Review: Summary questions of the last Lecture

- Explain the meaning of Spectral Clustering in one sentence. Why spectral?

→ It means that the clustering of nodes in a graph is done on the basis of frequency components of the graph signal representing the cluster labels of the nodes.

- What is represented by the solution of Laplacian formulation relaxing Balanced Graph cut problem?

→ It represents the second eigenvector of an appropriate Laplacian, which gives the given balancing condition.

- What does the solution of MinCut problem mean?

→ It is the second eigenvector of Laplacian, which gives the balanced cardinalities of the clustered groups.

- What does the solution of NormalizedCut problem mean?
- It is the second eigenvector of Symetric Laplacian, which gives the balanced volumes of the clustered groups.

Spectral Clustering: Relaxing Balanced Cuts

Optimization formulation for spectral clustering

$$\min_{\boldsymbol{f}} \boldsymbol{f}^T \boldsymbol{L} \boldsymbol{f} \text{ subject to } f_i \in R,\ \ \boldsymbol{f} \perp \mathbf{1}_N,\ \ \|\boldsymbol{f}\| = \sqrt{\boldsymbol{N}}$$

$$f_i \in \{1, -1\}$$

The solution

$$\lambda_2 = \min_{\boldsymbol{x}^T\boldsymbol{x}=\mathbf{1},\ \boldsymbol{x}\perp\boldsymbol{u}_1} \boldsymbol{x}^T \boldsymbol{L} \boldsymbol{x}, \text{ where } \boldsymbol{u}_1 = \mathbf{1}_N \text{ for } \lambda_1 = 0$$

$\rightarrow$ second eigenvector $\boldsymbol{x}$ of $\boldsymbol{L}$

Since the elements in $\boldsymbol{x}$ are not integer and $\|\boldsymbol{x}\| = \mathbf{1}$, $\boldsymbol{f}$ can be obtained by

$$f_i = \begin{cases} 1 & if\ \ x_i \geq 0 \\ -1 & if\ \ x_i < 0 \end{cases} \rightarrow \|\boldsymbol{f}\| = \sqrt{\boldsymbol{N}} \quad \not\Rightarrow \quad \boxed{|\boldsymbol{A}| = |\boldsymbol{B}|}$$

Spectral Clustering: Approximating RatioCut

RatioCut

$$\min_{\boldsymbol{A},\boldsymbol{B}} RCut(\boldsymbol{A},\boldsymbol{B}) = \min_{\boldsymbol{A},\boldsymbol{B}} \sum_{i\in \boldsymbol{A}, j\in \boldsymbol{B}} w_{ij}\left(\frac{1}{|\boldsymbol{A}|}+\frac{1}{|\boldsymbol{B}|}\right)$$

Define graph function $\boldsymbol{f}$ for cluster membership of RatioCut: $f_i = \begin{cases} \sqrt{\frac{|\boldsymbol{B}|}{|\boldsymbol{A}|}} & if\ v_i \in \boldsymbol{A} \\ -\sqrt{\frac{|\boldsymbol{A}|}{|\boldsymbol{B}|}} & if\ v_i \in \boldsymbol{B} \end{cases}$

$$\boldsymbol{f}^T\boldsymbol{L}\boldsymbol{f} = \frac{1}{2}\sum_{i,j} w_{ij}\,(f_i - f_j)^2 = (|\boldsymbol{A}| + |\boldsymbol{B}|)RCut(\boldsymbol{A},\boldsymbol{B})$$

Since $(|\boldsymbol{A}| + |\boldsymbol{B}|)$ is constant,

$$\min_{\boldsymbol{A},\boldsymbol{B}} RCut(\boldsymbol{A},\boldsymbol{B}) = \min_{\boldsymbol{f}} \ \boldsymbol{f}^T\boldsymbol{L}\boldsymbol{f},$$

$$\text{subject to } \ f_i \in \left\{\sqrt{\frac{|\boldsymbol{B}|}{|\boldsymbol{A}|}}, -\sqrt{\frac{|\boldsymbol{A}|}{|\boldsymbol{B}|}}\right\}$$

➡ $\boxed{|\boldsymbol{A}| = |\boldsymbol{B}|}$

Still NP hard…Require relaxation.

Spectral Clustering: Approximating RatioCut

$$\boxed{\min_{\boldsymbol{A},\boldsymbol{B}} RCut(\boldsymbol{A},\boldsymbol{B}) = \min_{\boldsymbol{A},\boldsymbol{B}} \ \boldsymbol{f}^T\boldsymbol{L}\boldsymbol{f}} \qquad s.t.\ f_i = \begin{cases} \sqrt{\frac{|\boldsymbol{B}|}{|\boldsymbol{A}|}} & if\ v_i \in \boldsymbol{A} \\ -\sqrt{\frac{|\boldsymbol{A}|}{|\boldsymbol{B}|}} & if\ v_i \in \boldsymbol{B} \end{cases}$$

$\|\boldsymbol{f}\|^2 = \sum_i f_i^2 = |\boldsymbol{A}|\frac{|\boldsymbol{B}|}{|\boldsymbol{A}|} + |\boldsymbol{B}|\frac{|\boldsymbol{A}|}{|\boldsymbol{B}|} = |\boldsymbol{A}| \ + |\boldsymbol{B}| \ = \boldsymbol{N} \rightarrow$ not sufficient for $f_i \in \left\{\sqrt{\frac{|\boldsymbol{B}|}{|\boldsymbol{A}|}}, -\sqrt{\frac{|\boldsymbol{A}|}{|\boldsymbol{B}|}}\right\}$

Optimization formulation for RatioCut (same with balanced mincut)

$\min_{\boldsymbol{f}} \boldsymbol{f}^T\boldsymbol{L}\boldsymbol{f}$ subject to, $f_i \in R$, $\|\boldsymbol{f}\| = \sqrt{\boldsymbol{N}}$

$$\boxed{\boldsymbol{f}^T\boldsymbol{L}\boldsymbol{f} \neq (|\boldsymbol{A}| + |\boldsymbol{B}|)\sum_{i\in \boldsymbol{A}, j\in \boldsymbol{B}} w_{ij}\left(\frac{1}{|\boldsymbol{A}|} + \frac{1}{|\boldsymbol{B}|}\right) \ \not\Rightarrow \ |\boldsymbol{A}| = |\boldsymbol{B}|}$$

Spectral Clustering: Approximating RatioCut

Optimization formulation for RatioCut (same with balanced mincut)

$$\min_{\boldsymbol{f}} \boldsymbol{f}^T \boldsymbol{L} \boldsymbol{f} \text{ subject to } f_i \in R,\ \|\boldsymbol{f}\| = \sqrt{\boldsymbol{N}}$$

$$|\boldsymbol{A}| = |\boldsymbol{B}| \ \rightarrow\ \sum_i f_i = 0 \ \leftrightarrow \boldsymbol{f} \perp \boldsymbol{1}_{\boldsymbol{N}}$$

Optimization formulation for RatioCut (same with balanced Mincut)

$$\min_{\boldsymbol{f}} \boldsymbol{f}^T \boldsymbol{L} \boldsymbol{f} \text{ subject to } f_i \in R, \boldsymbol{f} \perp \boldsymbol{1}_{\boldsymbol{N}},\ \|\boldsymbol{f}\| = \sqrt{\boldsymbol{N}}$$

Spectral Clustering: Approximating RatioCut

The solution

$$\lambda_2 = \min_{\boldsymbol{x}^T\boldsymbol{x}=\mathbf{1},\, \boldsymbol{x}\perp\boldsymbol{u}_1} \boldsymbol{x}^T\boldsymbol{L}\boldsymbol{x}, \text{ where } \boldsymbol{u}_1 = \mathbf{1}_N \text{ for } \lambda_1 = 0.$$

$\rightarrow$ second eigenvector $\boldsymbol{x}$ of $\boldsymbol{L}$

Since the elements in $\boldsymbol{x}$ are not integer and $\|\boldsymbol{x}\| = \mathbf{1}$, $\boldsymbol{f}$ can be obtained by

$$f_i = \begin{cases} \sqrt{\dfrac{|\boldsymbol{B}|}{|\boldsymbol{A}|}} & if \;\; x_i \geq 0 \\ -\sqrt{\dfrac{|\boldsymbol{A}|}{|\boldsymbol{B}|}} & if \;\; x_i < 0 \end{cases} \rightarrow \|\boldsymbol{f}\| = \sqrt{\boldsymbol{N}}$$

$$\boxed{\boldsymbol{f}^T\boldsymbol{L}\boldsymbol{f} \neq (|\boldsymbol{A}| + |\boldsymbol{B}|)\sum_{i\in \boldsymbol{A}, j\in \boldsymbol{B}} w_{ij}\left(\frac{1}{|\boldsymbol{A}|} + \frac{1}{|\boldsymbol{B}|}\right) \nRightarrow |\boldsymbol{A}| = |\boldsymbol{B}|}$$

Spectral Clustering: Approximating NormalizedCut

NormalizedCut

$$\min_{\boldsymbol{A},\boldsymbol{B}} NCut(\boldsymbol{A},\boldsymbol{B}) = \min_{\boldsymbol{A},\boldsymbol{B}} \sum_{i\in \boldsymbol{A}, j\in \boldsymbol{B}} w_{ij}\left(\frac{1}{vol(\boldsymbol{A})} + \frac{1}{vol(\boldsymbol{B})}\right)$$

Balancing the clusters by considering the degrees of nodes

Define graph function $\boldsymbol{f}$ for cluster membership of NCut: $f_i = \begin{cases} \sqrt{\dfrac{vol(\boldsymbol{B})}{vol(\boldsymbol{A})}} & if\ v_i \in \boldsymbol{A} \\ -\sqrt{\dfrac{vol(\boldsymbol{A})}{vol(\boldsymbol{B})}} & if\ v_i \in \boldsymbol{B} \end{cases}$

$$\boldsymbol{f}^T \boldsymbol{L} \boldsymbol{f} = \sum_{i,j} w_{ij}\left(\sqrt{\frac{vol(\boldsymbol{B})}{vol(\boldsymbol{A})}} + \sqrt{\frac{vol(\boldsymbol{A})}{vol(\boldsymbol{B})}}\right)^2 = \sum_{i,j} w_{ij}\left(\frac{vol(\boldsymbol{B}) + vol(\boldsymbol{A})}{\sqrt{vol(\boldsymbol{A})}\sqrt{vol(\boldsymbol{B})}}\right)^2$$

$$\min_{A,B} \boldsymbol{f}^T \boldsymbol{L} \boldsymbol{f} = vol(\mathcal{V}) NCut(\boldsymbol{A},\boldsymbol{B}) \quad , f_i \in \left\{\sqrt{\frac{vol(\boldsymbol{B})}{vol(\boldsymbol{A})}}, -\sqrt{\frac{vol(\boldsymbol{A})}{vol(\boldsymbol{B})}}\right\} \Rightarrow vol(\boldsymbol{B}) = vol(\boldsymbol{A})$$

$\boxed{\text{NP hard of assignment } f_i.}$

Spectral Clustering: Approximating NormalizedCut

NormalizedCut

$$\min_{\boldsymbol{A},\boldsymbol{B}} NCut(\boldsymbol{A},\boldsymbol{B}) = \min_{\boldsymbol{A},\boldsymbol{B}} \sum_{i\in\boldsymbol{A},j\in\boldsymbol{B}} w_{ij}\left(\frac{1}{vol(\boldsymbol{A})}+\frac{1}{vol(\boldsymbol{B})}\right)$$

Define graph function $\boldsymbol{f}$ for cluster membership of NCut: $f_i = \begin{cases} \sqrt{\dfrac{vol(\boldsymbol{B})}{vol(\boldsymbol{A})}} & if\ v_i \in \boldsymbol{A} \\ -\sqrt{\dfrac{vol(\boldsymbol{A})}{vol(\boldsymbol{B})}} & if\ v_i \in \boldsymbol{B} \end{cases}$

Necessary condition for $f_i \in \left\{\sqrt{\dfrac{vol(\boldsymbol{B})}{vol(\boldsymbol{A})}}, -\sqrt{\dfrac{vol(\boldsymbol{A})}{vol(\boldsymbol{B})}}\right\}$

$$(\boldsymbol{D}\boldsymbol{f})^T\mathbf{1}_N = 0, \qquad \boldsymbol{f}^T\boldsymbol{D}\boldsymbol{f} = vol(\mathcal{V})$$

Spectral Clustering: Approximating NormalizedCut

NormalizedCut

$$\min_{\boldsymbol{A},\boldsymbol{B}} NCut(\boldsymbol{A},\boldsymbol{B}) = \min_{\boldsymbol{A},\boldsymbol{B}} \sum\nolimits_{i\in\boldsymbol{A},j\in\boldsymbol{B}} w_{ij}\left(\frac{1}{vol(\boldsymbol{A})}+\frac{1}{vol(\boldsymbol{B})}\right)$$

Define graph function $\boldsymbol{f}$ for cluster membership of NCut: $f_i = \begin{cases} \sqrt{\frac{vol(\boldsymbol{B})}{vol(\boldsymbol{A})}} & if\ v_i \in \boldsymbol{A} \\ -\sqrt{\frac{vol(\boldsymbol{A})}{vol(\boldsymbol{B})}} & if\ v_i \in \boldsymbol{B} \end{cases}$

$$\boldsymbol{f}^T\boldsymbol{L}\boldsymbol{f} = vol(\mathcal{V})NCut(\boldsymbol{A},\boldsymbol{B}), \qquad (\boldsymbol{D}\boldsymbol{f})^T\mathbf{1}_N = 0, \qquad \boldsymbol{f}^T\boldsymbol{D}\boldsymbol{f} = vol(\mathcal{V}),$$

Optimization formulation for NormalizedCut

$\min_{\boldsymbol{f}} \boldsymbol{f}^T\boldsymbol{L}\boldsymbol{f}$ subject to $f_i \in R$, $\quad \boldsymbol{D}\boldsymbol{f} \perp \mathbf{1}_N$, $\quad \boldsymbol{f}^T\boldsymbol{D}\boldsymbol{f} = vol(\mathcal{V})$

Spectral Clustering: Approximating NormalizedCut

Optimization formulation for NormalizedCut

$$\min_{\boldsymbol{f}} \boldsymbol{f}^T \boldsymbol{L}\boldsymbol{f} \text{ subject to } f_i \in R, \quad \boldsymbol{D}\boldsymbol{f} \perp \mathbf{1}_N, \quad \boldsymbol{f}^T\boldsymbol{D}\boldsymbol{f} = vol(\mathcal{V})$$

Can we apply Rayleigh-Ritz now?

Define $\boldsymbol{h} = \boldsymbol{D}^{1/2}\boldsymbol{f}$

Optimization formulation for NormalizedCut

$$\min_{\boldsymbol{h}} \boldsymbol{h}^T \boldsymbol{D}^{-1/2}\boldsymbol{L}\boldsymbol{D}^{-1/2}\boldsymbol{h} \text{ subject to } h_i \in R, \quad \boldsymbol{h} \perp \boldsymbol{u}_{1,L_{sym}}, \quad \boldsymbol{h}^T\boldsymbol{h} = vol(\mathcal{V})$$

$$\min_{\boldsymbol{h}} \boldsymbol{h}^T \boldsymbol{L}_{sym}\boldsymbol{h} \text{ subject to } h_i \in R, \quad \boldsymbol{h} \perp \boldsymbol{u}_{1,L_{sym}}, \quad ||\boldsymbol{h}|| = \sqrt{vol(\mathcal{V})}$$

Spectral Clustering: Approximating NormalizedCut

Optimization formulation for NormalizedCut

$$\min_{\boldsymbol{h}} \boldsymbol{h}^T \boldsymbol{L_{sym}} \boldsymbol{h} \text{ subject to } h_i \in R, \quad \boldsymbol{h} \perp \boldsymbol{u_{1,L_{sym}}}, \quad \|\boldsymbol{h}\| = \sqrt{vol(\mathcal{V})}$$

Solution by Rayleigh-Ritz?

$$\boldsymbol{h} = \boldsymbol{u_{2,L_{sym}}}, \quad \boldsymbol{f} = \boldsymbol{D}^{-1/2}\boldsymbol{h}$$

$\rightarrow$ eigenvector of $\boldsymbol{L_{rw}}$

$\rightarrow Lu = \lambda Du$

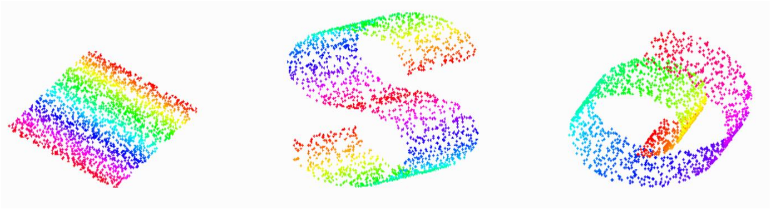
$$f_i \leftarrow \begin{cases} \sqrt{\dfrac{vol(\boldsymbol{B})}{vol(\boldsymbol{A})}} & if \;\; h_i \geq 0 \\ -\sqrt{\dfrac{vol(\boldsymbol{A})}{vol(\boldsymbol{B})}} & if \;\; h_i < 0 \end{cases} \quad \leftrightarrow \boldsymbol{f}^T \boldsymbol{D} \boldsymbol{f} = vol(\mathcal{V}) \nrightarrow vol(\boldsymbol{A}) = vol(\boldsymbol{B})$$

Outline of Lecture

- Graph Spectral Theory
  - Definition of Graph
  - Graph Laplacian
  - Laplacian Smoothing
- Graph Node Clustering
  - Minimum Graph Cut
  - Ratio Graph Cut
  - Normalized Graph Cut
- Manifold Learning
  - Spectral Analysis in Riemannian Manifolds
  - Dimension Reduction, Node Embedding
- Semi-supervised Learning (SSL)
  - Self-Training Methods
  - SSL with SVM
  - SSL with Graph using MinCut
  - SSL with Graph using Harmonic Functions
- Semi-supervised Learning (SSL) : conti.
  - SSL with Graph using Regularized Harmonic Functions
  - SSL with Graph using Soft Harmonic Functions
  - SSL with Graph using Manifold Regularization
  - SSL with Graph using Laplacian SVMs
  - SSL with Graph using Max-Margin Graph Cuts
  - Online SSL and SSL for large graph
- Graph Convolution Networks (GCN)
  - Graph Filtering in GCN
  - Graph Pooling in GCN
  - Spectral Filtering in GCN
  - Spatial Filtering in GCN
- Recent GCN papers

Manifold Learning

$$\mathbb{R}^d \Longrightarrow \mathbb{R}^m$$



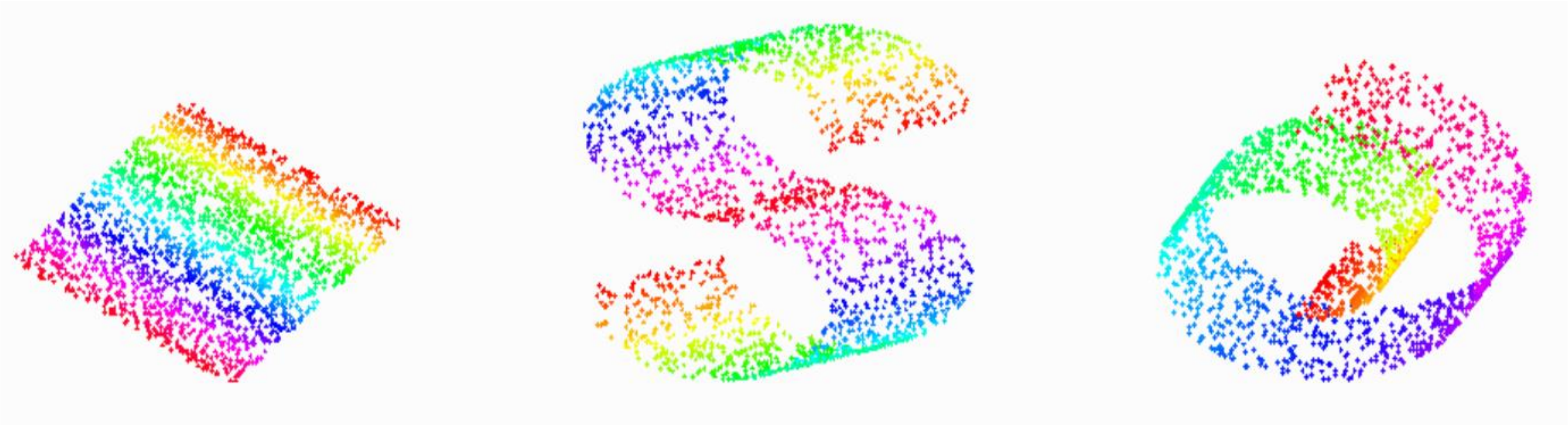
Manifold Learning

problem: definition reduction/manifold learning

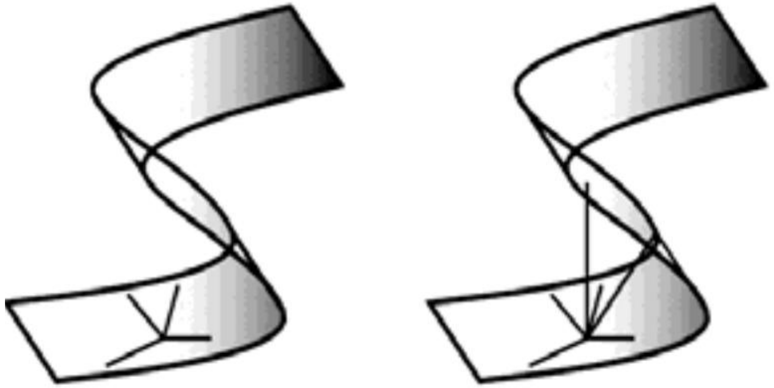
Given $\{\boldsymbol{x}_i\}_{i=1}^{N}$ from $\mathbb{R}^d$, find $\{\boldsymbol{y}_i\}_{i=1}^{N}$ in $\mathbb{R}^m$, where $m \ll d$.

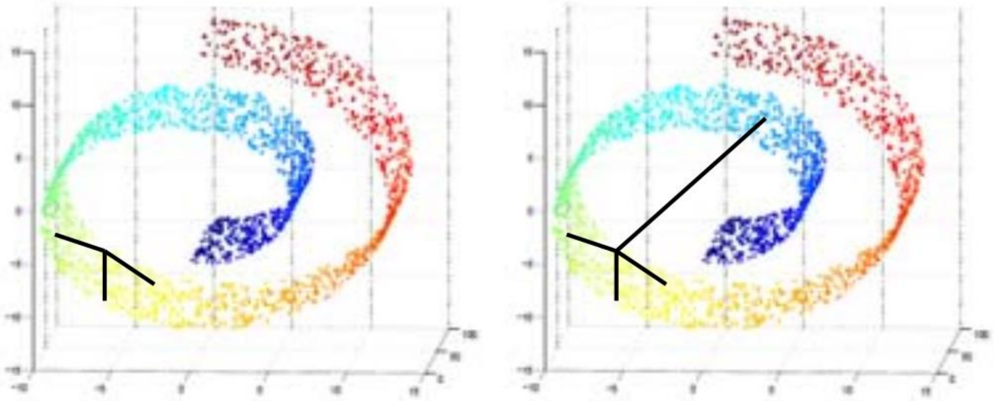
- What do we know about the dimensionality reduction?
  - representation/visualization ($2D\ or\ 3D$)
  - an old example: globe to a map ($3D \rightarrow 2D$)
  - often assuming $\mathcal{M} \subset \mathbb{R}^d$
  - feature extraction
  - linear vs. nonlinear dimensionality reduction
- What do we know about linear vs. nonlinear methods?
  - linear: ICA, PCA, LDA, SVD, ...
  - nonlinear often preserve only local distances

Manifold Learning: Linear vs. Non-linear



Manifold Learning: Preserving (just) local distances





$d(\boldsymbol{y}_i, \boldsymbol{y}_j) = d(\boldsymbol{x}_i, \boldsymbol{x}_j)$ only if $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $d(\boldsymbol{y}_i, \boldsymbol{y}_j)$ are small.

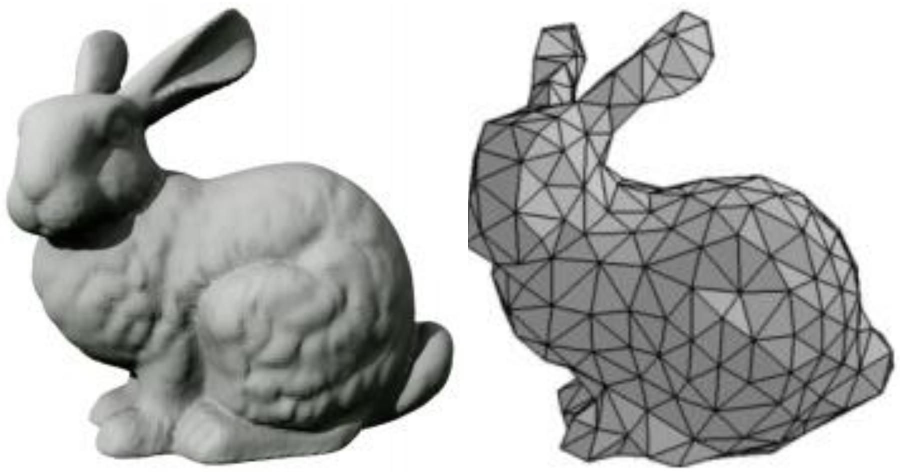
$$\min \sum_{i,j} w_{ij} \left\| \boldsymbol{y}_i - \boldsymbol{y}_j \right\|^2$$

$\boldsymbol{y}_i$ looks similar to $\boldsymbol{y}_j$?

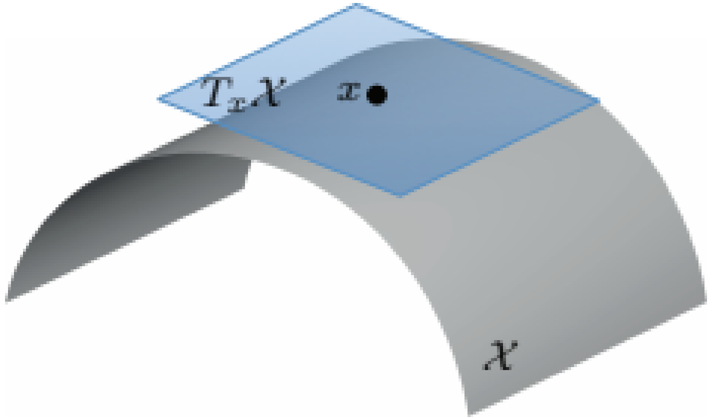
Yes in Euclidean space, but No in Manifolds

Manifold Learning: Riemannian manifolds

- Manifold $\mathcal{X}$ = topological space
- Tangent space $T_x\mathcal{X}$ = local Euclidean representation of manifold $\mathcal{X}$ around x



Manifolds



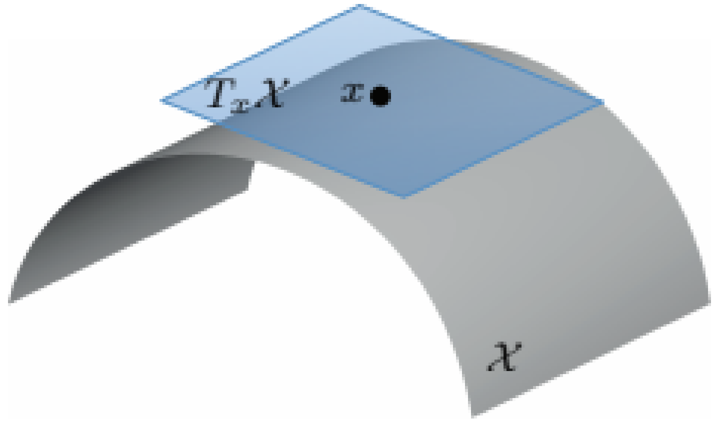
Tangent space

[Geometric deep learning on graphs and manifolds](), Michael Bronstein et al., SIAM Tutorial 12 July 2018, Portland

Manifold Learning: Riemannian manifolds

- Manifold $\boldsymbol{\mathcal{X}} =$ topological space
- Tangent space $T_{\boldsymbol{x}}\boldsymbol{\mathcal{X}} =$ local Euclidean representation of manifold $\boldsymbol{\mathcal{X}}$ around $\boldsymbol{x}$
- Riemannian metric describes the local intrinsic structure at $\boldsymbol{x}$

$$<\cdot,\cdot>_{T_{\boldsymbol{x}}\boldsymbol{\mathcal{X}}}: T_{\boldsymbol{x}}\boldsymbol{\mathcal{X}} \times T_{\boldsymbol{x}}\boldsymbol{\mathcal{X}} \rightarrow \mathbb{R}$$



Tangent space

Manifold Learning: Riemannian manifolds

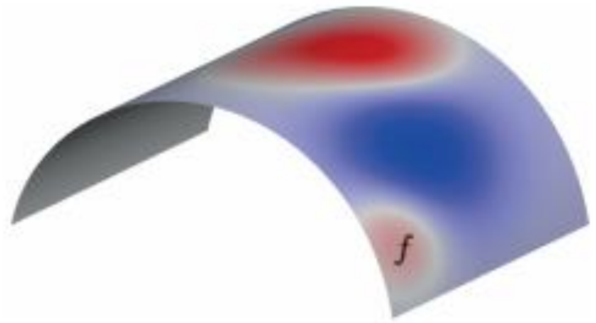
- Manifold $\boldsymbol{\mathcal{X}}$ = topological space

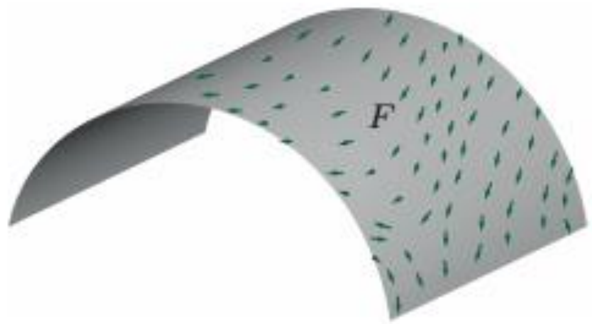
- Tangent space $T_{\boldsymbol{x}}\boldsymbol{\mathcal{X}} =$ local Euclidean representation of manifold $\boldsymbol{\mathcal{X}}$ around $\boldsymbol{x}$

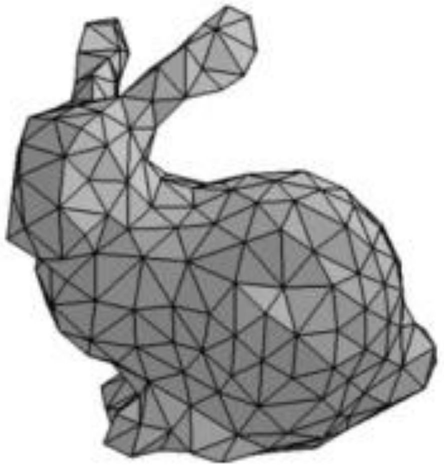
- Riemannian metric describes the local intrinsic structure at $\boldsymbol{x}$

$$<\cdot,\cdot>_{T_{\boldsymbol{x}}\boldsymbol{\mathcal{X}}} : T_{\boldsymbol{x}}\boldsymbol{\mathcal{X}} \times T_{\boldsymbol{x}}\boldsymbol{\mathcal{X}} \rightarrow \mathbb{R}$$

- Scalar fields $f : \boldsymbol{\mathcal{X}} \rightarrow \mathbb{R}$ and Vector fields $F : \boldsymbol{\mathcal{X}} \rightarrow T_{\boldsymbol{x}}\boldsymbol{\mathcal{X}}$







Manifold Learning: Riemannian manifolds

- Riemannian metric describes the local intrinsic structure at $\boldsymbol{x}$

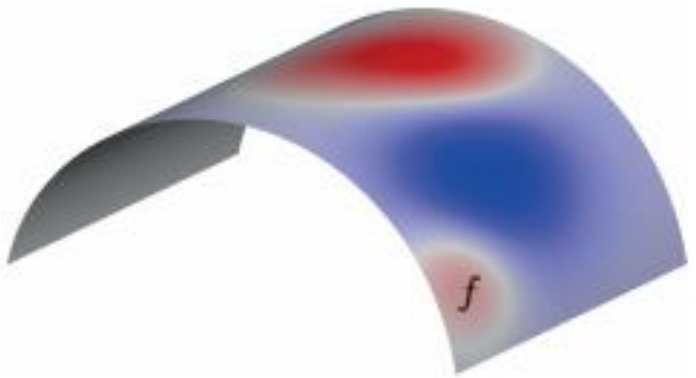
$$<\cdot,\cdot>_{T_x\mathcal{X}}: T_x\mathcal{X} \times T_x\mathcal{X} \to \mathbb{R}$$

- Scalar fields $f: \mathcal{X} \to \mathbb{R}$ and
  Vector fields $F: \mathcal{X} \to T_x\mathcal{X}$

- Hilbert spaces with inner products

$$< f, g >_{L^2(\mathcal{X},x)} = \int f(\boldsymbol{x}) g(\boldsymbol{x}) d\boldsymbol{x}$$

$$< F, G >_{L^2(T\mathcal{X},x)} = \int < F(\boldsymbol{x}), G(\boldsymbol{x}) >_{T_x\mathcal{X}} d\boldsymbol{x}$$



$L^2(\mathcal{X}, \boldsymbol{\mu})$:
square integrable
in measure space $\mathcal{X}$
w.r.t. measure $\boldsymbol{\mu}$.

Is $f(x) = 1/t$ in $L^2([\mathbf{0}, \mathbf{1}], \boldsymbol{x})$?

Manifold Learning: Manifold Laplacian

- **Laplacian** Δ: $L^2(\boldsymbol{\mathcal{X}}) \to L^2(\boldsymbol{\mathcal{X}})$

$$\Delta f(\boldsymbol{x}) = -div\nabla f(\boldsymbol{x})$$

  where gradient ∇:$L^2(\boldsymbol{\mathcal{X}}) \to L^2(T\boldsymbol{\mathcal{X}})$
  and divergence div:$L^2(T\boldsymbol{\mathcal{X}}) \to L^2(\boldsymbol{\mathcal{X}})$
  are adjoint operators, i.e.,

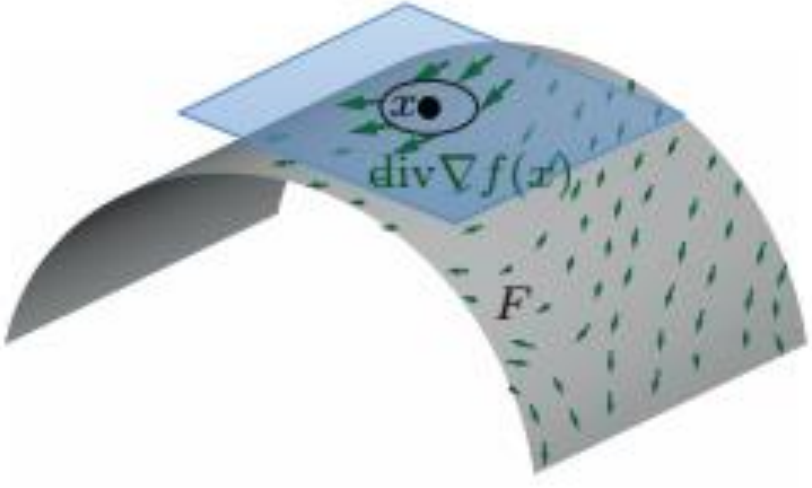
$$< \nabla f, G >_{L^2(T\boldsymbol{\mathcal{X}})} = < f, -div\ G >_{L^2(\boldsymbol{\mathcal{X}})}$$

  (ex) Let F be a differentiable vector field

$$\mathbf{F} = F_x\mathbf{i} + F_y\mathbf{j} + F_z\mathbf{k}$$

  Then

$$\text{div } \mathbf{F} = \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} + \frac{\partial F_z}{\partial z}$$



Book: Laplacian on Riemannian Manifold

Manifold Learning: Manifold Laplacian

- **Laplacian** Δ: $L^2(\boldsymbol{\mathcal{X}}) \to L^2(\boldsymbol{\mathcal{X}})$

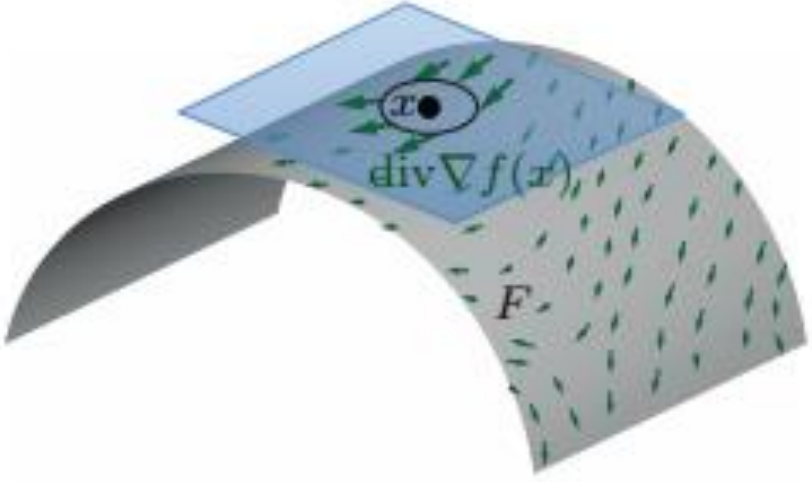
  $$\Delta f(\boldsymbol{x}) = -div\nabla f(\boldsymbol{x})$$

  where gradient ∇:$L^2(\boldsymbol{\mathcal{X}}) \to L^2(T\boldsymbol{\mathcal{X}})$
  and divergence div:$L^2(T\boldsymbol{\mathcal{X}}) \to L^2(\boldsymbol{\mathcal{X}})$
  are adjoint operators, i.e.,

  $$< \nabla f, G >_{L^2(T\boldsymbol{\mathcal{X}})} = < f, -div\ G >_{L^2(\boldsymbol{\mathcal{X}})}$$

- Manifold Laplacian is **self-adjoint**

  $$< \Delta f, f >_{L^2(\boldsymbol{\mathcal{X}})} = < f, \Delta f >_{L^2(\boldsymbol{\mathcal{X}})}$$



Manifold Learning: Manifold Laplacian

- **Laplacian** Δ: $L^2(\mathcal{X}) \to L^2(\mathcal{X})$

  $$\Delta f(\boldsymbol{x}) = -div\nabla f(\boldsymbol{x})$$

  where gradient ∇:$L^2(\mathcal{X}) \to L^2(T\mathcal{X})$
  and divergence div:$L^2(T\mathcal{X}) \to L^2(\mathcal{X})$
  are adjoint operators, i.e.,

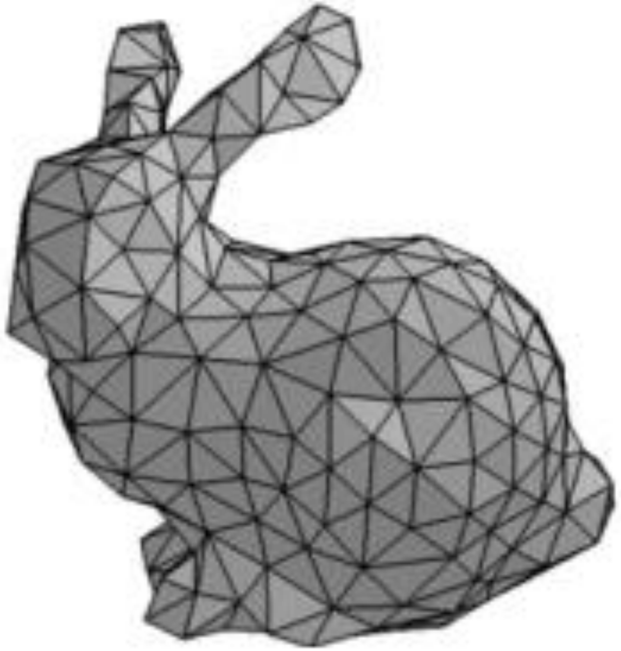
  $$< \nabla f, G >_{L^2(T\mathcal{X})} = < f, -div\ G >_{L^2(\mathcal{X})}$$

- Laplacian is **self-adjoint**

  $$< \Delta f, f >_{L^2(\mathcal{X})} = < f, \Delta f >_{L^2(\mathcal{X})}$$

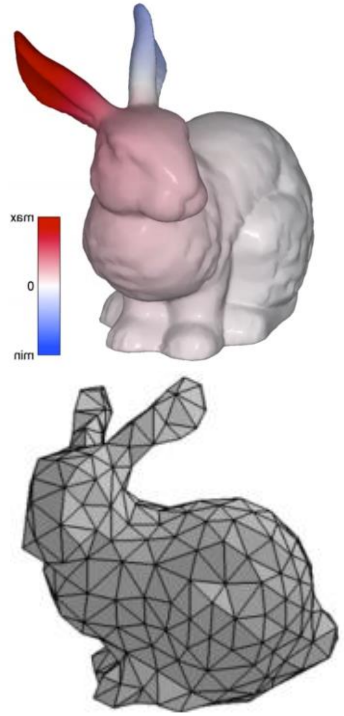
- Dirichlet energy of f

  $$< \nabla f, \nabla f >_{L^2(T\mathcal{X})} = \int f(\boldsymbol{x})\, \Delta f(\boldsymbol{x}) d\boldsymbol{x} \quad \Leftrightarrow \quad \boldsymbol{f}^T \boldsymbol{L}\boldsymbol{f}$$



Laplacian Eigenmaps for Dim. Reduction (Belken et al., ohio-state doc., 2002)

Manifold Learning: Laplacian Eigenmaps



Step 1 (Adjacency Graph): Given N points $\{\boldsymbol{x_1}, \boldsymbol{x_2}, \dots, \boldsymbol{x_N}\}$ in R^l, we construct a graph of N nodes $\{\boldsymbol{x_i}\}$ and weighted edges $\{w_{\boldsymbol{ij}}\}$, where

$$w_{\boldsymbol{ij}} = \begin{cases} \boldsymbol{e}^{\frac{-\left\|x_i - x_j\right\|^2}{t}} \text{ or } 1 & if \quad \left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\| \le \epsilon \ \text{ or } j \in \boldsymbol{kNN}_i \\ 0 & o.w. \end{cases}$$

Step 2: Solve generalized eigenproblem (Normalized Cut):

$$\boldsymbol{Lf} = \boldsymbol{\lambda Df} \quad \rightarrow \{\boldsymbol{f}_1, \boldsymbol{f}_2, \dots, \boldsymbol{f}_N\}, \text{ where } 0 = \lambda_1 \le \dots \le \lambda_N$$

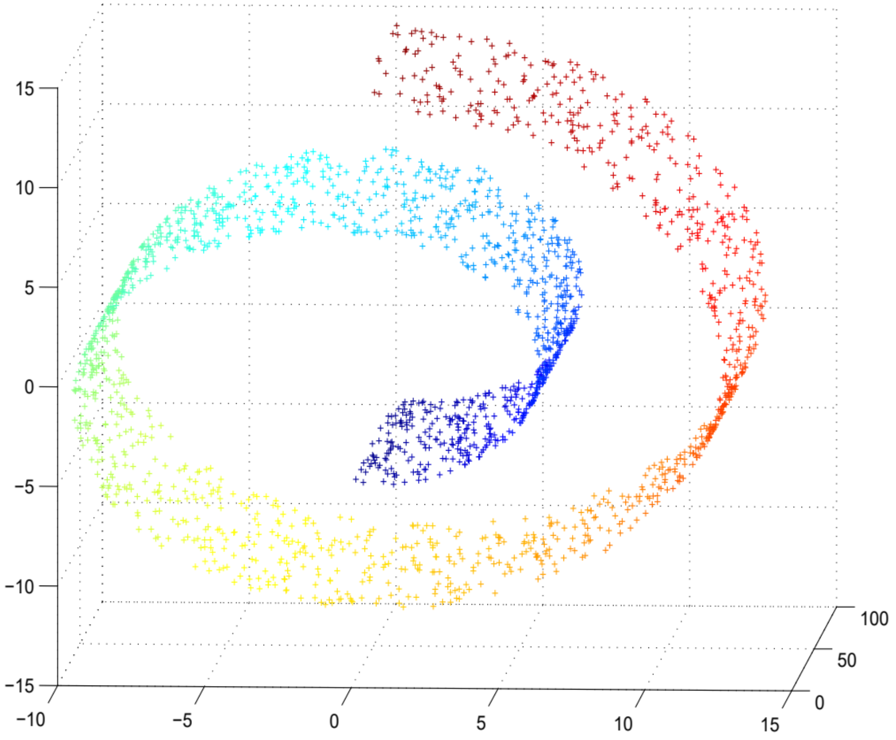
Step 3: Assign m new coordinates: use m eigenvectors for embedding $\boldsymbol{x_i}$ in m dimensional Euclidean space.

$$\boldsymbol{x_i} \rightarrow \{\boldsymbol{f}_2(i), \dots, \boldsymbol{f}_{m+1}(i)\}$$

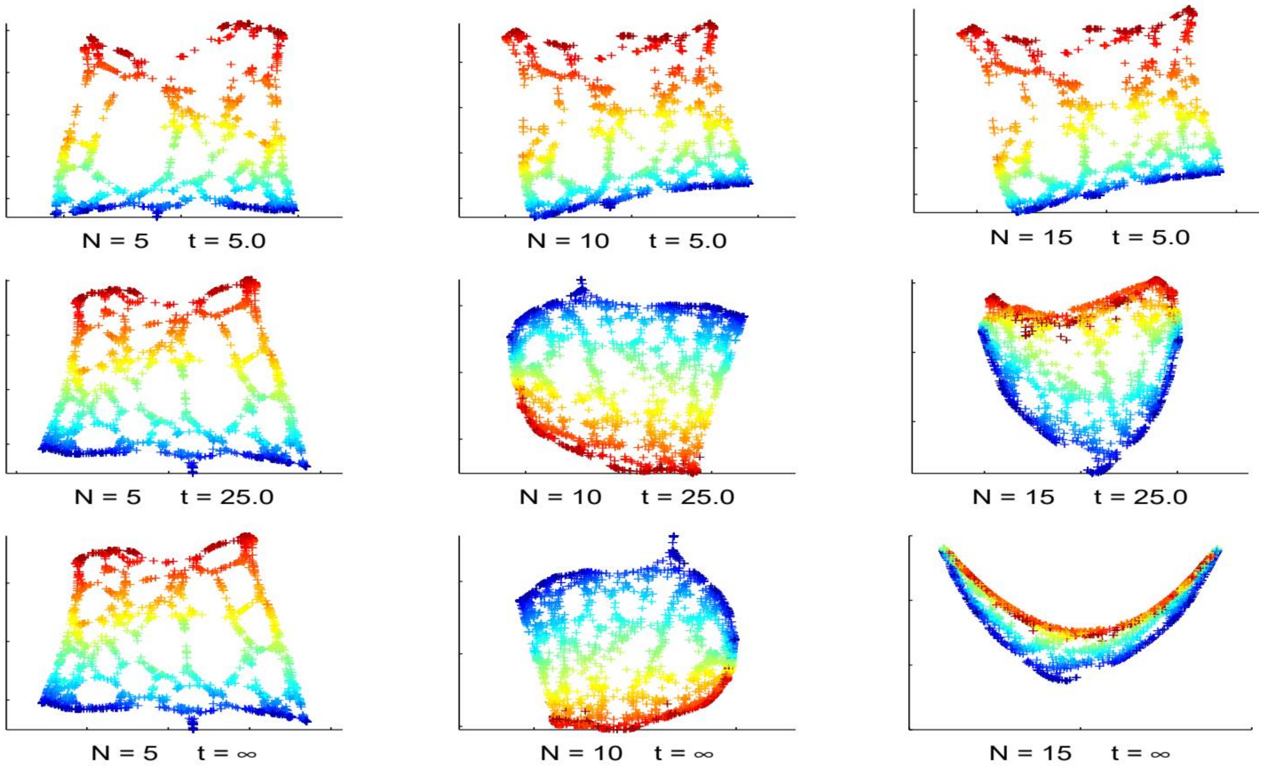
Laplacian Eigenmaps for Dim. Reduction (Belken et al., ohio-state doc., 2002)

Manifold Learning: Laplacian Eigenmaps

Swiss Roll

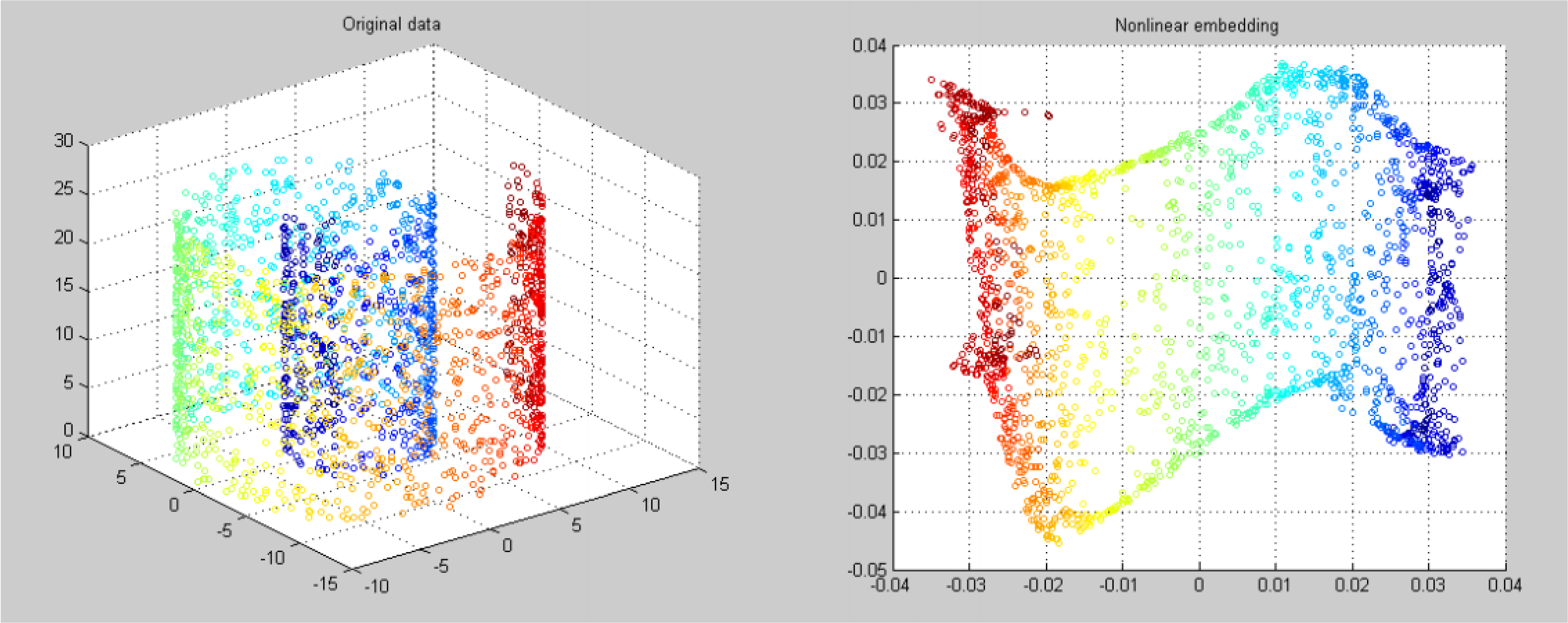


2D embeddings



N −nearest neighbors, t: heat kernel parameter

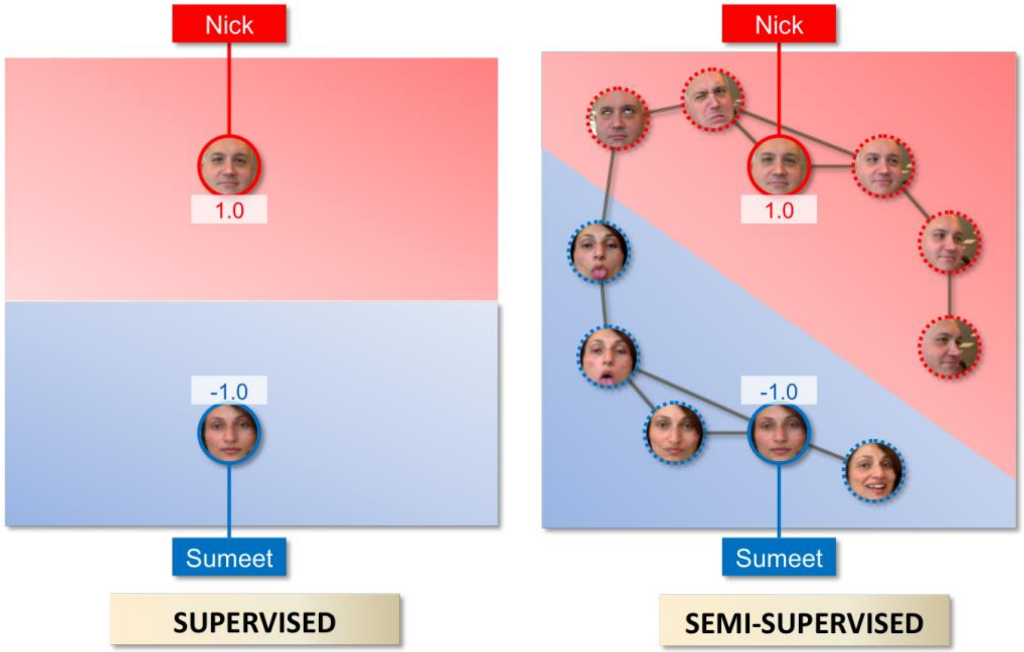
Manifold Learning: Laplacian Eigenmaps



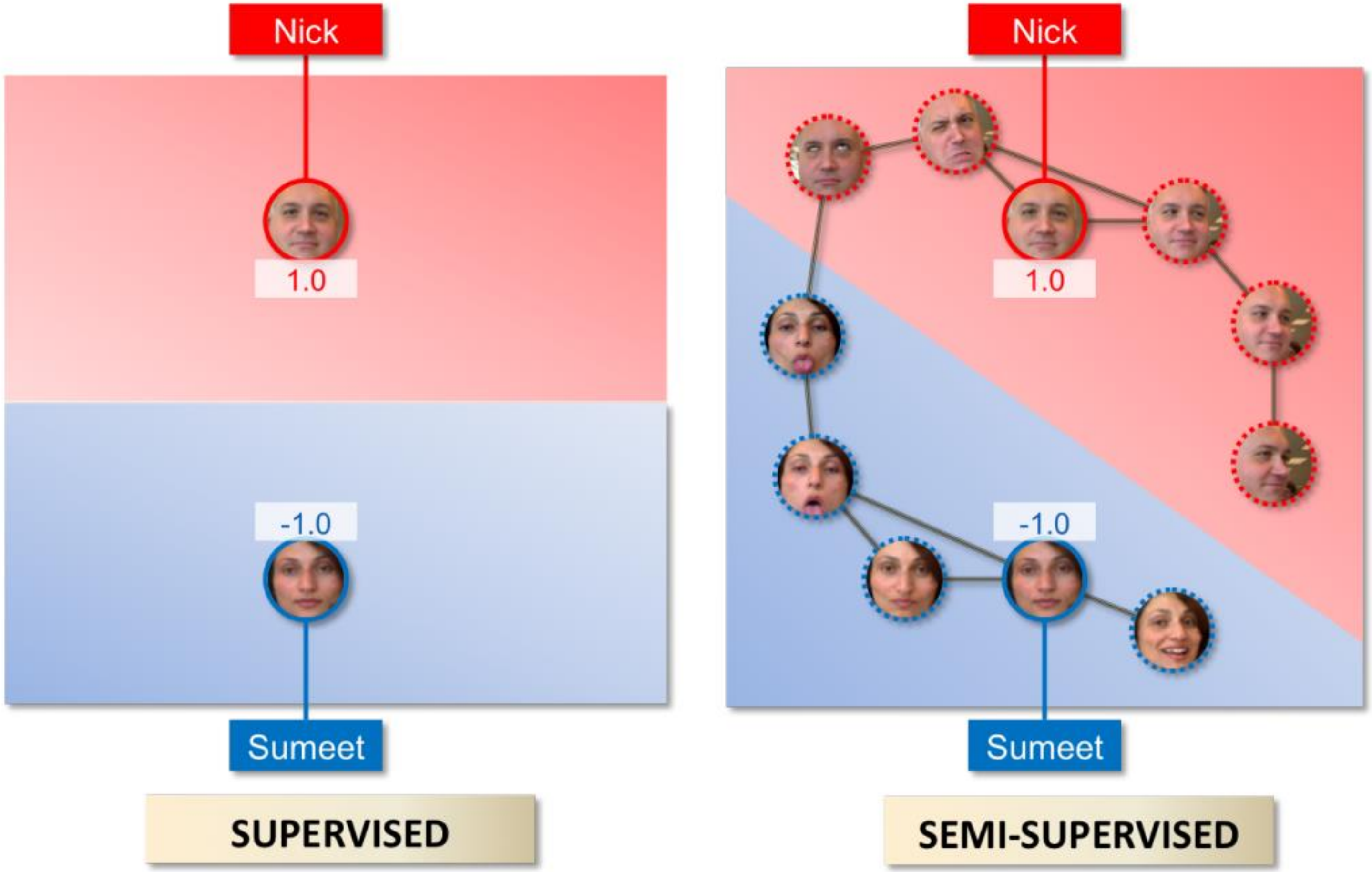
Laplacian-eigenmap-diffusion-map-manifold-learning, (Taylor, Mathworks, 2002)

SSL

semi-supervised learning



Semi-supervised learning: How is it possible?



This is how children learn! hypothesis

Semi-supervised learning (SSL)

SSL problem: definition

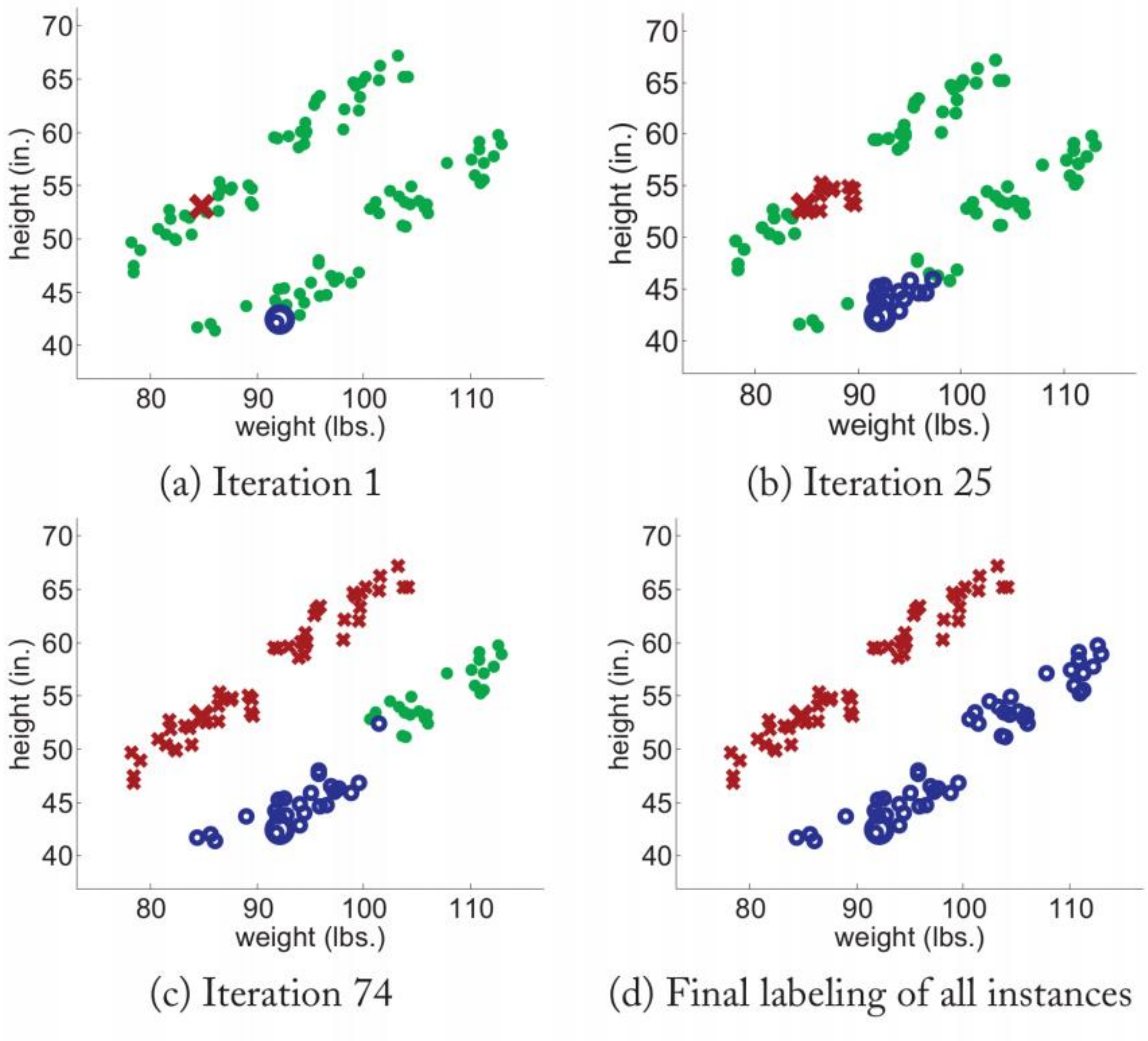
Given $\{\boldsymbol{x}_i\}_{i=1}^{N}$ from $\mathbb{R}^d$ and $\{y_i\}_{i=1}^{n}$, with $n \ll N$, find $\{y_i\}_{i=n+1}^{N}$ (**transductive**)

or find $\boldsymbol{f}$ predicting $\{y_i | y_i = \boldsymbol{f}(\boldsymbol{x}_i), i = n+1, \ldots, N\}$ well (**inductive**).

Some facts about SSL

- assumes that the unlabeled data is useful
- works with data geometry assumptions
  - cluster assumption — low-density separation
  - manifold assumption
  - smoothness assumptions, …
  - inductive or transductive/out-of-sample extension

SSL: Self-Training



(a) Iteration 1 (b) Iteration 25

(c) Iteration 74 (d) Final labeling of all instances

SSL: Self-Training

SSL: Self-Training

Input: $\mathcal{L} = \{\boldsymbol{x}_i, y_i\}_{i=1}^{n}$ and $\mathcal{U} = \{\boldsymbol{x}_i\}_{i=n+1}^{N}$

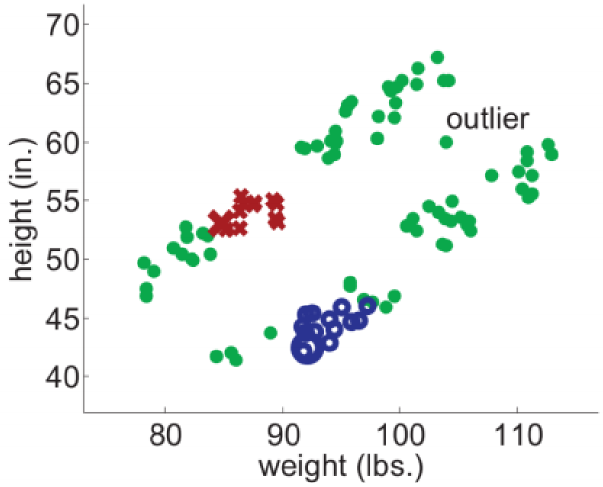
Repeat:

- train $\boldsymbol{f}$ using $\mathcal{L}$
- apply $\boldsymbol{f}$ to some of $\mathcal{U}$ and add them to $\mathcal{L}$

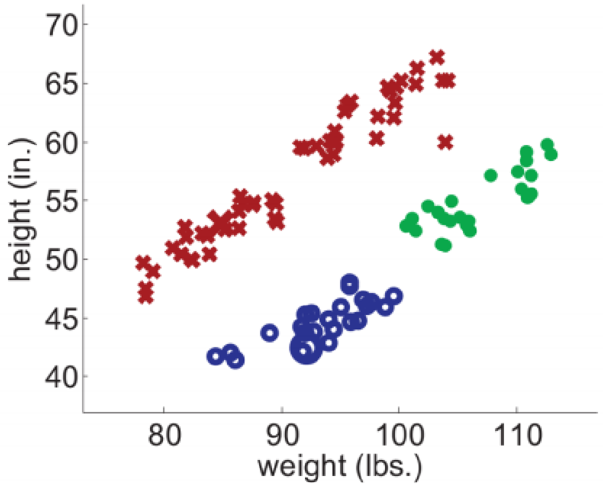
What are the properties of self-training?

- heavily depends on the classifier
- nobody uses it anymore
- errors propagate (unless the clusters are well separated)

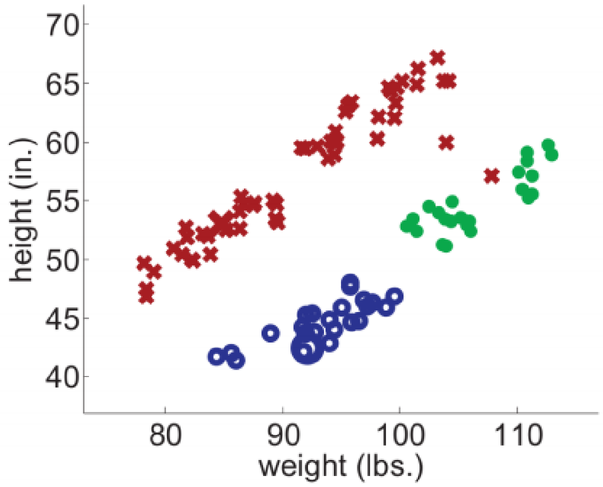
SSL: Self-Training : Bad Case



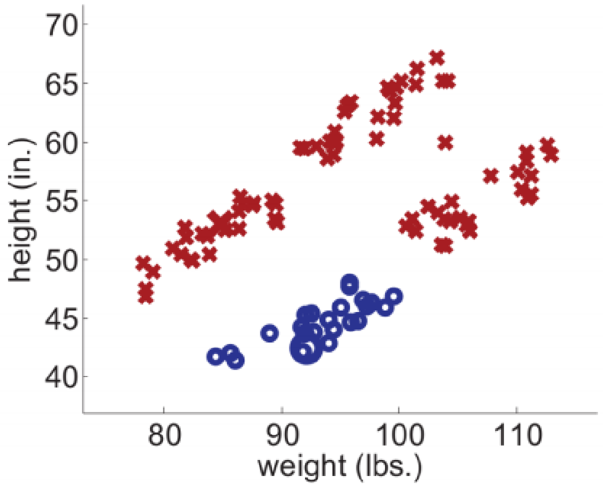
(a)



(b)

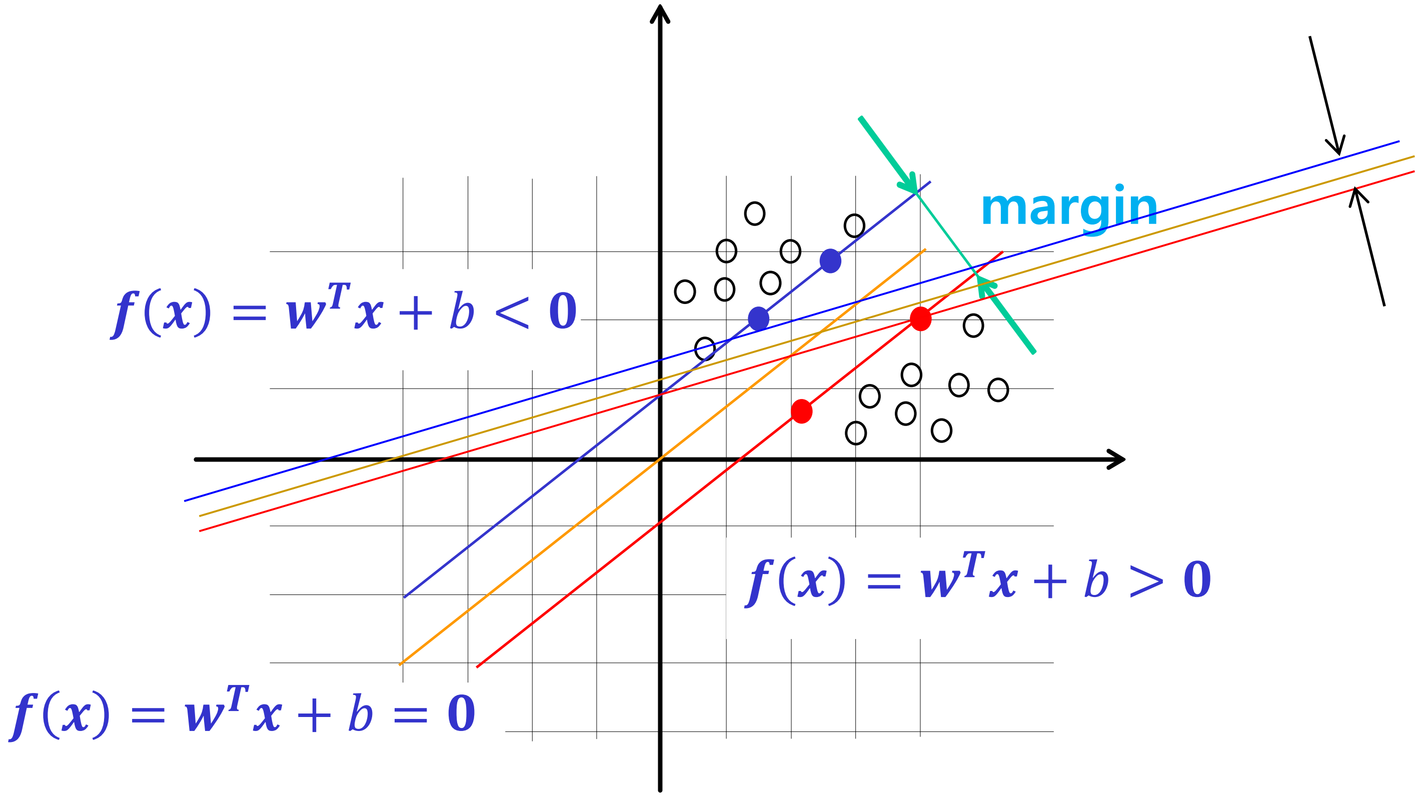


(c)



(d)

SSL: Classical SVM (Review)



Maximal Margin Hyperplane = Optimal Hyperplane

Summary questions on the lecture

- What is manifold learning?
- What is Riemannian Manifold?
- What is the role of heat kernel?
- What is cluster assumption for semi-supervised learning?
- What is manifold assumption for semi-supervised learning?