Clustering & Embedding

Peiyan Li
Multi-Label Learning
11, 10, 2017

Outline

- Motivation
- Paper Sharing
  - AnnexML: Approximate Nearest Neighbor Search for Extreme Multi-label Classification 【KDD'17】
  - Label Embedding Trees for Large Multi-class Tasks 【NIPS'10】
- Summarization

Motivation

Table 1: Dataset statistics & download

Dataset	Download	Feature Dimensionality	Label Dimensionality	Number of Train Points	Number of Test Points	Avg. Points per Label	Avg. Labels per Point	Citations
Mediamill	Download	120	101	30993	12914	1902.15	4.38	[2] + [19]
Bibtex	Download	1836	159	4880	2515	111.71	2.40	[2] + [20]
Delicious	Download	500	983	12920	3185	311.61	19.03	[2] + [21]
RCV1-2K	Download	47236	2456	623847	155962	1218.56	4.79	[2] + [26]
EURLex-4K	Download	5000	3993	15539	3809	25.73	5.31	[1] + [27]
AmazonCat-13K	Download Dataset Download Feature Vector and Label Meta-data	203882	13330	1186239	306782	448.57	5.04	[28]
AmazonCat-14K	Download Dataset Download Feature Vector and Label Meta-data	597540	14588	4398050	1099725	1330.1	3.53	[29] + [30]
Wiki10-31K	Download Dataset Download Feature Vector and Label Meta-data	101938	30938	14146	6616	8.52	18.64	[1] + [23]
Delicious-200K	Download	782585	205443	196606	100095	72.29	75.54	[1] + [24]
WikiLSHTC-325K	Download	1617899	325056	1778351	587084	17.46	3.19	[2] + [25]
Wikipedia-500K	Download Dataest Download Feature Vector and Label Meta-data	2381304	501070	1813391	783743	24.75	4.77	-
Amazon-670K	Download Dataset Download Feature Vector and Label Meta-data	135909	670091	490449	153025	3.99	5.45	[1] + [28]
Ads-1M	-	164592	1082898	3917928	1563137	7.07	1.95	[2]
Amazon-3M	Download Dataset Download Feature Vector and Label Meta-data	337067	2812281	1717899	742507	31.64	36.17	[29] + [30]
Ads-9M	-	2082698	8838461	70455530	22629136	14.32	1.79	[2]

Extreme multilabel classification

Main challenge:

1. Large scale
2. Sparse data
3. Label imbalance
4. Tail labels

Frequency	WikiLSHTC-325K		Amazon-670K	
1	79,732	(24.82%)	71,817	(10.76%)
≤ 2	112,788	(35.11%)	309,976	(46.45%)
≤ 3	137,596	(42.84%)	435,442	(65.25%)
≤ 4	157,541	(49.04%)	509,203	(76.31%)
≤ 5	174,341	(54.27%)	555,905	(83.30%)
≤ 10	226,956	(70.65%)	637,379	(95.51%)
All	321,222	(100.00%)	667,317	(100.00%)

To do what?

Design more complicated structure like tree, graph, DNN and etc.
Clustering, dimensionality reduction, sampling methods, embedding…

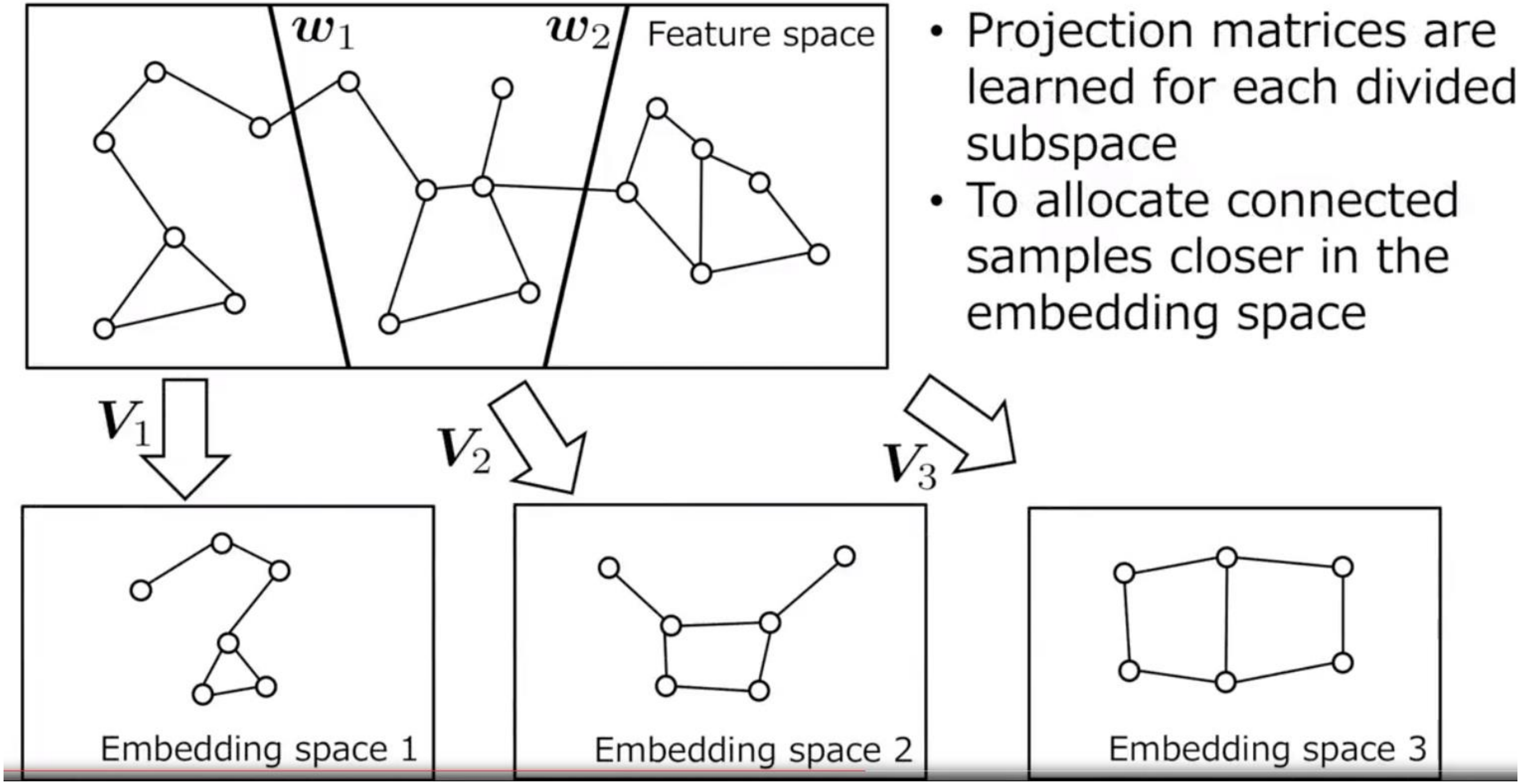
- **Basic Idea:**

reproducing the KNN graph of label vectors in the embedding space to improve both the prediction accuracy and speed of the KNN classifier.

Steps:

- ✓ Learn to partition data points
- ✓ Learn embedding
- Nearest neighbor search

Overview of AnnexML Training



- Projection matrices are learned for each divided subspace
- To allocate connected samples closer in the embedding space

Partition Data Points

Goal: learn a multi-class classifier

Construct the KNNG as weak supervision

$$\boldsymbol{N_Y^{(i)} = arg\ max_{S:S\subseteq I,|S|=n,i\notin S} \sum_{j\in S} \frac{y_i^T y_j}{|y_i||y_j|}}$$

Tips: label imbalance, inverted index

$$\max_{w_{c_i}} \sum_{j\in N_Y^{(i)}} \log\sigma\left(w_{c_i}^T x_j\right) + \sum_{k\in S^-} \log\sigma\left(-w_{c_i}^T x_k\right) - \lambda|w_c|_1$$

where $\mathrm{c}_i = \arg\max_c w_{c_i}^T x_i$ is the partition to which the i-th point belongs at this time step, $S^- \subset I$

1. to assign the approximate nearest neighbors $\boldsymbol{N_Y^{(i)}}$ to the same partition c_i to which the i-th point belongs.
2. the randomly selected points $\boldsymbol{S^-}$ should not be included in this partition
3. to make $\boldsymbol{w_c}$ sparse

Learning embeddings

Goal: reconstruct the KNNG of label vectors in the embedding space

How; preserving similarities in the original space and the embedded space.
Embedding: $\boldsymbol{z_i = V_c x_i}$, find a $\boldsymbol{V_c}$ for partition c.

$$\boldsymbol{R(x_i, y_i) \coloneqq cos(z_i, z_j) = \frac{z_i^T z_j}{||z_i||\,||z_j||} = \frac{x_i^T V_c^T V_c x_j}{||V_c x_i||||V_c x_j||}}$$

$$P(x_j|x_i) = \frac{\exp(\gamma R(x_i, x_j))}{\exp\left(\gamma R(x_i, x_j)\right) + \sum_{k \in S_c^-} \exp(\gamma R(x_i, x_k))}$$

where $S_c^- \subset I_c$ is the set of indices randomly selected from data points in the corresponding partition c.

$$\min_{V_c} \sum_{i \in I_c} \sum_{j \in N_{Y_c}^{(i)}} -\log\, P(x_j|x_i)$$

Summarization

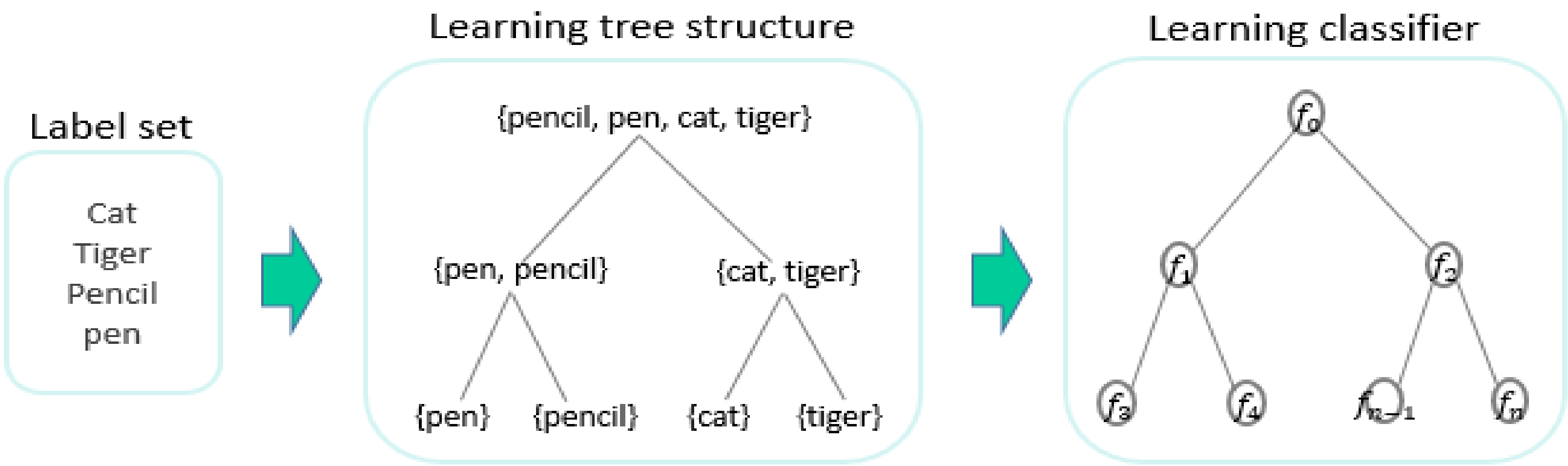
- AnnexML: Approximate Nearest Neighbor Search for Extreme Multi-label Classification
    1. As fast as tree based methods, with higher accuracy
    2. Weak supervised clustering (use k-nn as a weak supervision)
    3. Tail labels and core labels, which are more important?

- Limitations
    1. Too complex ??

Label Embedding Trees for Large Multi-class Tasks

- **Introduction**
- **Each node has**
  - **Label set**
  - **Classifier of children**
- **Label Predictors:** $F = \{f_1, f_2, \cdots, f_n\}$
- **Label sets:** $L = \{l_0, l_1, \cdots, l_n\}$

How to split the label set(construction)?
How to learn classifier(optimization)?



Learning Label Tree Structure

Basic idea: group together labels into the same label set that are likely to be confused at test time.

Algorithm 2 Learning the Label Tree Structure

Train k One-vs-Rest classifiers $\bar{f}_1, \ldots, \bar{f}_k$ independently (no tree structure is used).
Compute the confusion matrix $\bar{C}_{ij} = |\{(x, y_i) \in \mathcal{V} : \text{argmax}_r \bar{f}_r(x) = j\}|$ on validation set $\mathcal{V}$.
For each internal node l of the tree, from root to leaf, partition its label set ℓ_l between its children's label sets $L_l = \{\ell_c : c \in N_l\}$, where $N_l = \{c \in N \ : \ (l, c) \in E\}$ and $\cup_{c \in N_l} \ell_c = \ell_l$, by maximizing:

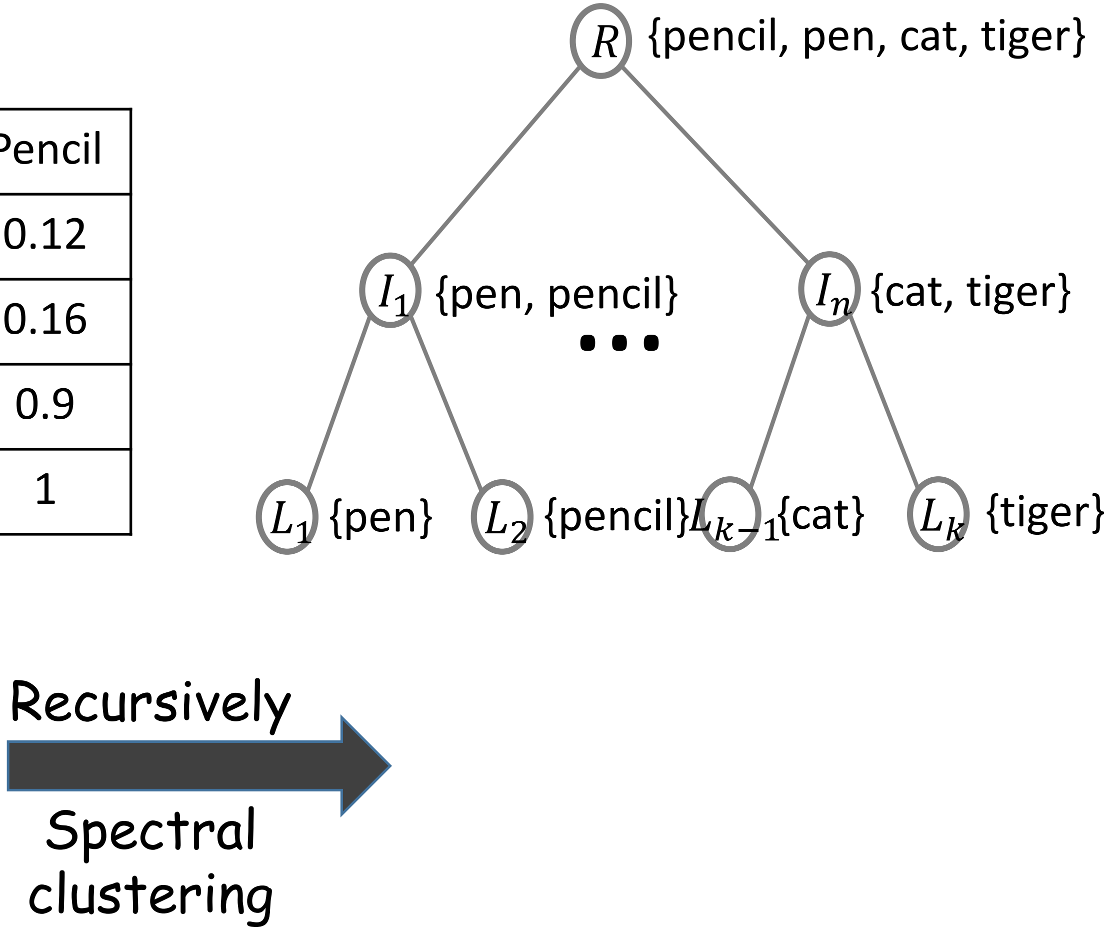
$$R_l(L_l) = \sum_{c \in N_l} \sum_{y_p, y_q \in \ell_c} A_{pq}, \quad \text{where } A = \frac{1}{2}(\bar{C} + \bar{C}^\top) \text{ is the symmetrized confusion matrix,}$$

subject to constraints preventing trivial solutions, e.g. putting all labels in one set (see [4]).

Learning Label Tree Structure

	Cat	Tiger	Pen	Pencil
Cat	1	0.6	0.1	0.12
Tiger	0.6	1	0.2	0.16
Pen	0.1	0.2	1	0.9
pencil	0.12	0.16	0.9	1

Confusion matrix



Learning Label Tree Structure

Basic form of tree loss:

$$R(f_{tree}) = I(f_{tree}(x) \neq y) dP(x,y)$$
$$= \int \max_{i \in B(x) = \{b_1(x), b_{D(x)}(x)\}} I(y \notin l_i) dP(x,y)$$

Where I is the indicator function, D is the depth in the tree of the final for prediction x

$$b_j(x) = argmax_{\{c:(b_{j-1}(x),c) \in E\}} f_c(x)$$

If there is at least one misclassification in the path, penalize it.

Learning With Fixed Label Tree

Goal: minimize the tree loss over the variables F

Given training data $\{(\boldsymbol{x_i}, \boldsymbol{y_i}),\ \boldsymbol{i} = \mathbf{1}, \cdots, \boldsymbol{m}\}$

Relaxation 1

$$R_{emp}(f_{tree}) = \frac{1}{m}\sum_{i=1}^{m} \max_{j \in B(x)} I(y_i \notin l_j) \le \frac{1}{m} \overset{\text{exmaple}}{\boxed{\sum_{i=1}^{m}}} \boxed{\sum_{j=1}^{n}} I\left(sgn\left(f_j(x_i)\right) \neq C_j(y_i)\right)$$

node

$$where\ C_j(y) = 1\ if\ y \in l_j\ and\ -1\ otherwise$$

Replace indicator function with hinge loss and $f_j(x_i) = w_i^T \phi(x)$

$$\sum_{j=1}^{n}\left(\gamma \|w_j\|^2 + \frac{1}{m}\sum_{i=1}^{m} \xi_{ij}\right) \quad s.t. \forall i, j, \begin{cases} C_j(y_j) f_j(x_j) \ge 1 - \xi_{ij} \\ \xi_{ij} \ge 0 \end{cases}$$

Learning With Fixed Label Tree

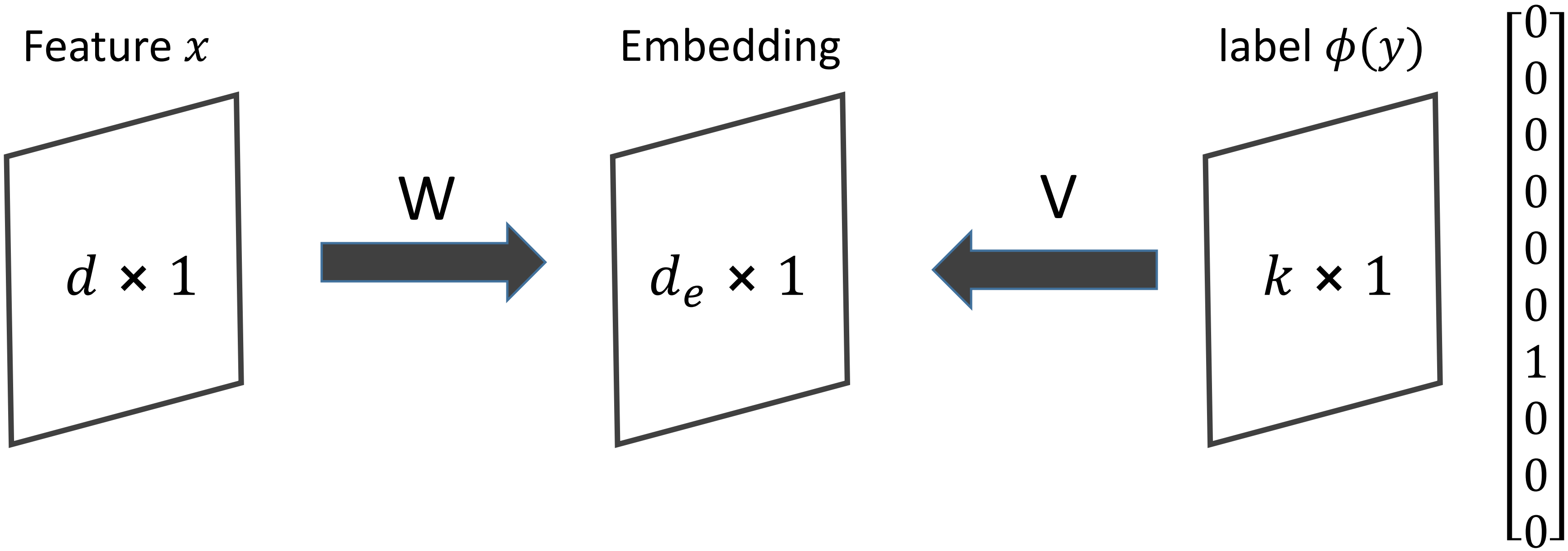
$$\sum_{j=1}^{n}\left(\gamma\|w_j\|^2 + \frac{1}{m}\sum_{i=1}^{m}\xi_{ij}\right) \quad s.t. \forall i,j, \begin{cases} C_j(y_j)f_j(x_j) \geq 1 - \xi_{ij} \\ \xi_{ij} \geq 0 \end{cases}$$

Relaxation 2

$$\gamma\sum_{j=1}^{n}\|w_j\|^2 + \frac{1}{m}\sum_{i=1}^{m}\xi_i$$

$$s.t. \begin{cases} f_r(x_i) \geq f_s(x_i) - \xi_i, \forall r,s: y_i \in l_r \wedge y_i \notin l_s \wedge (\exists p: (p,r) \in E \wedge (p,s) \in E) \\ \xi_i \geq 0 \end{cases}$$

Label Embedding



- $d_e < d$
- For dimension reduction, computation time is reduced.

Label Embedding Without Tree

Goal:

$$f_{embed} = \underset{W,V}{\operatorname{argmax}} \quad S(Wx, V\phi(y))$$

$\phi(y)$ **is a k-dimensional vector with a 1 in the y-th position and 0 otherwise.**

How to learn W, V ?

- Non-Convex Joint Optimization

$$minimize \quad \gamma \|W\|_{FRO} + \frac{1}{m}\sum_{i=1}^{m} \xi_i$$

$$s.t. \begin{pmatrix} (Wx_i)^T V\phi(i) \geq (Wx_i)^T V\phi(j) - \xi_i, \ \forall j \neq i \\ \xi_i \geq 0, \qquad i = 1, \ldots, m \\ \|V_i\| \leq 1 \end{pmatrix}$$

Label Embedding Without Tree

Goal:

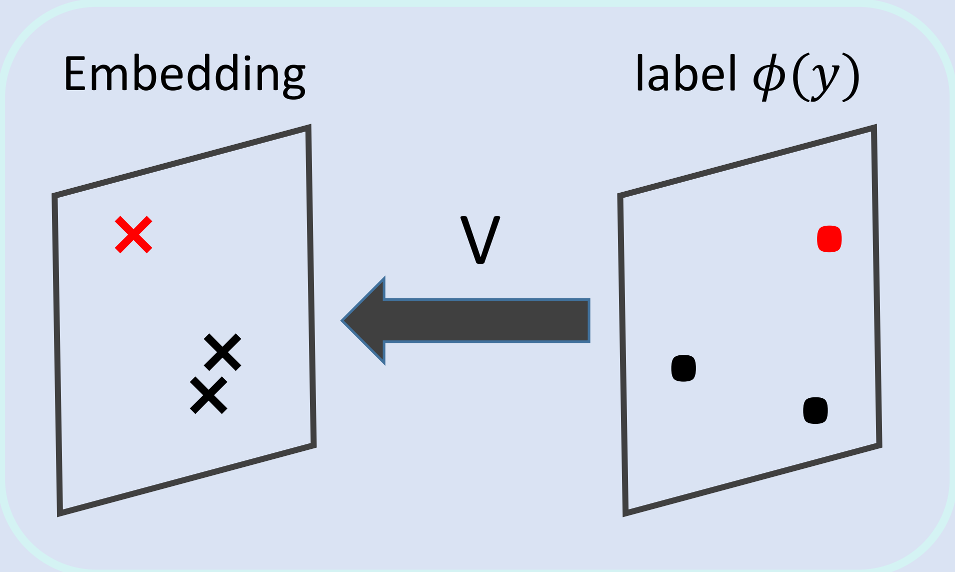
$$f_{embed} = \underset{W,V}{\operatorname{argmax}} \;\; S(Wx, V\phi(y))$$

$\phi(y)$ **is a k-dimensional vector with a 1 in the y-th position and 0 otherwise.**

How to learn W, V ?

- Sequence of Convex Problems

Learning V



Laplacian Eigenmaps

$$minimize \sum_{i,j=1}^{k} A_{ij} \left\| V_i - V_j \right\|^2$$

$$\text{s.t.} \begin{pmatrix} A = \frac{1}{2}(\bar{C} + \bar{C}^{\mathrm{T}}) \\ V^T D V = I \;\&\; D_{ii} = \sum_j A_{ij} \end{pmatrix}$$

A is the symmetrized confusion matrix. The same steps of learning a tree structure.

Label Embedding Without Tree

Goal:

$$f_{embed} = \underset{W,V}{\operatorname{argmax}} \quad S(Wx, V\phi(y))$$

$\phi(y)$ **is a k-dimensional vector with a 1 in the y-th position and 0 otherwise.**

How to learn W, V ?

- Sequence of Convex Problems

Learning W

$$minimize \quad \gamma \|W\|_{FRO} + \frac{1}{m}\sum_{i=1}^{m} \xi_i$$

$$s.t. \begin{pmatrix} \|Wx_i - V\phi(i)\|^2 \leq \|Wx_i - V\phi(j)\|^2 + \xi_i, \ \forall j \neq i \\ \xi_i \geq 0, \quad i = 1, \dots, m \end{pmatrix}$$

Learning Label Embedding Trees

Goal:

$$f_{embed} = \underset{W,V}{\operatorname{argmax}} \quad \mathrm{S}(Wx, V\phi(y))$$

$\phi(y)$ **is a k-dimensional vector with a 1 in the y-th position and 0 otherwise.**

- Sequence of Convex Problems

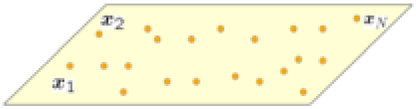
$$minimize \quad \gamma \|W\|_{FRO} + \frac{1}{m} \sum_{i=1}^{m} \xi_i$$

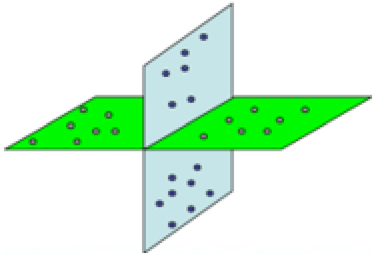
$$s.t. \begin{pmatrix} \|Wx_i - V\phi(r)\|^2 \geq \|Wx_i - V\phi(s)\|^2 - \xi_i, \\ \forall\, r, s\colon y_i \in l_r \wedge y_i \notin l_s \wedge (\exists p\colon (p, r) \in E \wedge (p, s) \in E) \\ \|V_i\| \leq 1,\ \xi_i \geq 0,\ i = 1, \dots, m \end{pmatrix}$$

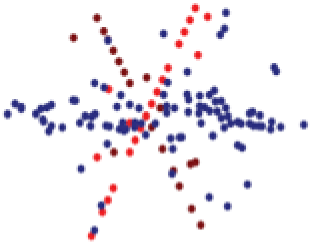
Summarization

- Label Embedding Trees for Large Multi-class Tasks 【NIPS'10】
- Limitations
  1. Learning one-vs-all classifier is costly for large-scale
  2. Disjoint partition of classes does not allow overlap
  3. Tree structure may be unbalanced
- Goal
  1. Jointly learns the splits and classifier weights
  2. Allowing overlapping of class labels among children
  3. Explicitly modeling the accuracy and efficiency trade-off
- See: Fast and Balanced: Efficient Label Tree Learning for Large Scale Object Recognition [NIPS'11]

Less is More…



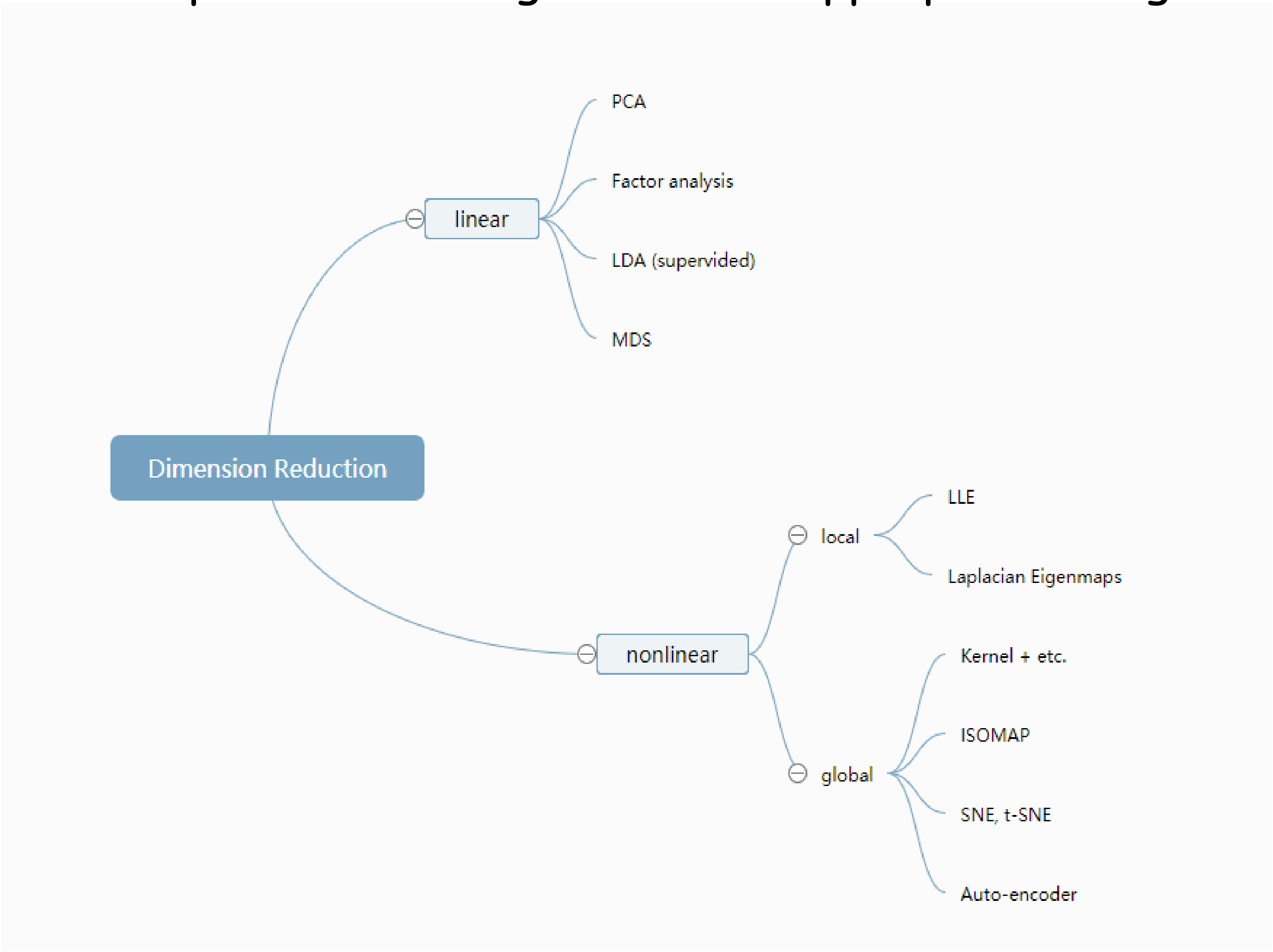




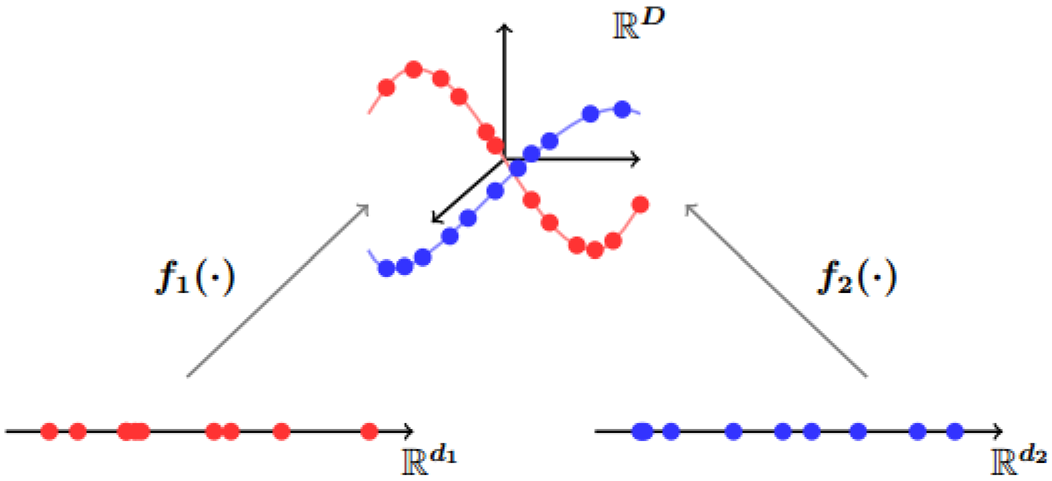
Thank you ☺

Nonlinear dimensionality reduction

Target: to find a small neighborhood around each data point and connects each point to its neighbors with appropriate weights.



Nonlinear manifolds

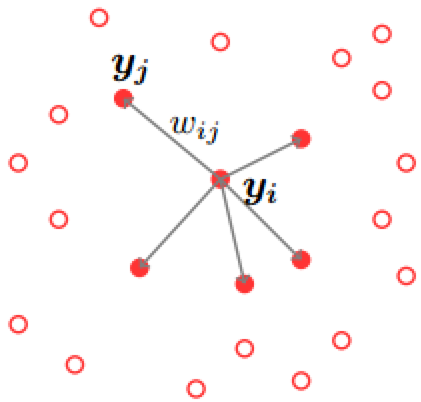


【Ehsan, SIAM12】

- Mappings are nonlinear
- Tasks:
  - Cluster data into manifolds
  - Find low-dimensional representations

Nonlinear dimensionality reduction

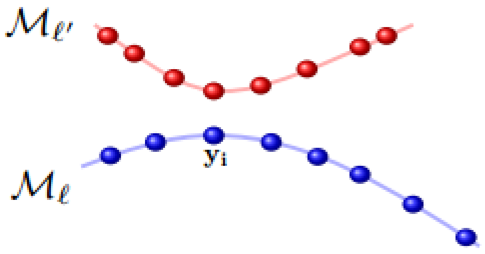
- Nonlinear dimension reduction
  1. Build nearest neighbor graph
  2. Learn weights
  3. Find embedding from weights



- LLE [Roweis, Science'00], LE [Belkin, NIPS'02], ISOMAP [Tenenbaum, Science'00], SNE [Hinton, NIPS'03], T-SNE [Maaten, JMLR'08]
  - Same in the first step
  - Different in the second step

Sparse manifold clustering and embedding

- Method (SMCE)
  1. Learn the neighborhood graph and its weights
  1. Find embedding from weights

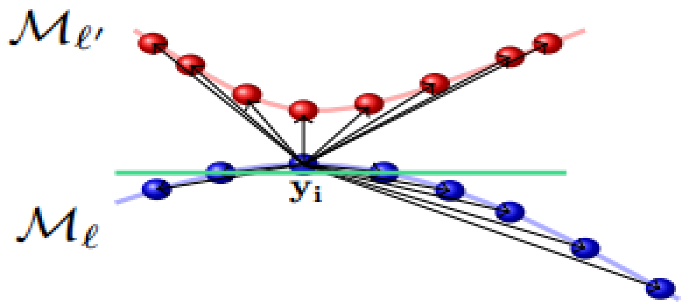


- Weights encode information for both clustering and embedding
  1. Deal with manifolds close to each other
  2. Deal with manifolds of different dimensions
     - Automatically pick the right number of neighbors

E. Elhamifar and R. Vidal, Sparse Manifold Clustering and Embedding, NIPS'11

Sparse manifold clustering and embedding

- M_l of intrinsic dimension d_l
- Affine span of $d_l + 1$ points from M_l is close to y_l



- Optimization program

$$\min \ \|q_i \odot c_i\|_1 \ \ \mathrm{s.t.} \left[\frac{y_1 - y_i}{\|y_1 - y_i\|_2} \quad \cdots \quad \frac{y_N - y_i}{\|y_N - y_i\|_2}\right] c_i \approx 0, \ \ \mathbf{1}^{\mathrm{T}} c_i = 1$$

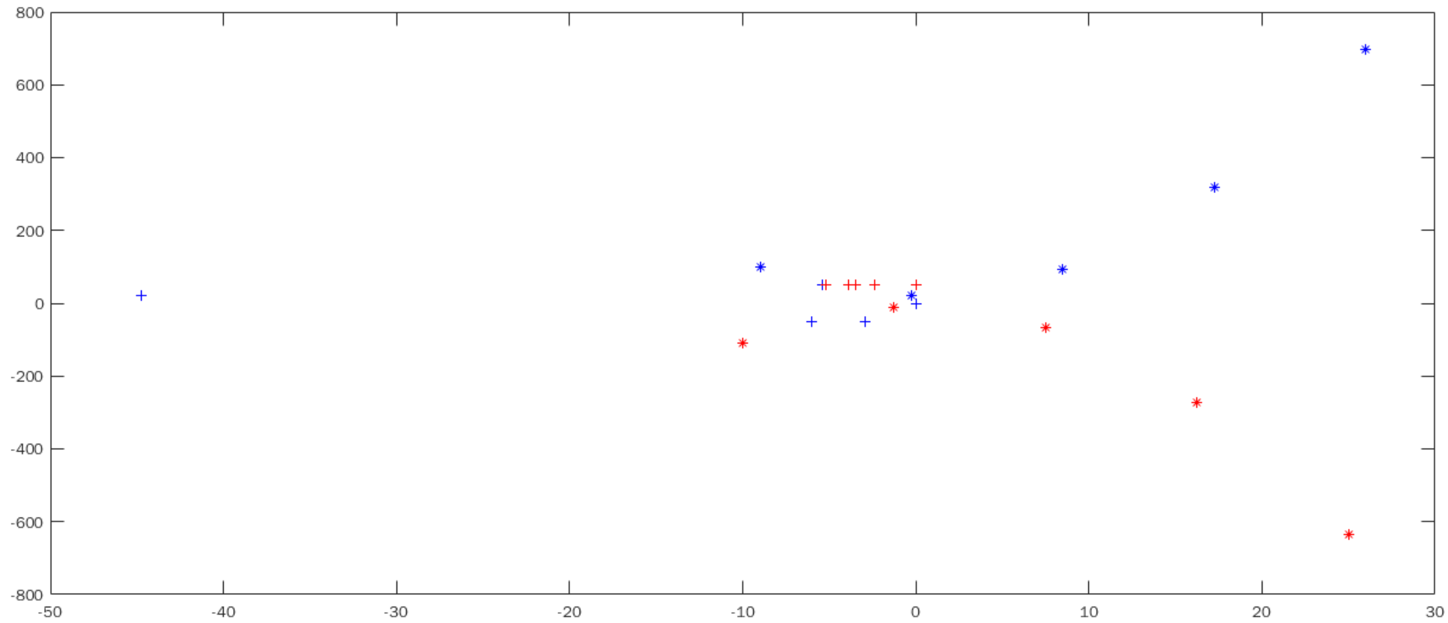
few close points　　　　span affine subspace

- Proximity inducing vector: $q_i = \left[\frac{\|y_1 - y_i\|_2}{\sum_{t \neq i} \|y_t - y_i\|_2} \quad \cdots \quad \frac{\|y_N - y_i\|_2}{\sum_{t \neq i} \|y_t - y_i\|_2}\right]^{\mathrm{T}}$

Sparse manifold clustering and embedding

$$\cdot \left[\frac{y_1 - y_i}{\|y_1 - y_i\|_2} \quad \cdots \quad \frac{y_N - y_i}{\|y_N - y_i\|_2}\right] c_i \approx 0, \ \mathbf{1}^{\mathrm{T}} c_i = 1$$

$\boxed{\text{span affine subspace}}$



1. Two manifolds marked by blue and red.
2. For the blue one on the left, calculate equation above.
3. r* -> r+, b* -> b+

Sparse manifold clustering and embedding

$$q_i = \left[\frac{\|y_1 - y_i\|_2}{\sum_{t \neq i} \|y_t - y_i\|_2} \quad \cdots \quad \frac{\|y_N - y_i\|_2}{\sum_{t \neq i} \|y_t - y_i\|_2}\right]^{\mathrm{T}}$$

$\boxed{\text{few close points}}$

Original:

$$\mathrm{Y}_i = [y_1 - y_i \quad \cdots \quad y_N - y_i]$$
$$\|\mathrm{Y}_i c_i\|_2 \leq \epsilon, and \;\; \mathbf{1}^{\mathrm{T}} c_i = 1$$

Target:

The elements of q_i should be chosen such that the points are close to y_i have smaller weights, allowing the assignment of nonzero coefficients (c_{ij}) to them.

After obtaining $C = [c_{ij}]$, we can use it to do clustering, dimensionality reduction.

Sparse manifold clustering and embedding

Exploited the self-expressiveness property of the data for

- Clustering subspaces
- Clustering and embedding of nonlinear manifolds

→ AnnexML, an extreme multilabel classification algorithm