Graph-based Techniques for Searching Large-Scale Noisy Multimedia Data

Shih-Fu Chang
Department of Electrical Engineering
Department of Computer Science

Columbia University

Joint work with Jun Wang (IBM), Tony Jebara (Columbia U), Wei Liu, Junfeng He, and Yu-Gang Jiang (Fudan U)
Graph-based Semi-Supervised Learning

• Given a small set of labeled data and a large number of unlabeled data in a high-dimensional feature space
  – Build sparse graphs with local connectivity
  – Propagate information over graphs of large data sets
  – Hopefully robust to noise and scalable to gigantic sets
Intuition

- Capture local structures via sparse graph

Through Sparse Graph Construction (e.g., kNN)
Possible Applications: Propagating Labels in Interactive Search & Auto Re-ranking

Image/Video data

Processing (denoising, cropping …)

Feature Extraction

Compute Similarity

Graph Construction

Top-rank results

Interactive browse / label

User Interface

Existing Ranking/filtering System

Applications

Search, Browsing

S.-F. Chang, Columbia U.
Example: Web Search Reranking

Google Search “Statue of Liberty”
Application: Web Search Reranking

Keyword Search

Web Images

Top images as + Bottom imgs as -

Label Diagnosis Diffusion

Rerank
Application: Web Search Reranking

Google Search “Tiger”
Application: Web Search Reranking

Keyword Search → Web Images → Label Diagnosis Diffusion

Top images as + Bottom imgs as -

Rerank

How to Handle Noisy Labels before Propagation? Scalability?
Background Review

- Given a dataset $\mathcal{X} = (\mathcal{X}_l, \mathcal{X}_u)$ of labeled samples $\mathcal{X}_l$, and unlabeled samples $\mathcal{X}_u$

- Undirected graph $g = \{\mathcal{X}, \mathcal{E}\}$ of samples $\mathcal{X}$ as vertices and edges $\mathcal{E}$ weighted by sample similarity
  
  $$w_{ij} = k(x_i, x_j)$$

- Define weight matrix $W = \{w_{ij}\}$; vertex degree $D = \text{diag} ([d_1, \cdots, d_n])$

  $$d_i = \sum_j w_{ij}$$
**Example**

- **Weight matrix**
  \[ W = \begin{bmatrix}
  0 & 1 & 0 & 0 & 0 \\
  1 & 0 & 1 & 0 & 1 \\
  0 & 1 & 0 & 2 & 0 \\
  0 & 0 & 2 & 0 & 2 \\
  0 & 1 & 0 & 2 & 0 \\
  \end{bmatrix} \]

- **Node degree**
  \[ D = \begin{bmatrix}
  1 & 0 & 0 & 0 & 0 \\
  0 & 3 & 0 & 0 & 0 \\
  0 & 0 & 3 & 0 & 0 \\
  0 & 0 & 0 & 4 & 0 \\
  0 & 0 & 0 & 0 & 3 \\
  \end{bmatrix} \]

**Label matrix**

\[ Y = \begin{bmatrix}
  1 & 0 & 0 \\
  0 & 1 & 0 \\
  0 & 0 & 1 \\
  0 & 0 & 1 \\
 0 & ? & ? \\
\end{bmatrix} \]

**Graph-based SSL**

\[ G = \{ \mathcal{X}, \mathcal{E} \} \]
\[ \{ W, D, Y \} \]

**Label prediction**

\[ F = \begin{bmatrix}
  1 & 0 & 0 \\
  0 & 1 & 0 \\
  0 & 0 & 1 \\
  0.1 & 0.2 & 0.9 \\
\end{bmatrix} \]
Some Options of Constructing Sparse Graph

- Distance Threshold
- K-Nearest Neighbor Graph

\[
\max_{\hat{P}} \sum_{ij} \hat{P}_{ij} W_{ij} \quad \hat{P}_{ij} = 1 \text{ if } x_i \text{ and } x_j \text{ connect}
\]
\[
s.t. \sum_j \hat{P}_{ij} = k, \hat{P}_{ii} = 0, \forall i, j \in 1, \ldots, n
\]

- B-Matched Graph

\[
\max_{P} \sum_{ij} P_{ij} W_{ij}
\]
\[
s.t. \sum_j P_{ij} = b, P_{ii} = 0, P_{ij} = P_{ji}, \forall i, j \in 1, \ldots, n
\]

(Huang and Jebara, AISTATS 2007)

(Jebara, Wang, and Chang, ICML 2009)
Several Ways of Constructing Sparse Graphs

- Distance threshold
- Rank threshold (kNN)
- B-Match

(k,b=4)

(k,b=6)
Examples of Graph Construction

(KNN) \( k = 4 \)  

(B-Matching) \( b = 4 \)
Graph Construction – Edge Weighting

- **Binary Weighting**
  \[ W = \mathcal{P} \]

- **Gaussian Kernel Weighting**
  \[ W_{ij} = \mathcal{P}_{ij} \exp \left( -\frac{d(x_i, x_j)}{2\sigma^2} \right) \]

- **Locally Linear Reconstruction Weighting**
  \[
  \min_W \sum_i \| x_i - \sum_{j=1}^n \mathcal{P}_{ij} w_{ij} x_j \|^2 \\
  \text{s.t. } \sum_j w_{ij} = 1, w_{ij} \geq 0
  \]
Measure Smoothness: Graph Laplacian

Graph Laplacian \( \Delta = D - W \), and normalized Laplacian \( L = D^{-1/2} \Delta D^{-1/2} \)

smoothness of function \( f \) over graph

\[
< f, Lf > = f^T L f \\
= \sum_{i=1}^{n} \sum_{j=1}^{n} W_{ij} \left\| \frac{f(x_i)}{\sqrt{D_{ii}}} - \frac{f(x_j)}{\sqrt{D_{jj}}} \right\|^2
\]

Multi-class \( < F, LF > = tr(F^T L F) \)
Classical Methods:

(Zhu et al ICML03, Zhou et al NIPS04, Joachim ICML03)

• Predict a graph function \( F \) via cost optimization

\[
F^* = \arg\min_F Q(F) = \arg\min_F \left\{ Q_{\text{smooth}}(F) + Q_{\text{fit}}(F) \right\}
\]

- Local and Global Consistency - \textbf{LGC} (Zhou et al, NIPS 04)

\[
\min_{F \in \mathbb{R}^{V \times C}} \text{tr}\{F^T L F + \mu (F - Y)^T (F - Y)\} \quad \rightarrow \quad F^* = (L/\mu + I)^{-1} Y = PY
\]

- Gaussian Random Fields – \textbf{GRF} (Zhu et al, ICML03)

\[
\min_{F \in \mathbb{R}^{V \times C}} \text{tr}(F^T \triangle F) \\
\text{s.t.} \quad F_l = Y_l \\
\nabla_{F_{u}} (Q) = 0
\]
Empirical Observations
(Jebara, Wang, and Chang, ICML 2009)

- Compare method-graphs-weights
- B-matching tends to outperform kNN
- B-Matching particularly good for GTAM + local linear (LLR) weight
Noisy Label and other Challenges

LGC Propagation

GRF Propagation

Unbalanced Labels

Ill Label Locations

Noisy Data and Labels
Label Unbalance - A Quick Fix

- Normalize labels within each class based on node degrees

Example:

\[ D = \begin{bmatrix}
  1 & 0 & 0 & 0 & 0 & 0 \\
  0 & 3 & 0 & 0 & 0 & 0 \\
  0 & 0 & 3 & 0 & 0 & 0 \\
  0 & 0 & 0 & 4 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 & 4 \\
\end{bmatrix}, \quad Y = \begin{bmatrix}
  1 & 0 & 0 \\
  0 & 1 & 0 \\
  1 & 0 & 0 \\
  0 & 0 & 1 \\
  0 & 0 & 0 \\
\end{bmatrix}, \quad V = \begin{bmatrix}
  \frac{1}{1+3} & 0 & 0 & 0 & 0 & 0 \\
  0 & \frac{3}{3} & 0 & 0 & 0 & 0 \\
  0 & 0 & \frac{3}{1+3} & 0 & 0 & 0 \\
  0 & 0 & 0 & \frac{4}{4} & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix} \]
Dealing with Noisy Labels

-- Graph Transduction via Alternate Minimization


- Change \textit{uni-variate} optimization to \textit{bi-variate} formulation:

\[
\min_{\mathbf{F} \in \mathbb{R}^{V \times c}} \text{tr}\{\mathbf{F}^T \mathbf{L} \mathbf{F} + \mu (\mathbf{F} - \mathbf{Y})^T (\mathbf{F} - \mathbf{Y})\}
\]

\[
\min_{\mathbf{F}, \mathbf{Y}} \frac{1}{2} \text{tr}\{\mathbf{F}^T \mathbf{L} \mathbf{F} + \mu (\mathbf{F} - \mathbf{VY})^T (\mathbf{F} - \mathbf{VY})\}
\]

\[s.t. \quad Y_{ij} \in \{0, 1\}, \sum_j Y_{ij} = 1\]
Alternate Optimization

- First, given $Y$ solve continuous valued $F$

$$\frac{\partial Q}{\partial F^*} = 0 \Rightarrow F^* = (L/\mu + I)^{-1}VY = PVY \quad P = (L/\mu + I)^{-1}$$

- Then, search optimal integer $Y$ given $F^*$

$$Q(Y) = \frac{1}{2} \text{tr} \left( Y^TV^T \left[ P^TLP + \mu(P^T - I)(P - I) \right] VY \right)$$

Gradient decent search
Alternate Minimization for Label Tuning

Example: 
\[
Y^T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}
\]

\[
V_Y Q = \begin{bmatrix} 0.8 & 0.1 \\ -0.23 & -0.25 \\ -0.31 & 0.07 \\ -0.17 & -0.04 \end{bmatrix}
\]

Add label: \((i^+, j^+) = \min_{i,j} \nabla_{(vy_u)} Q; y_{i^+j^+} = 1\)

Delete label: \((i^-, j^-) = \max_{i,j} \nabla_{(vy_i)} Q; y_{i^-j^-} = 0\)

- add label (3,1) 
- delete label (1,1)

Iteratively repeat the above procedure
Example – Toy Data

Consider adding label only

Label propagation by GTAM

Convergence procedure
(non-monotonic due to discrete step size)
Label Diagnosis and Self Tuning

(*LDST, Wang, Jian, & Chang, CVPR, 2009*)

Add label: \((i^+, j^+) = \min_{i,j} \nabla_{(v y_u)} Q; \quad y_{i+j} = 1\)

Delete label: \((i^-, j^-) = \max_{i,j} \nabla_{(v y_l)} Q; \quad y_{i-j} = 0\)

Decline of the cost function \(Q\) over iterations (with vs. without label tuning)
Application: Web Search Reranking

Google Search “Tiger”

Keyword Search

Web Images

Top images as +
Bottom imgs as -

Label Diagnosis Diffusion
Application: Web Search Reranking

Keyword Search

Web Images

Top images as +
Bottom imgs as -

Label Diagnosis Diffusion

Rerank
Figure 4. Example images of text search results from flickr.com. A total of nine text queries are used: dog, tiger, panda, bird, flower, airplane, forbidden city, statue of liberty, golden bridge.
Effects of Graph-based reranking

The comparison of the precision of the top 100 ranked images over different categories of images.

<table>
<thead>
<tr>
<th>Method</th>
<th>Text</th>
<th>SVM</th>
<th>LGC</th>
<th>GTAM</th>
<th>VisualRank</th>
<th>LDST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy (%)</td>
<td>67.11</td>
<td>74.22</td>
<td>79.89</td>
<td>81.44</td>
<td>79.00</td>
<td>87.56</td>
</tr>
</tbody>
</table>

The accuracy of the top ranked *Flickr* images by different approaches.

VisualRank: Jing & Baluja, 08
Possible Applications: Propagating Labels in Interactive Search & Auto Re-ranking

Image/Video data → Processing (denoising, cropping …) → Feature Extraction → Compute Similarity → Graph Construction → Label Propagation

Interactive browse / label → User Interface → No predefined Category

Interactive Mode

Search, Browsing

S.-F. Chang, Columbia U.
Application: Brain Machine Interface for Image Retrieval
-- denoise unreliable labels from brain signal decoding
(joint work with Sajda et al, ACMMM 2009, J. of Neural Engineering, May 2011)

Use EEG brain signals to detect target of interest

Use image graph to tune & propagate information

Rapid Serial Presentation of Caltech 101 Images

Graph-Based Visual Pattern Discovery
The Paradigm

Database (any target that may interest users)
The Paradigm

Database

Neural (EEG) decoder

EEG-scores
The Paradigm

Database

Neural (EEG) decoder

Exemplar labels (noisy)

Graph-based Semi-Supervised Learning

image features

prediction score
The Paradigm

Pre-triage

Post-triage
The Paradigm

Human inspects only a small sample set via BCI

Machine filters out noise and retrieves targets from very large DB

- General: no predefined target models, no keyword
- High Throughput: neuro-vision as bootstrap of fast computer vision

Pre-triage

Post-triage
The Neural Signatures of “Recognition”

D. Linden, Neuroscientist, 2005, the Oddball Effect

- Standard
- Target
- Novel

Novel (P3a)
- 442 - 444 ms

Target (P3b)
- 474 - 476 ms

Graphs showing EEG waveforms at Fz, Cz, and Pz locations.
Effect of graph-based reranking (BCI test)

Top (noisy) results of Brain EEG signal detection

Top results after graph-based label denoising & propagation

P-R curve significantly improved
More Example Results

Top 20 results of EEG detection

Top 20 results of Hybrid System (BCI-VPM)

Top 20 results of EEG detection

Top 20 results of Hybrid System (BCI-VPM)
Graph over million points and more

- k-NN graph construction + label prediction

\[ G(V, E, W), \quad L = D - W \]
\[ \hat{L} = D^{-1/2}LD^{-1/2} \]
\[ \text{time} = O(kn^2) \]

- infeasible for large-scale tasks

- Idea: AnchorGraph Regularization

complexity: \( O(m^2n) \)

# anchors \( m \ll n \)

(W. Liu, J. He, S.-F. Chang, ICML2010)
Active topic in research

• Large-scale spectral analysis (Fergus et al, ‘09)
  – Approximate solutions as linear combinations of a small number of eigenfunctions of graph Laplacian
  – Elegant solutions with linear complexity
  – But only applicable to ideal data distributions (separable uniform or Gaussian)

• Matrix approximation via Nyström (Zhang et al, ‘09)

\[
W = W_{nm} W_{mm}^{-1} W_{nm}^T
\]

  – Complexity \( O(dmn) \)
  – But may not be positive semidefinite -> non-convex
Idea: Build low-rank graph via anchors

(Liu, He, Chang, ICML10)

- Use anchor points to “abstract” the graph structure
- Compute data-to-anchor similarity: sparse local embedding
  \[ Z \in \mathbb{R}^{n \times m}, m \ll n \]
- Data-to-data similarity \( W = \text{inner product in the embedded space} \)

\[
W_{ij} = \sum_{k=1}^{m} Z_{ik} Z_{jk} = Z_i Z_j^T
\]

\[
\begin{align*}
\min_{Z_{ik}} & \|x_i - \sum_{k \in \langle i \rangle} Z_{ik} u_k\|^2 & \text{s.t.} & \sum_{k \in \langle i \rangle} Z_{ik} = 1, Z_{ik} \geq 0
\end{align*}
\]
Probabilistic Intuition

- Affinity between samples $i$ and $j$, $W_{ij}$
  - $W_{ij} =$ probability of two-step Markov random walk

\[ \mathcal{A} = \mathcal{P} \mathcal{D} \mathcal{P}^{-1} \text{, where } \mathcal{P} = \text{diag}(1^{T}Z), \, m << n \]

AnchorGraph: sparse, positive semi-definite
AnchorGraph Regression

- Apply the same sparse embedding principle to labels

\[ f(x_i) = \sum_{k=1}^{m} Z_{ik} f(u_k), \quad \text{(multi-class)} F = Z A \]

- The whole graph regularization process becomes low-rank

\[
\min_{A \in \mathbb{R}^{m \times c}} \left\| Z_l A - Y_l \right\|_F^2 + \gamma \text{tr} \left( A^T Z^T (I - Z \Lambda^{-1} Z^T) Z A \right)
\]

\[
A^* = \left( Z_l^T Z_l + \gamma Z^T Z - \gamma (Z^T Z) \Lambda^{-1} (Z^T Z) \right)^{-1} Z_l^T Y_l
\]

Small matrix inversion \( O(m^2 n) \)

\[
F^*_{n \times c} = Z_{n \times m} A^*_{n \times c}
\]

Predicted function over graph = embedding matrix \( \cdot \) inferred labels on anchors
Intuition: Anchor Graph SSL

Use low-rank ARG to infer optimal labels on anchors and samples

Predict optimal labels in the anchor space (~100 labels)

\[ O(m^2n) \]

Propagate to original sample space (~million labels)
Performance - small data set

- USPS-Train: 7,291 images of digits, 10 classes, 10 samples per class
- AGR^0: K-means anchors and naïve Z
- AGR: K-means anchors and optimized Z

<table>
<thead>
<tr>
<th>Method</th>
<th>Error Rate (%)</th>
<th>Time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1NN</td>
<td>20.15</td>
<td>0.12</td>
</tr>
<tr>
<td>LGC with 6NN graph</td>
<td>8.79</td>
<td>403.02</td>
</tr>
<tr>
<td>GFHF with 6NN graph</td>
<td>5.19</td>
<td>413.28</td>
</tr>
<tr>
<td>AGR^0</td>
<td>7.40</td>
<td>10.20</td>
</tr>
<tr>
<td>AGR</td>
<td>6.56</td>
<td>16.57</td>
</tr>
</tbody>
</table>

40x speedup

accuracy comparable to analytical optimum
Large Data Set Evaluation

- 630,000 MNIST images over 10 classes, 100 labeled images only
- Conventional analytical solutions infeasible
- Among scalable solutions - reduce error rates by 30%-50%

<table>
<thead>
<tr>
<th>Method</th>
<th>Error Rate (%)</th>
<th>Training Time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1NN</td>
<td>39.65</td>
<td>5.46</td>
</tr>
<tr>
<td>Eigenfunction ('09)</td>
<td>36.94</td>
<td>44.08</td>
</tr>
<tr>
<td>PVM ('09)</td>
<td>29.37</td>
<td>266.89</td>
</tr>
<tr>
<td>AGR^0</td>
<td>24.71</td>
<td>232.37</td>
</tr>
<tr>
<td>AGR</td>
<td>19.75</td>
<td>331.72</td>
</tr>
</tbody>
</table>

30%-50% gain
Extension to Web-Scale

• Techniques described above not scalable to Web-scale or dynamic data sets
  – Cannot handle cases when $n = \sim \text{billions}$
  – For dynamic data, updating graph is expensive

• Preferred:
  learn *Inductive Models* to handle novel dynamic data
Data Subsampling & Learn Inductive Model

Web-scale database

one million data points

subsampling

novel data point $x$

$z(x)$

anchor points

predict $x$’s label

$f(x) = \tilde{z}(x)a$

anchors’ labels $a$

Anchor Graph Construction

data-to-anchor map $z(x)$

Seed labels

Anchor Graph Regularization

$x$
ARG over 80M Tiny Images + CIFAR-10

<table>
<thead>
<tr>
<th>training images</th>
<th>test images</th>
</tr>
</thead>
<tbody>
<tr>
<td>airplane</td>
<td></td>
</tr>
<tr>
<td>automobile</td>
<td></td>
</tr>
<tr>
<td>bird</td>
<td></td>
</tr>
<tr>
<td>cat</td>
<td></td>
</tr>
<tr>
<td>deer</td>
<td></td>
</tr>
<tr>
<td>dog</td>
<td></td>
</tr>
<tr>
<td>frog</td>
<td></td>
</tr>
<tr>
<td>horse</td>
<td></td>
</tr>
<tr>
<td>ship</td>
<td></td>
</tr>
<tr>
<td>truck</td>
<td></td>
</tr>
<tr>
<td>background</td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>1NN</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------</td>
</tr>
<tr>
<td></td>
<td>1K Anchors</td>
</tr>
<tr>
<td>Accuracy (%)</td>
<td>51.66 ± 0.28</td>
</tr>
<tr>
<td>Training Time (s)</td>
<td>0</td>
</tr>
<tr>
<td>Test Time (s)</td>
<td>6.29e-4</td>
</tr>
</tbody>
</table>

Learn ARG

Novel test sample

80 Million Tiny Images

1 Million samples
(1% labels from CIFAR-10)

\[
f(x) = z(x)^T \alpha
\]
Additional Issues

• Multi-edge Graph
  – Multiple relation edges between nodes

• Multi-feature Graph
  – Build graphs in multiple feature spaces
  – Joint optimization

• Label tuning vs. Active Learning
Image-Based Multi-Edge Graph

Liu et al, ACM Multimedia 2010

two images with the same tag

• one edge connecting the two regions sharing the tag, but not all
• How to propagate label over multiple edges?
Extension to Multi-Feature Graphs

How to handle noisy labels in multiple graphs?

How to handle noisy labels in multiple graphs?

Graph 1

Graph K

User Input

Label Propagation

Ranking list
Multi-graph SSL vs. single-graph

Improve performance by 20%-80%

Mean Average Precision

The number of labels

Caltech 101 data set
References and Tools


