\boldsymbol{f}^\top\boldsymbol{f}}$
  3. **Step3**: Taking $\mathbf{z} = \mathbf{U}^\top\boldsymbol{f} = [\mathbf{z}_1, \mathbf{z}_2, \ldots\mathbf{z}_n]^\top]$, then $\frac{\boldsymbol{f}^\top L\boldsymbol{f}}{\boldsymbol{f}^\top\boldsymbol{f}} = \frac{\boldsymbol{f}^\top\mathbf{U}\Lambda\mathbf{U}^\top\boldsymbol{f}}{\boldsymbol{f}^\top\boldsymbol{f}} = \frac{[\mathbf{z}_1,\mathbf{z}_2,\ldots\mathbf{z}_n]diag\lambda_i[\mathbf{z}_1,\mathbf{z}_2,\ldots\mathbf{z}_n]^\top}{\sum_{i=1}^n |z_i|^2} = \frac{\sum_{i=1}^n \lambda_i|\mathbf{z}_i|^2}{\sum_{i=1}^n |\mathbf{z}_i|^2}$
  4. **Step4**: Note that the first row of $\mathbf{U}^\top$ is constant vector $\mathbf{1}_n$ and by the constraint $\boldsymbol{f}^\top\mathbf{1}_n = 0$, then $\mathbf{z} = \mathbf{U}^\top\boldsymbol{f} = [0, \mathbf{z}_2, \ldots\mathbf{z}_n]^\top$.
  5. **Step5**: $\frac{\boldsymbol{f}^\top L\boldsymbol{f}}{\boldsymbol{f}^\top\boldsymbol{f}} = \frac{\sum_{i=1}^n \lambda_i|\mathbf{z}_i|^2}{\sum_{i=1}^n |\mathbf{z}_i|^2} \geq \frac{\lambda_2\sum_{i=2}^n |\mathbf{z}_i|^2}{\sum_{i=2}^n |\mathbf{z}_i|^2} = \lambda_2$.

Laplacian embedding—1-D embedding cont'd

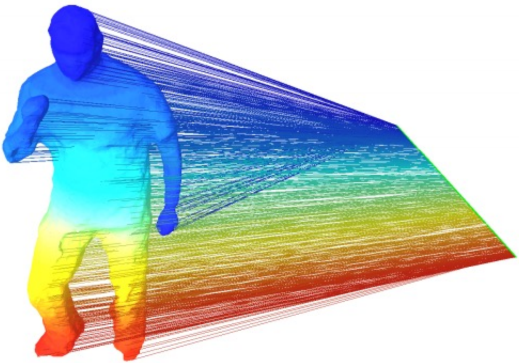


Figure: Example of mapping a graph on the 1D Fiedler vector(i.e. the eigenvector of the first nonzero eigenvalue)

Laplacian embedding—Higher-D embedding cont'd

- Consider more general situation—embedding the graph in a k-dimensional Euclidean space

$$\mathbf{f}(X) = \{f_1(\mathbf{x}_1)^T; f_2(\mathbf{x}_2)^T; ....; f_n(\mathbf{x}_n)^T\} \in R^{n \times k}$$

  where $\mathbf{f}_i(\mathbf{x}_i) \in R^k$ is the $k-$dimensional representation of the ith vertex. Then the embedding is given by the $n \times k$ matrix $\mathbf{f} = \{\mathbf{f}^{(1)}, \mathbf{f}^{(2)}, ..., \mathbf{f}^{(k)}\}$ where the ith row provides the embedding coordinates of the ith vertex.

- we need to minimize

$$\{\mathbf{f}^{(1)}, ..\mathbf{f}^{(k)}\} = arg \min_{\mathbf{f}^\top \mathbf{f} = \mathbf{I}_k, \mathbf{f}^\top \mathbf{1}_n = 0} tr(\mathbf{f}^\top L \mathbf{f}) = \sum_{i,j=1}^{n} w_{ij} \|\mathbf{f}_i\,(\mathbf{x}_i) - \mathbf{f}_i\,(\mathbf{x}_j)\,\|^2$$

- Like the 1-d case, the solution is provided by the eigenvectors corresponding to the k lowest nonzero eigenvalues of the L.

Laplacian embedding—Higher-D embedding cont'd

Algorithm:

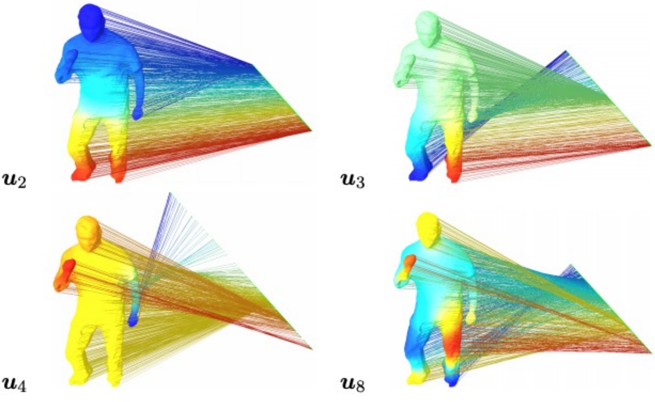
1. **Step1**: Eigen-decomposition to $L = \mathbf{U}\Lambda\mathbf{U}^\top$;
2. **Step2**: Select the k smallest non-zero eigenvalues $\lambda_2 \leq \ldots \leq \lambda_{k+1}$
3. **Step3**: We obtain the $n \times k$ matrix $\overline{\mathbf{U}} = [\boldsymbol{u}_2 \ldots \boldsymbol{u}_{k+1}]$

$$\overline{\mathbf{U}} = \underbrace{\begin{bmatrix} \boldsymbol{u}_2(v_1) & \ldots & \boldsymbol{u}_{k+1}(v_1) \\ \vdots & & \vdots \\ \boldsymbol{u}_2(v_n) & \ldots & \boldsymbol{u}_{k+1}(v_n) \end{bmatrix}}_{\mathbf{f}^{(1)},\ldots\ldots\ldots\ldots\ldots\mathbf{f}^{(k)}}$$

$\boldsymbol{u}_i^\top \boldsymbol{u}_j = \delta_{ij}$ (orthonormal vectors), hence $\overline{\mathbf{U}}^\top \overline{\mathbf{U}} = \mathbf{I}_k$.

4. **Step4**: Column $i(2 \leq i \leq k+1)$ of this matrix is the eigenvector $\boldsymbol{u}_i$ corresponding to the eigenvalues λ_i. i.e. the solution $\{\mathbf{f}^{(1)}, ..\mathbf{f}^{(k)}\}$. And, each column $(\boldsymbol{u}_i(v_1), \boldsymbol{u}_i(v_{2,}), ...\boldsymbol{u}_i(v_n))^\top$ also is the mapping of each vertex to eigenvectors $\boldsymbol{u}_i$. The ith row is the encoding representation to the ith vertex of graph.

Laplacian embedding—Higher-D embedding cont'd



Laplacian embedding—from the Graph PCA viewpoint

1. Note that
$$\mathbf{X} = \mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{U}^\top = (\mathbf{x}_1, ..\mathbf{x}_j, ...\mathbf{x}_n)$$
where $\mathbf{U} = \{\mathbf{u}_2, ...\mathbf{u}_{k+1}\}$ and $\sum_{j=1}^{n} \boldsymbol{u}_i(v_j) = 0$.
2. The covariance matrix of data set **X**:
$$\mathbf{S} = \frac{1}{n}\sum_{j=1}^{n} \boldsymbol{x}_j \boldsymbol{x}_j^\top = \frac{1}{n}\mathbf{X}\mathbf{X}^\top = \frac{1}{n}\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{U}^\top\mathbf{U}\mathbf{\Lambda}^{-\frac{1}{2}} = \frac{1}{n}\mathbf{\Lambda}^{-1}$$
3. The vectors $\boldsymbol{u}_2, \ldots, \boldsymbol{u}_{k+1}$ are the directions of maximum variance of the graph embedding, with $\lambda_2^{-1} \geq \ldots \geq \lambda_{k+1}^{-1}$

Other Laplacian matrix

- The combinatorial (unnormalized) Laplacian: $\mathbf{L} = \mathbf{D} - \mathbf{W}$;
- The normalized graph Laplacian (symmetric and semi-definite positive):

$$\mathbf{L}_n = \mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}$$

- The random-walk graph Laplacian:

$$\mathbf{L}_r = \mathbf{D}^{-1}\mathbf{L} = \mathbf{I} - \mathbf{W}_t$$

  where $\mathbf{W}_t = \mathbf{D}^{-1}\mathbf{W}$ is the *transition matrix* allows an analogy with Markov chains.

- These matrices are similar:

$$\mathbf{L}_r = \mathbf{D}^{-\frac{1}{2}}\mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}}\mathbf{D}^{\frac{1}{2}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{L}_n\mathbf{D}^{\frac{1}{2}}$$

Other Laplacian matrix-cont'd

- Some authors use the following matrix:

$$\mathbf{L}_a = \frac{1}{d_{\max}} \left(\mathbf{W} + d_{\max} \mathbf{I} - \mathbf{D} \right)$$

- This matrix is closely related to **L**

$$\mathbf{L}_a = \frac{1}{d_{\max}} \left(d_{\max} \mathbf{I} - \mathbf{L} \right)$$

- and we have:

$$\mathbf{L}_a \boldsymbol{u} = \mu \boldsymbol{u} \Longleftrightarrow \mathbf{L} \boldsymbol{u} = \lambda \boldsymbol{u}, \mu = 1 - \frac{\lambda}{d_{\max}}$$

Eigenvalues and eigenvectors of $\mathbf{L}_n$ and $\mathbf{L}_r$

- $\mathbf{L}_r \boldsymbol{m} = \lambda \boldsymbol{m} \Longleftrightarrow \mathbf{L}\boldsymbol{m} = \lambda \mathbf{D}\boldsymbol{m}$ hence, $\mathbf{L}_r : \lambda_1 = 0; \boldsymbol{m}_1 = \mathbf{1}$
- $\mathbf{L}_n \boldsymbol{v} = \lambda \boldsymbol{v}$. By virtue of the similarity transformation between the two matrices:

$$\mathbf{L}_n : \quad \lambda_1 = 0 \quad \boldsymbol{v}_1 = \mathbf{D}^{\frac{1}{2}}\mathbf{1}$$

- More generally, the two matrices have the same eigenvalues:

$$0 = \lambda_1 \leq \ldots \leq \lambda_i \ldots \leq \lambda_n$$

- Their eigenvectors are related by:

$$\boldsymbol{v}_i = \mathbf{D}^{\frac{1}{2}}\boldsymbol{m}_i, \forall i = 1 \ldots n$$

Spectral embedding using the random-walk Laplacian $\mathbf{L}_r$

- The $n \times k$ matrix contains the first k eigenvectors of $\mathbf{L}_r$

$$\mathbf{M} = \left[\begin{array}{ccc} \boldsymbol{m}_2 & \dots & \boldsymbol{m}_{k+1} \end{array} \right]$$

- It is straightforward to obtain the following expressions, where $\boldsymbol{d}$ and $\mathbf{D}$ are the degree-vector and the degree-matrix:

$$\boldsymbol{m}_i^\top \boldsymbol{d} = 0, \forall i, 2 \leq i \leq n$$
$$\mathbf{M}^\top \mathbf{D} \mathbf{M} = \mathbf{I}_k$$

- The isometric Laplacian embedding using the random-walk Laplacian:

$$\mathbf{Y} = \mathbf{M}^\top = \left[\begin{array}{ccc} \boldsymbol{y}_1 & \dots & \boldsymbol{y}_n \end{array} \right]$$

Other applications of graph Laplacian(1) -Graph partitioning problem

- The *graph-cut problem*: Partition the graph such that:
  1. Edges between groups have very low(minimized) weight("flow"), and;
  2. Edges within a group have high weight.

$$\operatorname{cut}(C_1, \ldots, C_k) := \frac{1}{2} \sum_{i,j=1}^{k} W\left(C_i, \bar{C}_j\right), \text{where } W(A, B) = \sum_{i \in A, j \in B} w_{ij}$$

- *Ratio cut:* (Hagen & Kahng 1992)

$$\text{Ratio Cut}(C_1, \ldots, C_k) := \frac{1}{2} \sum_{i=1}^{k} \frac{W\left(C_i, \bar{C}_i\right)}{|C_i|}$$

- *Normalized cut:* (Shi & Malik 2000)-minimize the normalize sum of weights in the resulting graphs

$$\mathrm{NCut}(C_1, \ldots, C_k) := \frac{1}{2} \sum_{i=1}^{k} \frac{W\left(C_i, \bar{C}_i\right)}{\operatorname{vol}(C_i)}$$

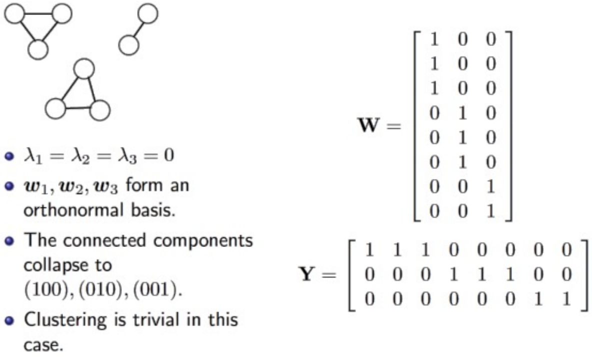
Other applications of graph Laplacian(2) -Spectral clustering

- Both ratio-cut and normalized-cut minimizations are NP-hard problems
- Spectral clustering is a way to solve relaxed versions of these problems:
  1. The smallest non-zero eigenvectors of the unnormalized Laplacian approximate the RatioCut minimization criterion,
  2. The smallest non-zero eigenvectors of the random-walk Laplacian approximate the NCut criterion.

Other applications of graph Laplacian(2) -spectral clustering using the random-walk Laplacian

- For details see (von Luxburg '07)
- **Input:** Laplacian $\mathbf{L}_r$ and the number k of clusters to compute.
- **Output:** Cluster $C_1, \ldots, C_k$
  1. Compute $\mathbf{M}$ formed with the first k eigenvectors of the random-walk Laplacian;
  2. Determine the spectral embedding $\mathbf{Y} = \mathbf{M}^\top$;
  3. Cluster the columns $\boldsymbol{y}_j, j = 1, \ldots, n$ into k clusters using the K-means algorithm.

Other applications of graph Laplacian(2) -spectral clustering



- $\lambda_1 = \lambda_2 = \lambda_3 = 0$
- $\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{w}_3$ form an orthonormal basis.
- The connected components collapse to $(100), (010), (001)$.
- Clustering is trivial in this case.

$$\mathbf{W} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{Y} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

Figure: Spectral clustering: The ideal Case

Other applications of graph Laplacian(2) -spectral clustering

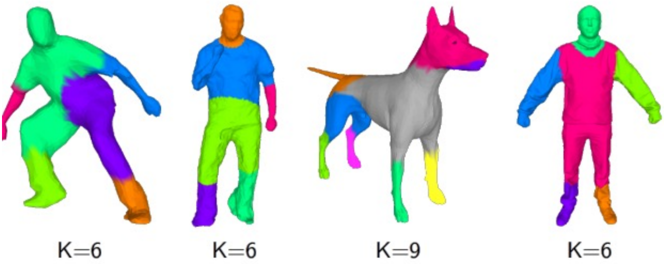


Figure: Mesh Segmentation Using Spectral Clustering

Other applications of graph Laplacian(3) -spectral clustering for image segmentation

- The image is seen as a graph of connected vertices, and the spectral clustering algorithm is used to do the graph cuts.
- Thus, image segmentation = Graph partitioning

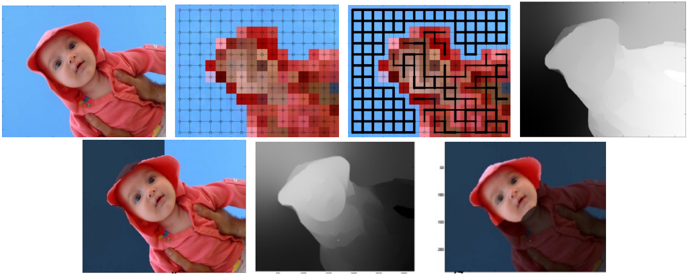
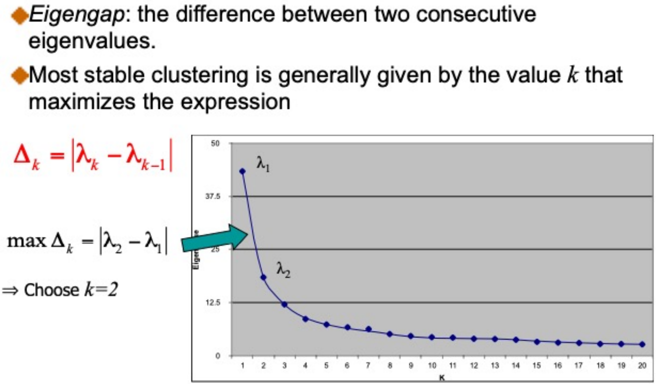


Figure: original image, grid, random walk, the second eigenvector, the second eigenvector sparsest cut, the 3^{rd} eigenvector and its sparsest cut

Conclusions to Laplacian embedding

- Note that $tr(\mathbf{f}^\top L\mathbf{f}) = \sum_{i,j=1}^{n} w_{ij} \| \mathbf{f}_i (\mathbf{x}_i) - \mathbf{f}_i (\mathbf{x}_j) \|^2$ quantifies how much $\mathbf{f}$ varies locally or how smooth it is over each vertex on graph. More specifically, the smaller the values of $tr(\mathbf{f}^\top L\mathbf{f})$, the smoother the vector $\mathbf{f}$ on graph, which indicates lower frequencies on graph. A smooth vector $\mathbf{f}$ assigns similar values to the instances that are close to each other on graph
- For eigendecomposition to $L = \mathbf{U}\Lambda\mathbf{U}^\top$ and $L\lambda_k = \lambda_k \mathbf{u}_k$, each λ_k can be interpreted as *graph frequencies* and eigenvectors $\mathbf{U}$ interpreted as corresponding graph frequency components. As the eigenvalue index increases(graph frequencies increasing), the number of oscillations tends to increase as well.
- The larger eigen-gap the mth and $(m+1)$th eigenvalues indicates that the high dimensional input patterns lie to a good approximation in a lower dimensional subspace of dimensionality m to PCA and MDS, also having the similar result in Lapalcian matrix to guide us how to chose the k.

Conclusions to Laplacian embedding



- *Eigengap*: the difference between two consecutive eigenvalues.
- Most stable clustering is generally given by the value k that maximizes the expression

$$\Delta_k = |\lambda_k - \lambda_{k-1}|$$

$$\max \Delta_k = |\lambda_2 - \lambda_1|$$

$\Rightarrow$ Choose k=2

50
37.5
25
12.5
0
Eige
λ_1
λ_2
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
K

Figure: Spectral gap: Selecting k

Conclusions to Laplacian embedding

- The set eigenvectors $\mathbf{U}$ of is also called Graph Fourier Transformation(GFT), and provides an orthogonal basis with increased variation each additional basis vector minimizes the increase in variation while guaranteeing orthogonality.
- Remember that k-dimensional Laplacian embedding to n vertices $X = \left\{\mathbf{x}_1^T; \mathbf{x}_2^T; \ldots; \mathbf{x}_n^T\right\}$ on graph,

$$\overline{\mathbf{U}} = \underbrace{\begin{bmatrix} \boldsymbol{u}_2\left(\mathbf{x}_1\right) & \ldots & \boldsymbol{u}_{k+1}\left(\mathbf{x}_1\right) \\ \vdots & & \vdots \\ \boldsymbol{u}_2\left(\mathbf{x}_n\right) & \ldots & \boldsymbol{u}_{k+1}\left(\mathbf{x}_n\right) \end{bmatrix}}_{\text{Fourier basis of graph}}$$

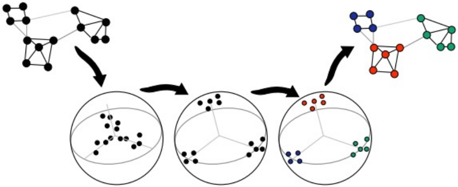
We can rewrite it as follows

$$\hat{X} = \overline{\mathbf{U}}^\top X = \begin{bmatrix} \boldsymbol{u}_2\left(\cdot\right) & \ldots & \boldsymbol{u}_2\left(\cdot\right) \\ \vdots & & \vdots \\ \boldsymbol{u}_{k+1}\left(\cdot\right) & \ldots & \boldsymbol{u}_{k+1}\left(\cdot\right) \end{bmatrix} \begin{bmatrix} \mathbf{x}_1^T \\ \vdots \\ \mathbf{x}_n^T \end{bmatrix}$$

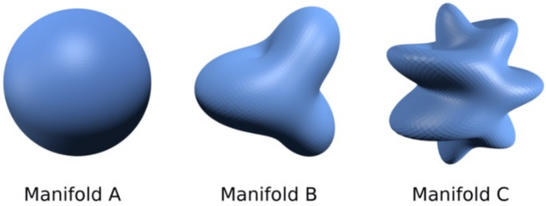
This GFT decompose a graph-signal $\mathbf{X}$ into its frequency components via $\hat{X} = \mathbf{U}^\top \mathbf{X}$ like DCT.

Conclusions to Laplacian embedding-cont'd

- The eigenvectors of the Laplacian (normalized or otherwise) associated the smallest k eigenvalue provide an embedding of the graph vertices in R^k;
- In all applications where the graph signals exhibit a cluster behavior, meaning that the signal is relatively smooth within each cluster, whereas it can vary arbitrarily from cluster to cluster, the GFT helps emphasizing the presence of clusters.
- And this embedding project the vertices of graph onto the unit sphere; Then some applications such as spectral partitioning can be performed on this unit sphere using a $k-$means or random hyperplanes.



Laplacian Beltrami operator—Manifold



- Above image show that low-D(2-dimensional manifold)surface embedded in high-D space(three-dimensional Euclidean space)
- A manifold is a space that is locally Euclidean.
- The graph can be viewed as a discretized approximation of the submanifold sampled by the input patterns.

Laplacian Beltrami operator—Manifold cont'd

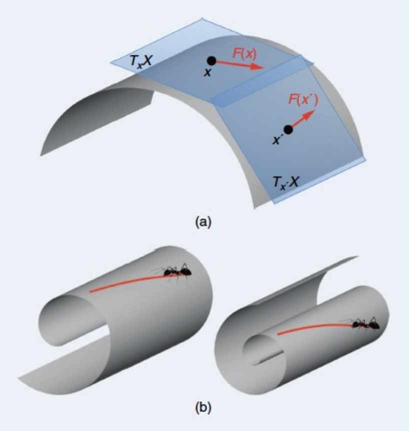


Figure: (a) The tangent space and tangent vectors on a 2-D manifold (surface); (b) Example of isometric deformations

Laplacian Beltrami operator—Manifold cont'd

- ***d*-dimensional manifold:** A (differentiable) d-dimensional manifold $\mathcal{X}$ is a topological space where each point x has a neighborhood that is topologically equivalent (homeomorphic) to a d-dimensional Euclidean space, called the tangent space and denoted by $T_x\mathcal{X}$
- **Tangent bundle:** The collection(disjoint union) of tangent spaces at all points denoted by $T\mathcal{X}$.
- **Inner product $T_x\mathcal{X}$:** $\langle \cdot, \cdot \rangle_{T_x X:} T_x\mathcal{X} \times T_x\mathcal{X} \to \mathbb{R}$, which is additionally assumed to depend smoothly on the position x and called a *Riemannian metric* in differential geometry and allows performing local measurements of angles, distances, and volumes. A manifold equipped with a metric is called a *Riemannian manifold.*
- A Riemannian manifold can be realized as a subset of a Euclidean space (in which case it is said to be embedded in that space) by using the structure of the Euclidean space to induce a Riemannian metric.

Laplacian Beltrami operator—Manifold cont'd

- An embedding is not necessarily unique; two different realizations of a Riemannian metric are called *isometries.*

Figure: Example of isometric deformations

- Two-dimensional (2-D) manifolds (surfaces) embedded into R^3 are used in computer graphics and vision to describe boundary surfaces of 3-D shapes.

Laplacian Beltrami operator—Manifold cont'd

- **Intrinsic**: Isometries do not affect the metric structure of the manifold, and consequently, they preserve any quantities that can be expressed in terms of the Riemannian metric. For example, the insect insect would not notice any difference induced by isometries. The insect in fact does not even know of the existence of the embedding space, as its only world is 2-D. This is an intrinsic viewpoint.
- **Extrinsic:** A human observer, on the other hand, sees a surface in 3-D space—this is an extrinsic point of view.

Laplacian Beltrami operator— Calculus on Manifold

- A scaler field smooth real function: $f : \mathcal{X} \rightarrow \mathbb{R}$ on the manifold;
- A tangent vector field $F : \mathcal{X} \rightarrow T\mathcal{X}$ is a mapping attaching a tangent vector $F(x) \in T_x\mathcal{X}$ to each point x
- **Two Hilbert spaces:** The scaler function Hilbert space $L^2(\mathcal{X})$, by defining the inner products:

  $$\langle f, g \rangle_{L^2(\mathcal{X})} = \int_{\mathcal{X}} f(x)g(x)dx$$

  and the Hilbert space of vector fields $L^2(T\mathcal{X})$, by defining the inner products:

  $$\langle F, G \rangle_{L^2(T\mathcal{X})} = \int_{\mathcal{X}} \langle F(x), G(x) \rangle_{T_x\mathcal{X}} dx$$

  Here, dx denotes a d-dimensional volume element induced by the Riemannian metric.

Laplacian Beltrami operator— Calculus on Manifold cont'd

- **Derivative on manifold** : lack of vector space structure on the manifold, so $f(x+dx)$ does not make sense; require to work locally in the tangent space, i.e. $T_x\mathcal{X}$;
- **Intrinsic gradient**: $\nabla f : L^2(\mathcal{X}) \to L^2(T\mathcal{X})$, direction steepest change at point $x \in \mathcal{X}$ but with difference compared to classical conception that it is now a tangent vector.
- **Directional derivative:** $df(x) : T\mathcal{X} \to R$ can be defined as a linear functional

$$df(x) = \langle \nabla f(x), \cdot \rangle_{T_x\mathcal{X}}$$

acting on tangent vectors $F(x) \in T_x\mathcal{X}$

Laplacian Beltrami operator— Calculus on Manifold cont'd

- **Intrinsic divergence:** div: $L^2(T\mathcal{X}) \to L^2(\mathcal{X})$ is an operator acting on tangent vector fields and is adjoint to the ∇f,

$$\langle F, \nabla f\rangle_{L^2(TX)} = \langle \nabla^* F, f\rangle_{L^2(X)} = \langle -\operatorname{div} F, f\rangle_{L^2(X)} \tag{5}$$

Proof: In general, for a function f and a vector field F, we have $\nabla \cdot (fF) = \langle \nabla f, F\rangle + f\nabla \cdot F$. Let $\Omega \subset \mathcal{X}$ is an open set of finite measure with a sufficiently nice boundary $\partial\Omega$ and by divergence theorem, we have

$$\int_{\partial\Omega} (fF) \cdot \vec{n} dS = \int_{\Omega} \nabla \cdot (fF) dV = \int_{\Omega} \langle \nabla f, F\rangle dV + \int_{\Omega} f\nabla \cdot F dV$$

where $\vec{n}$ is an outward pointing unit vector field on $\partial\Omega$. If we now make an additional assumption, such as $\partial\Omega = \emptyset$ i.e. Ω is without boundary, or that f or F vanish on $\partial\Omega = \emptyset$, we have

$$\int_{\Omega} \langle \nabla f, F\rangle dV = -\int_{\Omega} f\nabla \cdot F dV$$

Laplacian Beltrami operator

- **Laplacian Beltrami operator:** $\Delta : L^2(\mathcal{X}) \to L^2(\mathcal{X})$ is an operator

  $$\Delta f = -div(\nabla f)$$

  By (**??**), it can be seen that Laplacian is self-adjoint (symmetric), i.e.

  $$\langle \nabla f, \nabla f \rangle_{L^2(TX)} = \langle \Delta f, f \rangle_{L^2(X)} = \langle f, \Delta f \rangle_{L^2(X)} \qquad (6)$$

- **Dirichlet energy:** to measure the *smoothness* of a scalar field on the manifold.

  $$\min_f \mathcal{E}_{Dir}(f) = \langle \nabla f, \nabla f \rangle_{L^2(TX)} = \int_{\mathcal{X}} \|\nabla f\|^2_{T_x \mathcal{X}} dx \qquad (7)$$

Laplacian Beltrami operator cont'd

- The Laplacian can be interpreted as the difference between the average of a function on an infinitesimal sphere around a point and the value of the function at the point itself.
- It is important to note that all the previous definitions are coordinate free. By defining a basis in the tangent space, it is possible to express tangent vectors as d-dimensional vectors and the Riemannian metric as a $d \times d$ symmetric positive-definite matrix.

Discrete manifolds—Graph

- many practical situations in which one is given a sampling of points arising from a manifold but not the manifold itself. In computer graphics applications, reconstructing a correct discretization of a manifold from a point cloud is a difficult problem of its own, referred to as meshing.
- In manifold-learning problems, the manifold is typically approximated as a graph capturing the local affinity structure.

Reference:

1. Michael M. Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst, Geometric deep learning-going beyond Euclidean data,IEEE Signal Processing Magazine, 2017;