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

Shih-Fu Chang

Department of Electrical Engineering

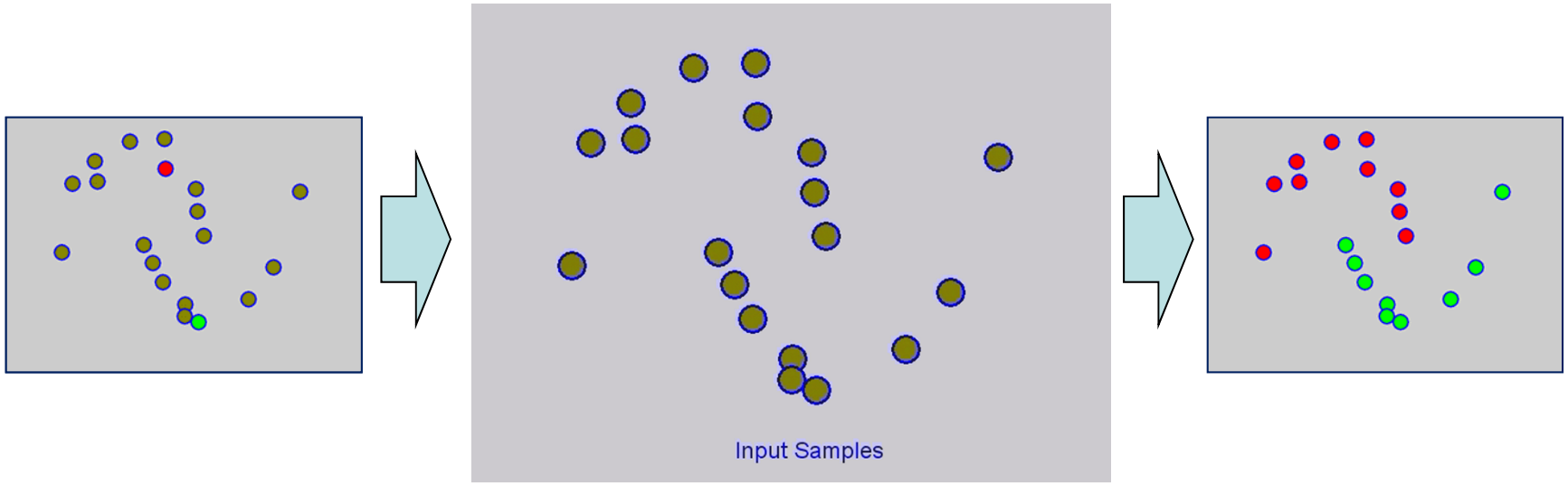
Department of Computer Science

Columbia University

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

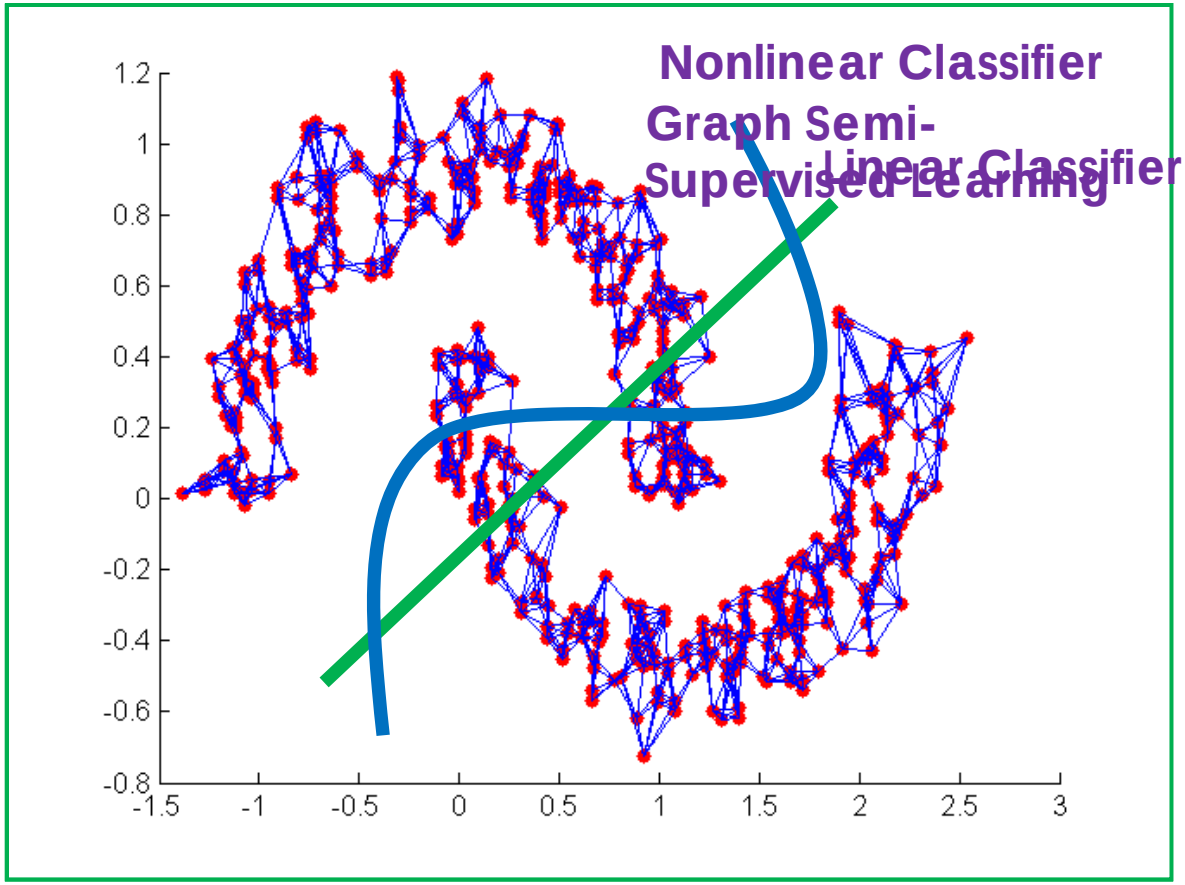
Graph-based Semi-Supervised Learning

- 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



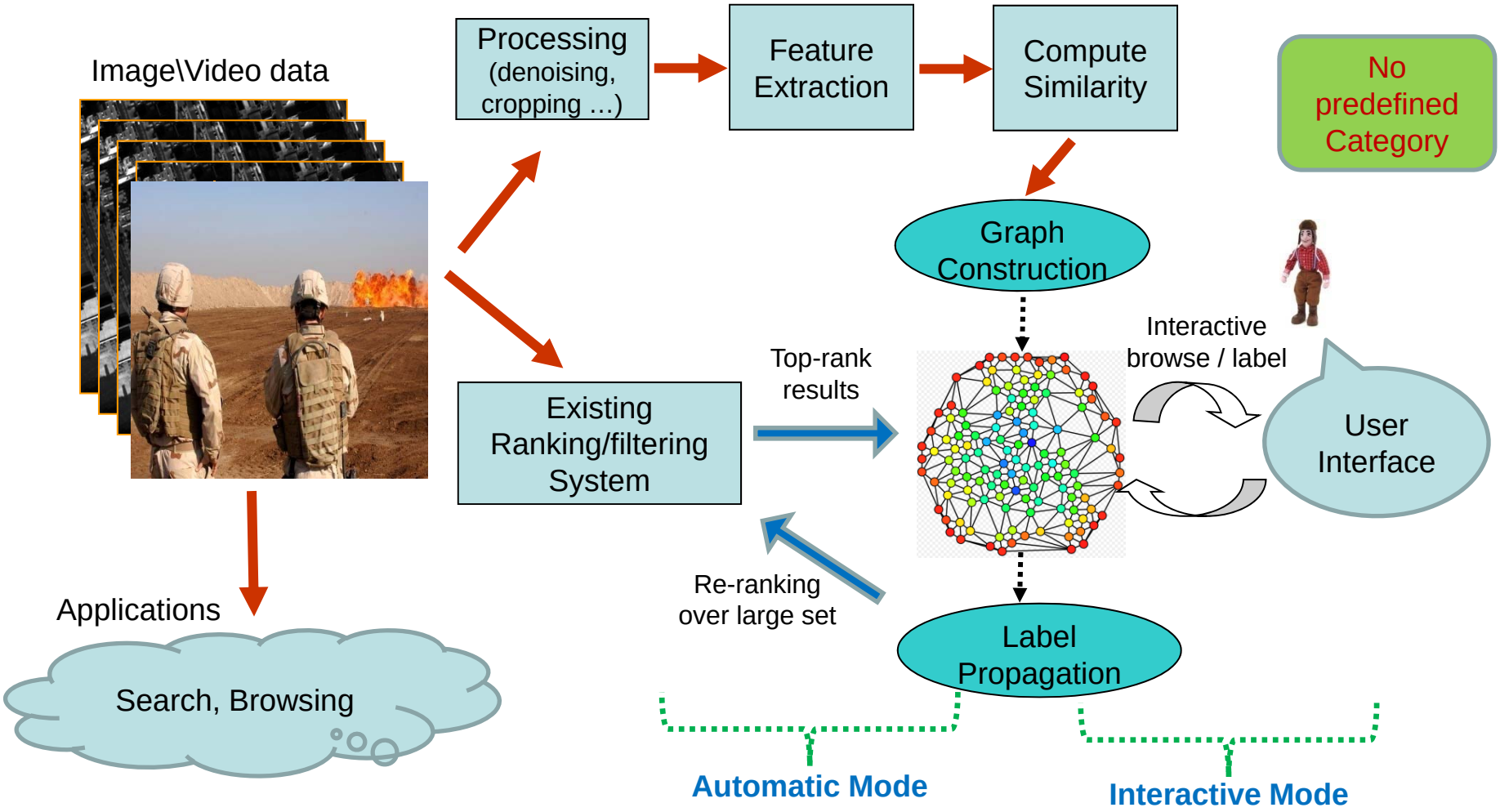
Intuition

- Capture local structures via sparse graph

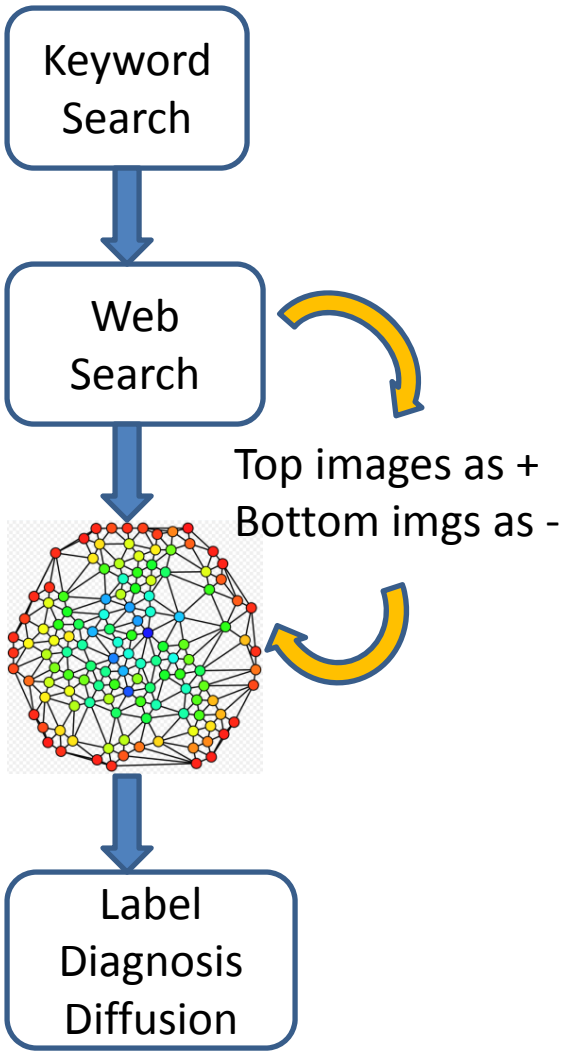


Through Spare Graph Construction (e.g., kNN)

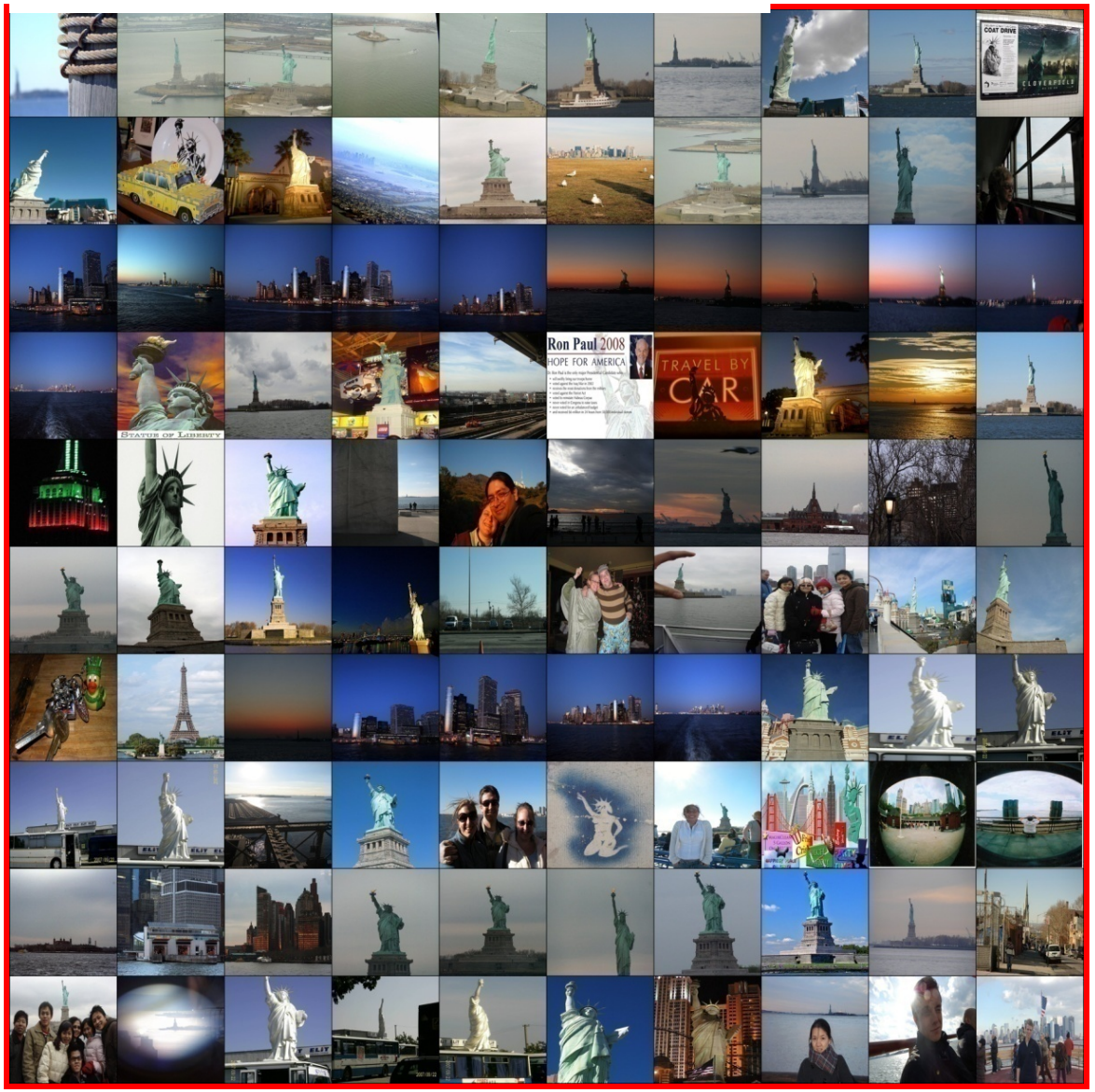
Possible Applications:
Propagating Labels in Interactive Search & Auto Re-ranking



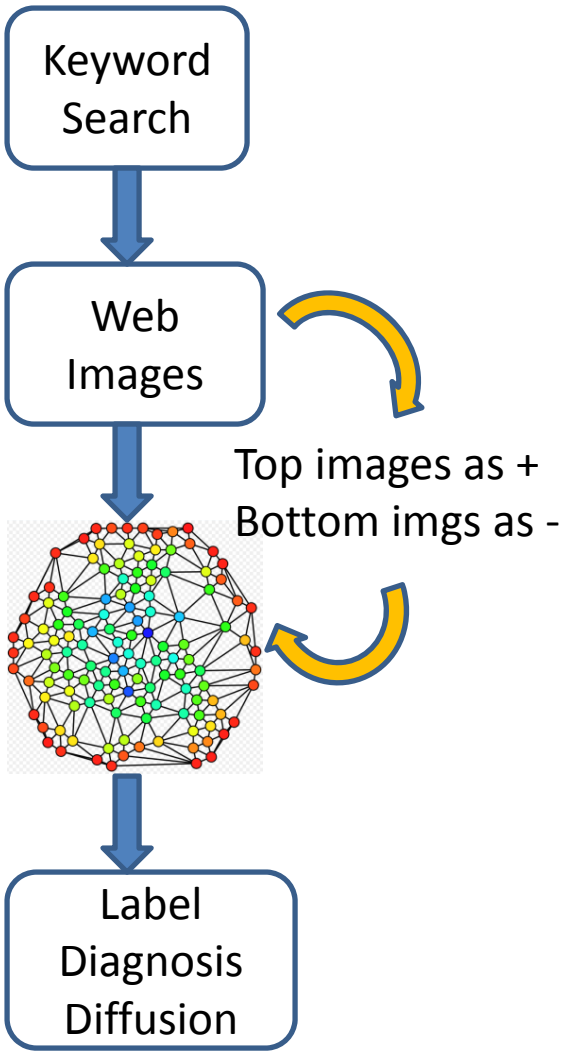
Example: Web Search Reranking



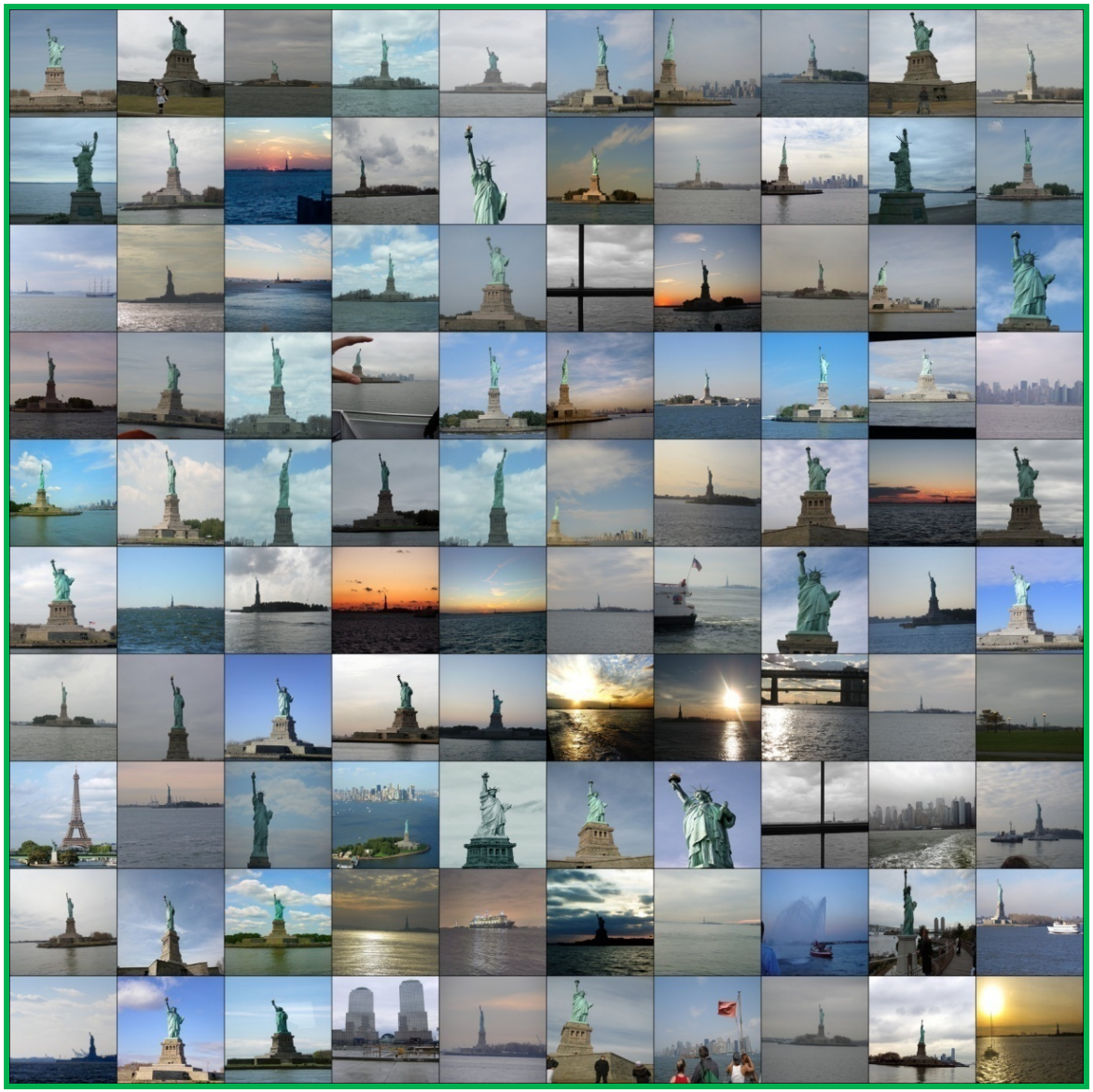
Google Search "Statue of Liberty"



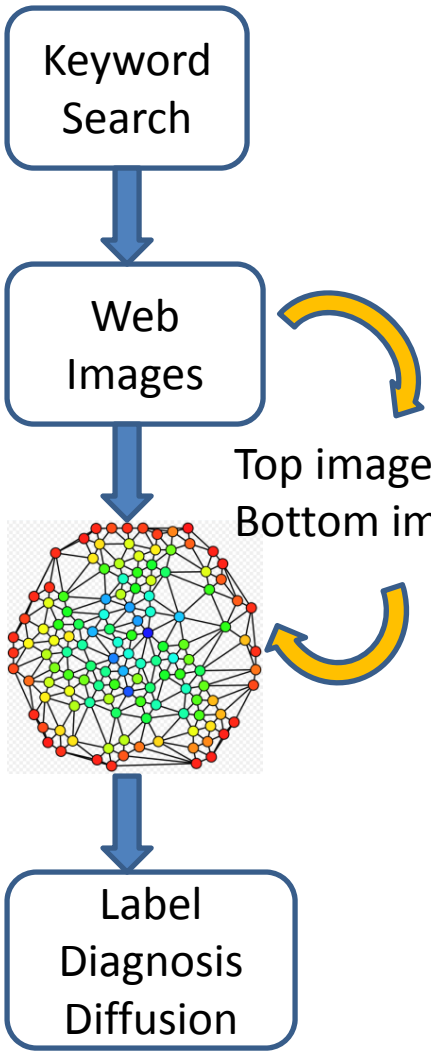
Application: Web Search Reranking



Rerank

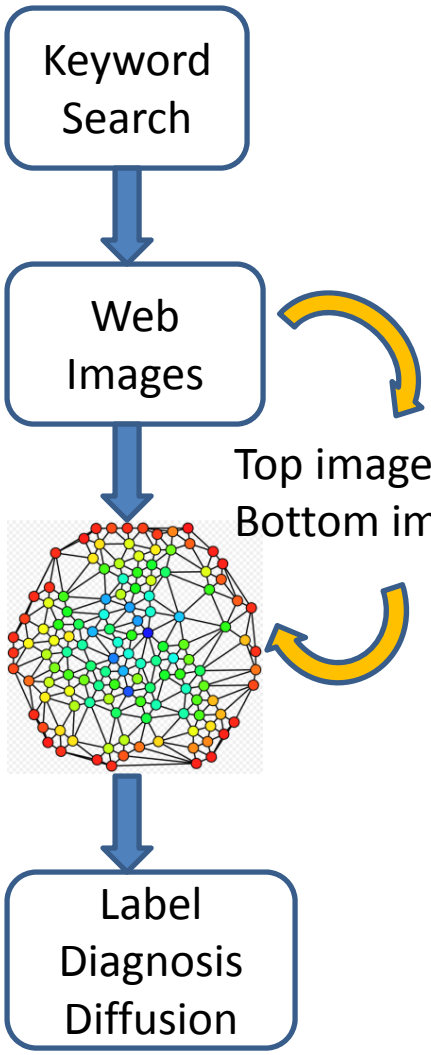


Application: Web Search Reranking



Google Search "Tiger"

Application: Web Search Reranking



Rerank

How to Handle Noisy Labels before Propagation?
Scalability?

Background Review

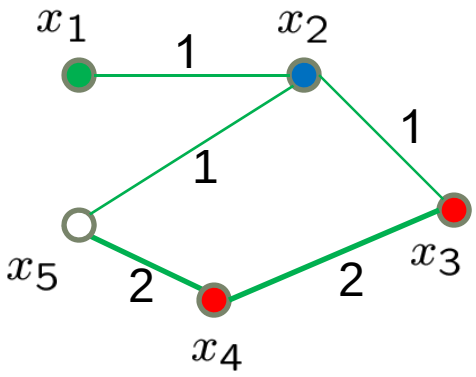
- Given a dataset $\mathcal{X} = (\mathcal{X}_l, \mathcal{X}_u)$ of labeled samples $\mathcal{X}_l$, and unlabeled samples $\mathcal{X}_u$
- *undirected* graph $\mathcal{G} = \{\mathcal{X}, \mathcal{E}\}$of samples $\mathcal{X}$as vertices and edges $\mathcal{E}$ weighted by sample similarity

$$w_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$$

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

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

Example



Weight matrix

$$\mathbf{W} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 2 & 0 \\ 0 & 0 & 2 & 0 & 2 \\ 0 & 1 & 0 & 2 & 0 \end{bmatrix}$$

Node degree

$$\mathbf{D} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 3 \end{bmatrix}$$

classes

samples

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

Label matrix

$$\mathcal{G} = \{\mathcal{X}, \mathcal{E}\}$$

$$\{\mathbf{W}, \mathbf{D}, \mathbf{Y}\}$$

Graph-based SSL

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0.1 & 0.2 & 0.9 \end{bmatrix}$$

Label prediction

Some Options of Constructing Sparse Graph

- Distance Threshold
- K-Nearest Neighbor Graph

$$\max_{\hat{P}} \sum_{ij} \hat{P}_{ij} W_{ij} \qquad \hat{P}_{ij} = 1 \; if \; x_i \; and \; x_j \; connect$$

$$s.t. \sum\nolimits_j \hat{P}_{ij} = k, \hat{P}_{ii} = 0, \; \forall i, j \; \in 1, \ldots, n$$

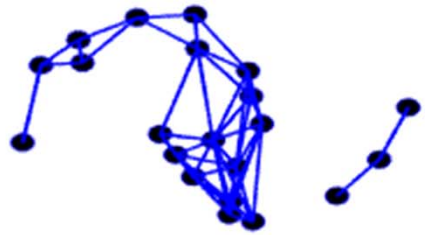
- B-Matched Graph

(Huang and Jebara, AISTATS 2007)
(Jebara, Wang, and Chang, ICML 2009)

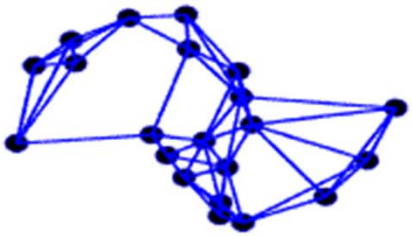
$$\max_{P} \sum_{ij} P_{ij} W_{ij}$$

$$s.t. \sum\nolimits_j P_{ij} = b, P_{ii} = 0, P_{ij} = P_{ji}, \; \forall i, j \; \in 1, \ldots, n$$

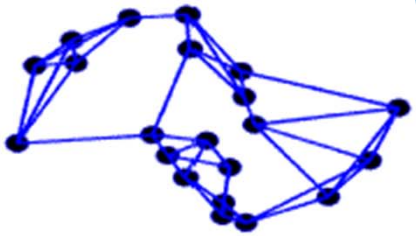
Several Ways of Constructing Sparse Graphs



(a)

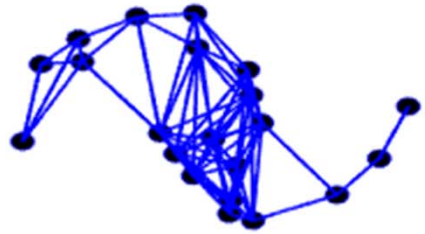


(c)

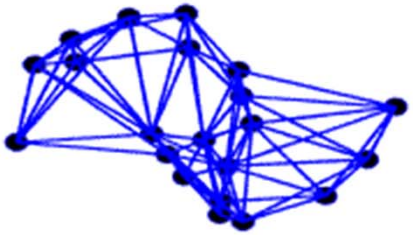


(e)

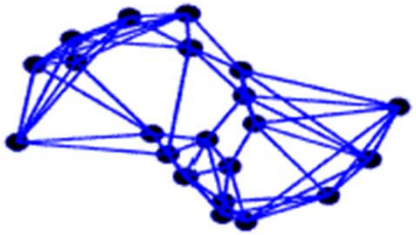
k,b=4



(b)



(d)



(f)

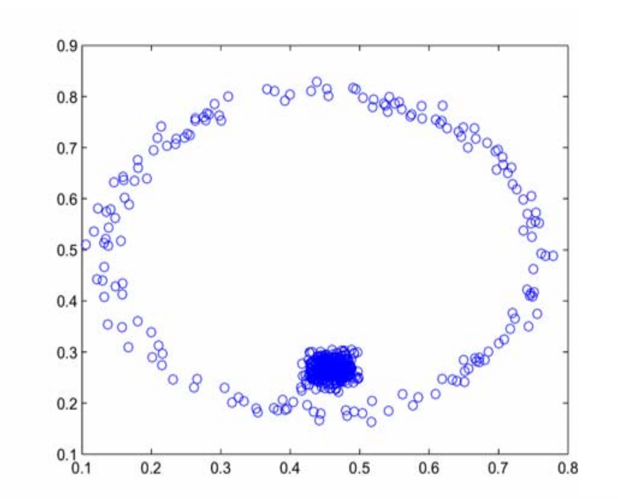
k,b=6

Distance threshold

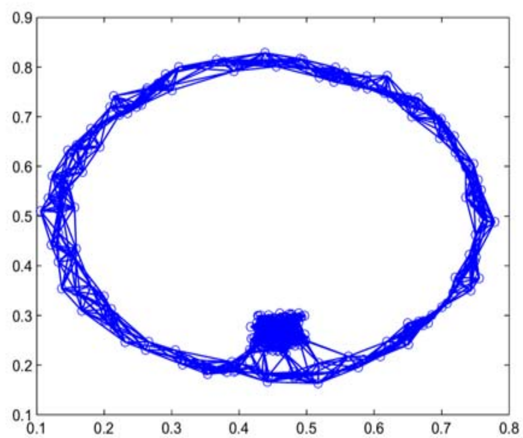
Rank threshold (kNN)

B-Match

Examples of Graph Construction

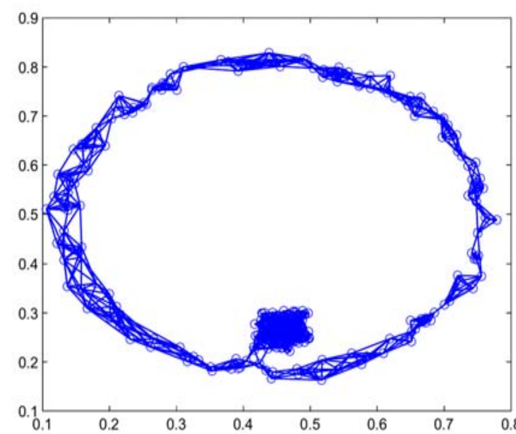


(KNN)



k = 4

(B-Matching)



b = 4

Graph Construction – Edge Weighting

- Binary Weighting

$$\mathbf{W} = \mathcal{P}$$

- Gaussian Kernel Weighting

$$\mathbf{W}_{ij} = \mathcal{P}_{ij} \exp\left(-\frac{d(\mathbf{x}_i, \mathbf{x}_j)}{2\sigma^2}\right)$$

- Locally Linear Reconstruction Weighting

$$\min_W \sum_i \|\mathbf{x}_i - \sum_{j=1}^{n} \mathcal{P}_{ij} w_{ij} \mathbf{x}_j\|^2$$

$$s.t. \sum\nolimits_j w_{ij} = 1, w_{ij} \geq 0$$

Measure Smoothness: Graph Laplacian

- Graph Laplacian $\Delta = \mathrm{D} - \mathrm{W}$, and normalized Laplacian $\mathrm{L} = \mathrm{D}^{-1/2}\Delta\mathrm{D}^{-1/2}$
- smoothness of function *f* over graph

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

Multi-class $< \mathbf{F}, \mathbf{LF} > = tr(\mathbf{F}^{\mathrm{T}}\mathbf{LF})$

Classical Methods:

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

- Predict a graph function (F) via cost optimization

prediction function — function smoothness — empirical loss

$$\mathbf{F}^* = \arg\min_{\mathbf{F}} \mathcal{Q}(\mathbf{F}) = \arg\min_{\mathbf{F}} \left\{ Q_{smooth}(\mathbf{F}) + Q_{fit}(\mathbf{F}) \right\}$$

- Local and Global Consistency - **LGC** (Zhou et al, NIPS 04)

$$\min_{\mathbf{F} \in \mathbb{R}^{|V| \times c}} \operatorname{tr}\{\mathbf{F}^\top \mathbf{L}\mathbf{F} + \mu(\mathbf{F}-\mathbf{Y})^\top(\mathbf{F}-\mathbf{Y})\} \quad \rightarrow \quad \mathbf{F}^* = (\mathbf{L}/\mu + \mathbf{I})^{-1}\mathbf{Y} = \mathbf{P}\mathbf{Y}$$

- Gaussian Random Fields – **GRF** (Zhu et al, ICML03)

$$\min_{\mathbf{F} \in \mathbb{R}^{|V| \times c}} \operatorname{tr}(\mathbf{F}^\top \triangle \mathbf{F})$$
$$s.t. \quad \mathbf{F}_l = \mathbf{Y}_l$$
$$\nabla_{F_u}(Q) = 0$$

Empirical Observations

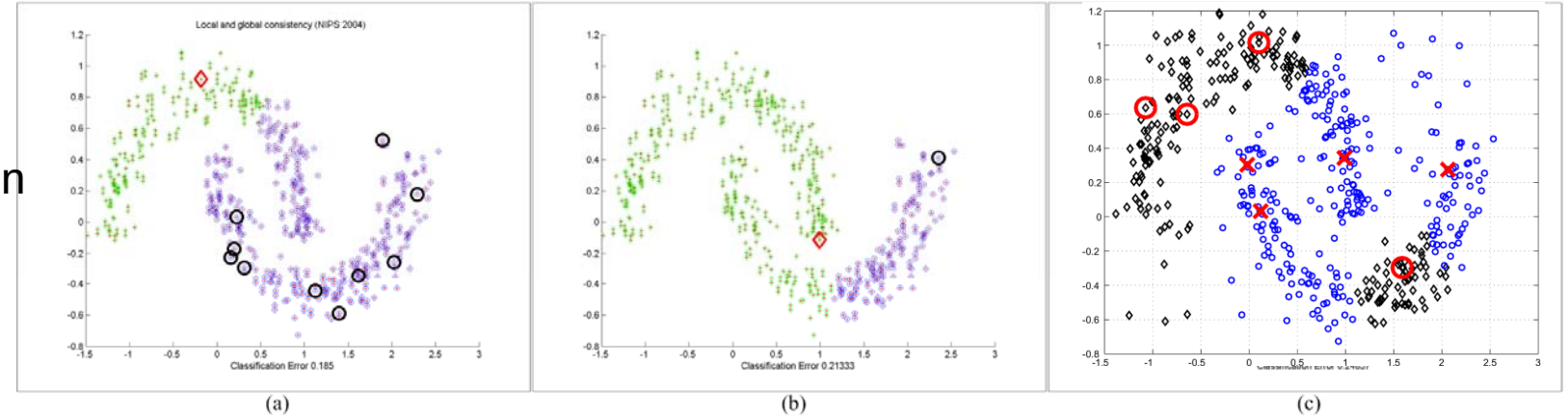
(Jebara, Wang, and Chang, ICML 2009)

- Compare **method-graphs-weights**
- B-matching tends to outperform kNN
- B-Matching particularly good for GTAM + local linear (LLR) weight

Data set	USPS	COIL	BCI	TEXT
QC + CMN	13.61	59.63	50.36	40.79
LDS	25.2	67.5	49.15	31.21
Laplacian	17.57	61.9	49.27	27.15
Laplacian RLS	18.99	54.54	48.97	33.68
CHM (normed)	20.53	-	46.9	-
LGC-KNN-BN	14.7	59.18	48.94	48.79
LGC-KNN-GK	12.42	57.3	48.42	48.09
LGC-KNN-LLR	15.8	56.75	48.65	40.28
LGC-BM-BN	14.4	59.31	48.34	40.44
LGC-BM-GR	11.89	58.17	48.17	37.39
LGC-BM-LLR	14.44	58.69	48.08	39.83
GRF-KNN-BN	19.11	64.45	48.77	47.65
GRF-KNN-GK	12.94	61.31	48.98	47.65
GRF-KNN-LLR	19.20	61.19	48.46	47.14
GRF-BM-BN	18.98	60.63	48.44	43.16
GRF-BM-GR	12.82	60.87	48.77	42.88
GRF-BM-LLR	18.95	60.84	48.25	42.94
GTAM-KNN-BN	6.42	29.70	47.56	49.36
GTAM-KNN-GK	4.77	16.69	47.29	49.13
GTAM-KNN-LLR	6.69	15.35	45.54	41.48
GTAM-BM-BN	5.2	25.83	47.92	17.81
GTAM-BM-GR	**4.31**	13.65	47.48	28.74
GTAM-BM-LLR	5.45	**12.57**	**43.73**	**16.35**

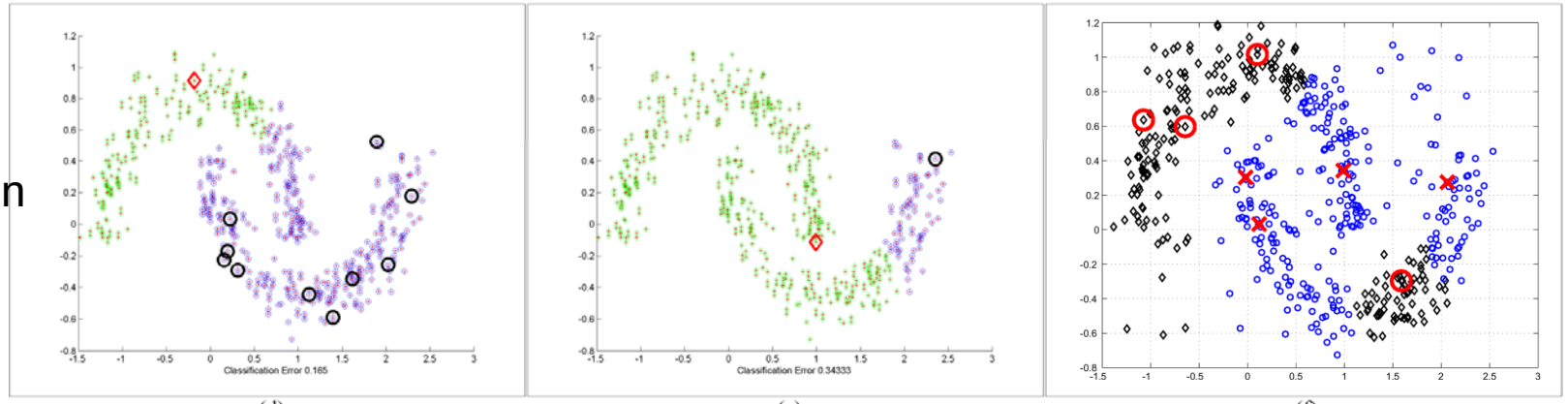
Noisy Label and other Challenges

LGC Propagation



(a) (b) (c)

GRF Propagation



(d) (e) (f)

Unbalanced Labels

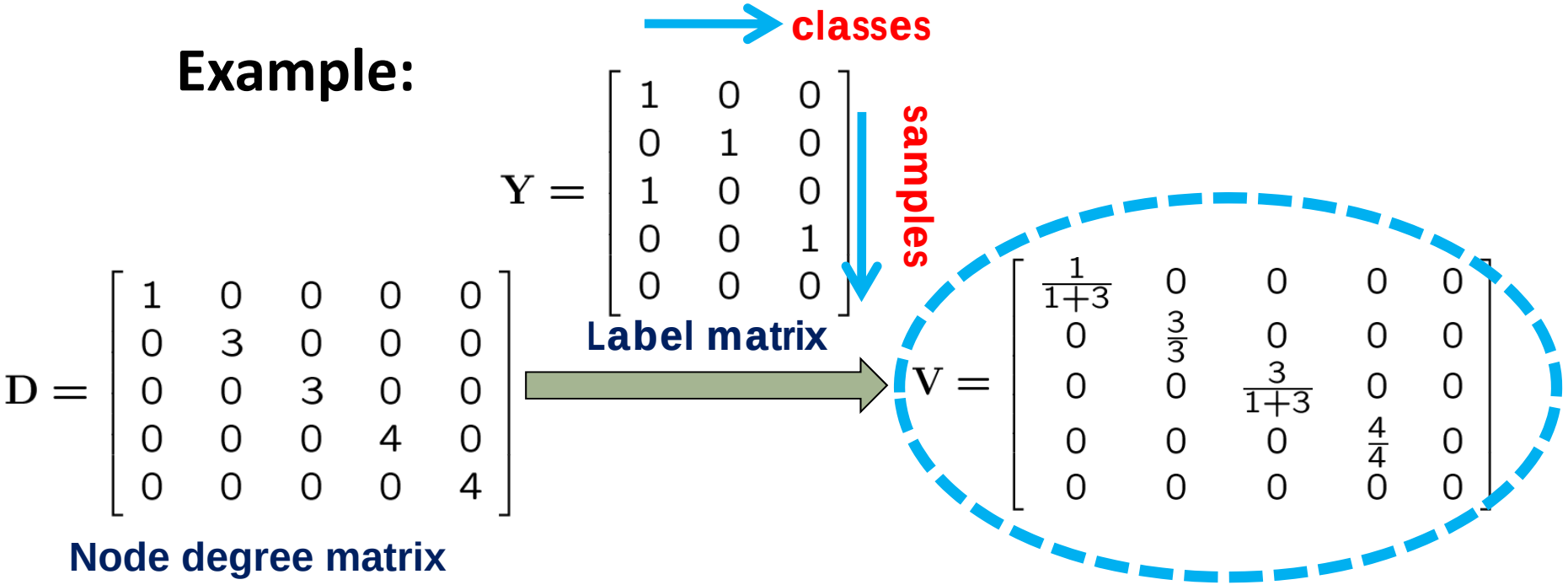
Ill Label Locations

Noisy Data and Labels

Label Unbalance - A Quick Fix

- **Normalize labels within each class based on node degrees**

Example:



Dealing with Noisy Labels

-- Graph Transduction via Alternate Minimization

*(**GTAM**, Wang, Jebara, & Chang, ICML, 2008) (**LDST**, Wang, Jiang, & Chang, CVPR, 2009)*

- Change *uni-variate* optimization to *bi-variate* formulation:

$$\min_{\mathbf{F}\in\mathbb{R}^{|V|\times c}} \operatorname{tr}\{\mathbf{F}^\top\mathbf{L}\mathbf{F} + \mu(\mathbf{F}-\mathbf{Y})^\top(\mathbf{F}-\mathbf{Y})\}$$

$$\min_{\mathbf{F},\mathbf{Y}} \tfrac{1}{2}\operatorname{tr}\left\{\mathbf{F}^T\mathbf{L}\mathbf{F} + \mu(\mathbf{F}-\mathbf{V}\mathbf{Y})^T(\mathbf{F}-\mathbf{V}\mathbf{Y})\right\}$$

$$s.t. \quad \mathbf{Y}_{ij}\in\{0,1\}, \textstyle\sum_j \mathbf{Y}_{ij} = 1$$

Alternate Optimization

- First, given Y solve continuous valued $\mathbf{F}$

$$\frac{\partial \mathcal{Q}}{\partial \mathbf{F}^*} = 0 \Rightarrow \mathbf{F}^* = (\mathbf{L}/\mu + \mathbf{I})^{-1}\mathbf{V}\mathbf{Y} = \mathbf{P}\mathbf{V}\mathbf{Y} \qquad \mathbf{P} = (\mathbf{L}/\mu + \mathbf{I})^{-1}$$

- Then, search optimal integer Y given F*

$$\mathcal{Q}(\mathbf{Y}) = \frac{1}{2}\mathrm{tr}\left(\mathbf{Y}^T\mathbf{V}^T\left[\mathbf{P}^T\mathbf{L}\mathbf{P} + \mu(\mathbf{P}^T - \mathbf{I})(\mathbf{P} - \mathbf{I})\right]\mathbf{V}\mathbf{Y}\right)$$

Gradient decent search

Alternate Minimization for Label Tuning

Example:

$$\mathbf{Y}^T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \qquad \nabla_Y Q = \begin{bmatrix} 0.8 & 0.1 \\ -0.23 & -0.25 \\ -0.31 & 0.07 \\ -0.17 & -0.04 \end{bmatrix}$$

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

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

add label (3,1)

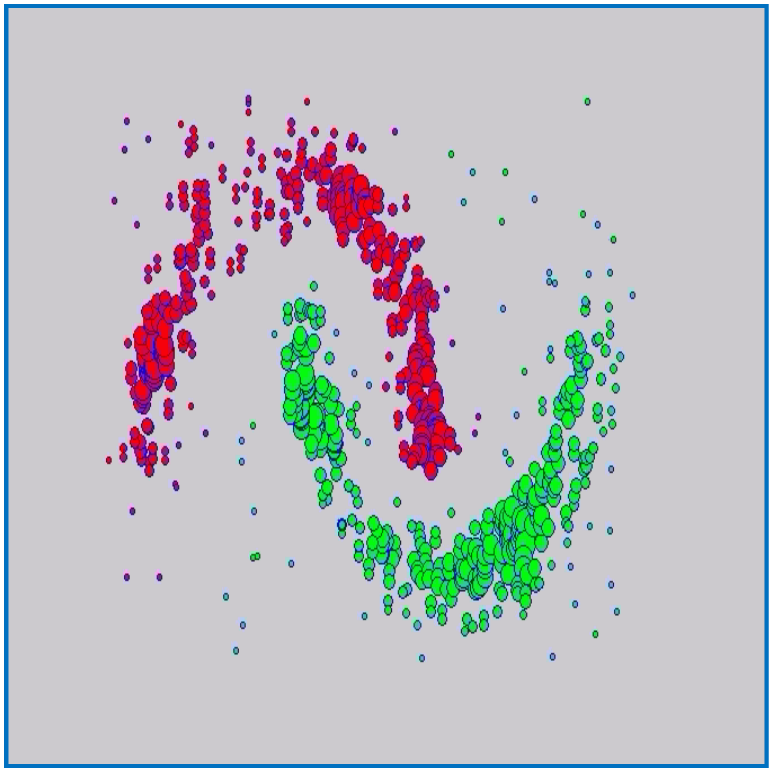
delete label (1,1)

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

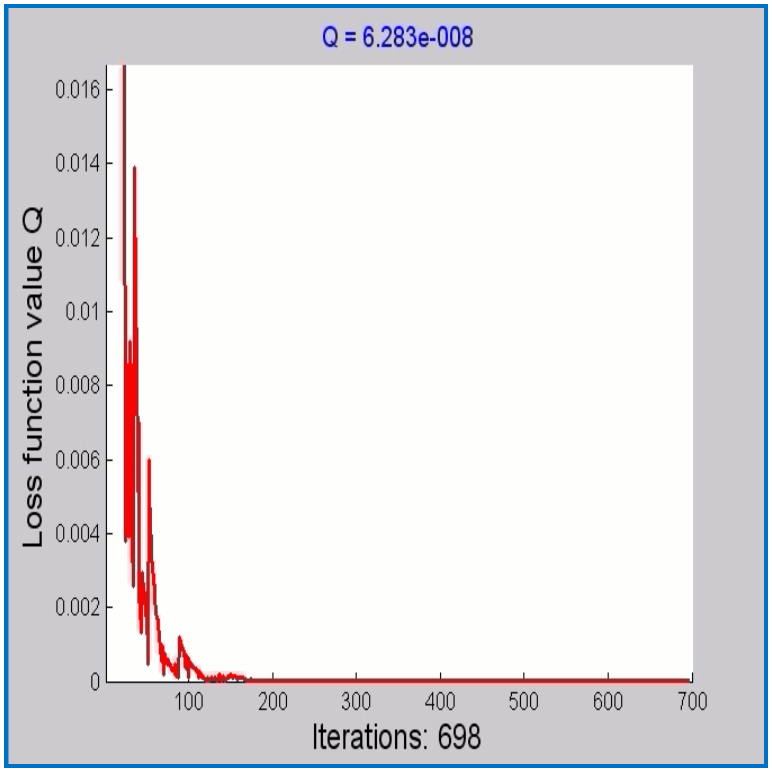
- Iteratively repeat the above procedure

Example – Toy Data

Consider adding label only



Label propagation by GTAM



Convergence procedure

(non-monotonic due to discrete step size)

Unlabeled **Positive** **Negative**

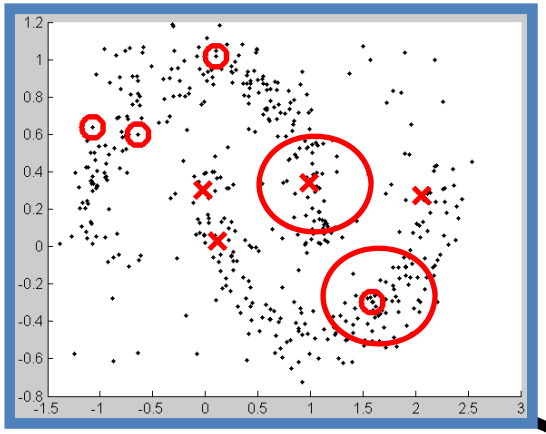
Label Diagnosis and Self Tuning

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

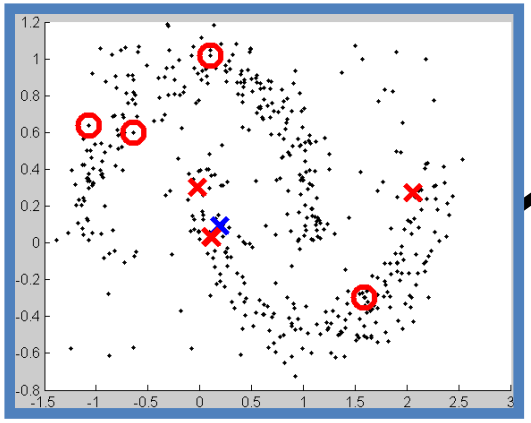
Add label: $(i^+, j^+) = \min_{i,j} \nabla_{(vy_u)} \mathcal{Q};\ y_{i^+j^+} = 1$

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

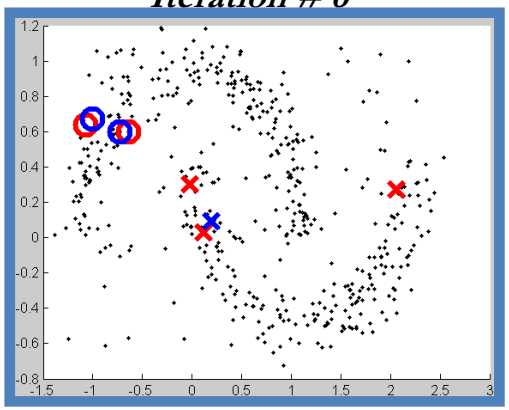
Initial Labels

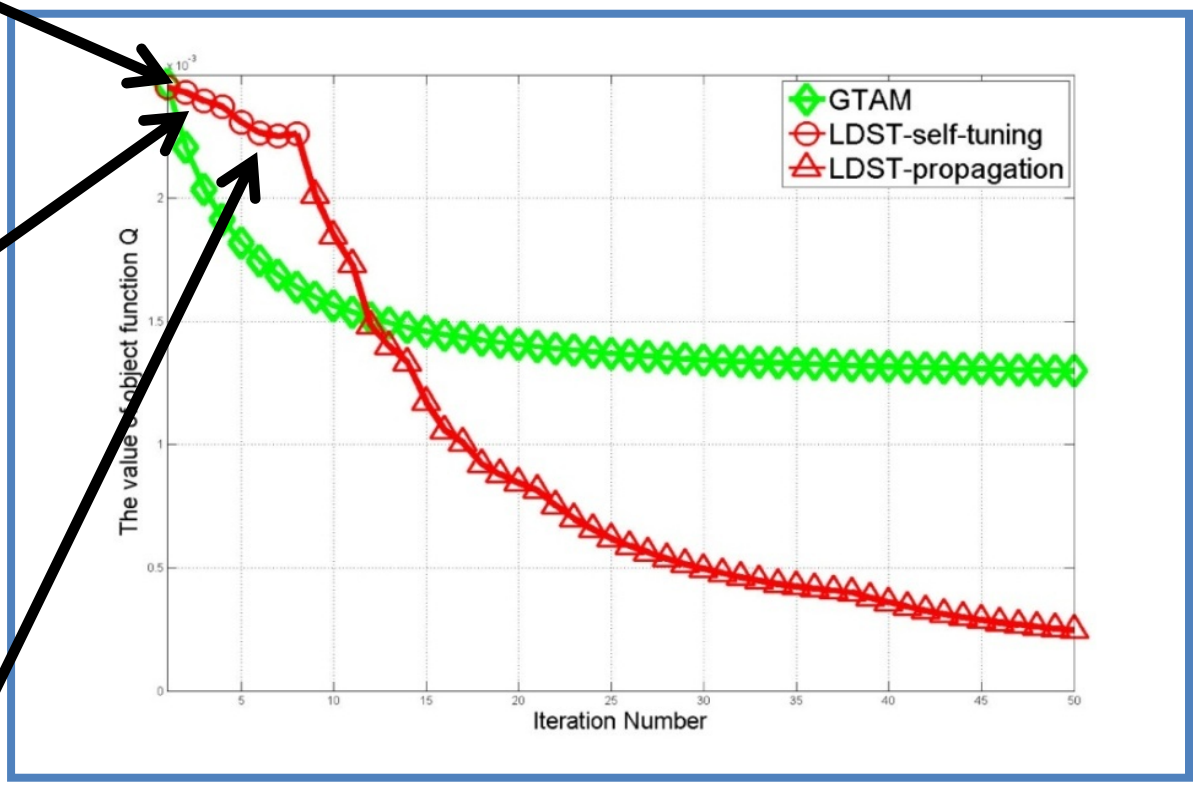


Iteration # 2



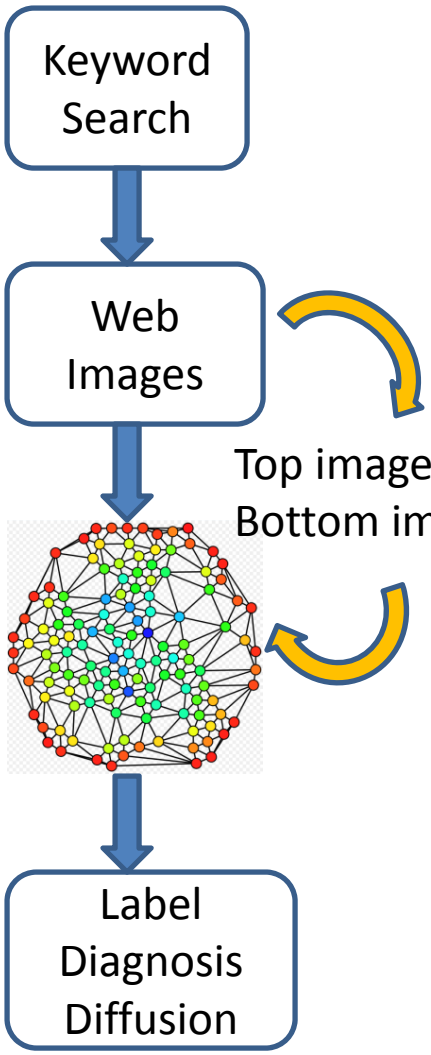
Iteration # 6





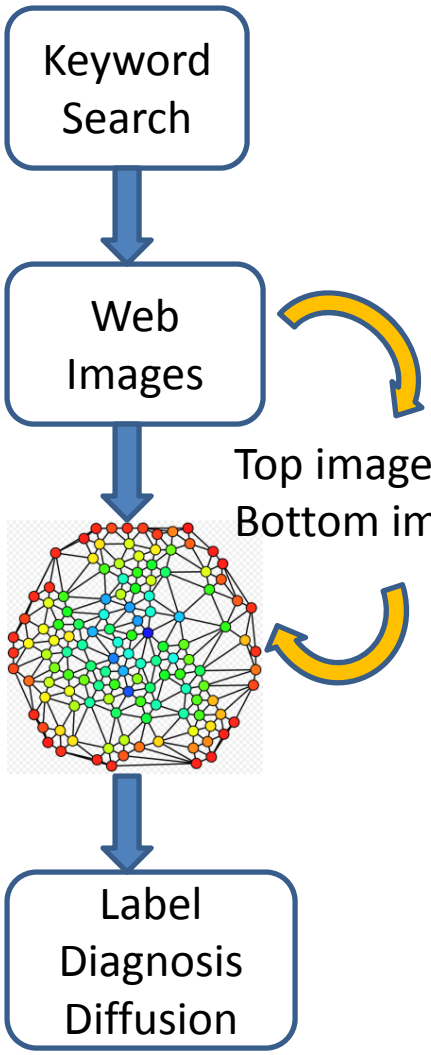
Decline of the cost function Q over iterations (with vs. without label tuning)

Application: Web Search Reranking



Google Search "Tiger"

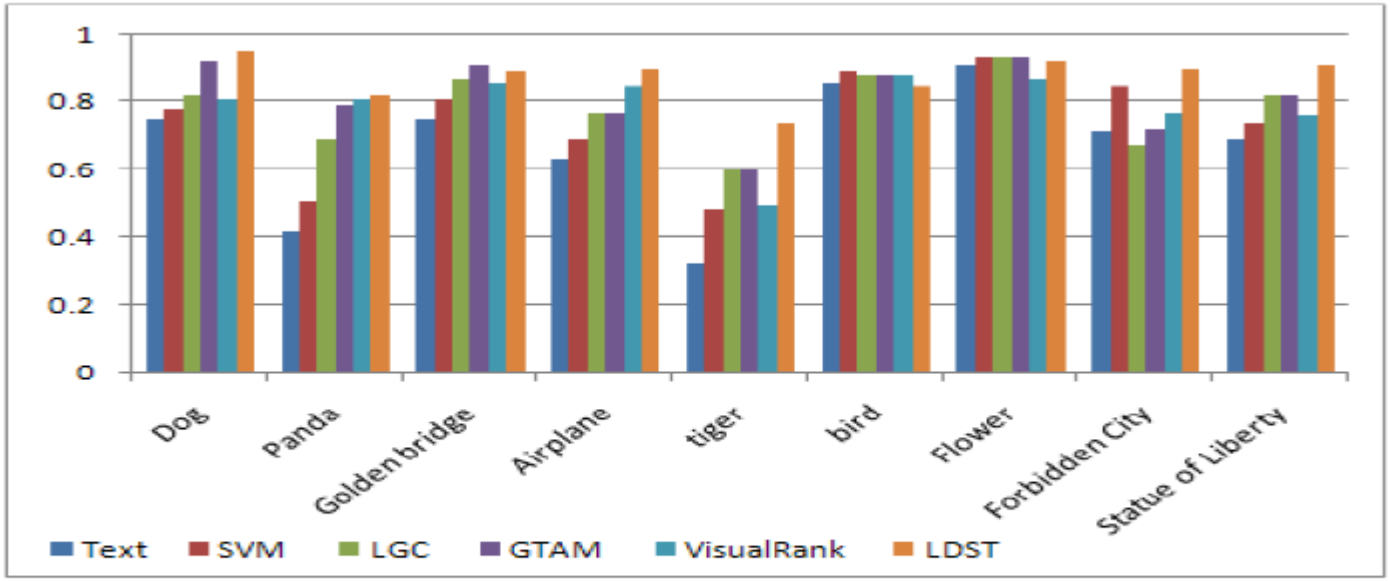
Application: Web Search Reranking



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



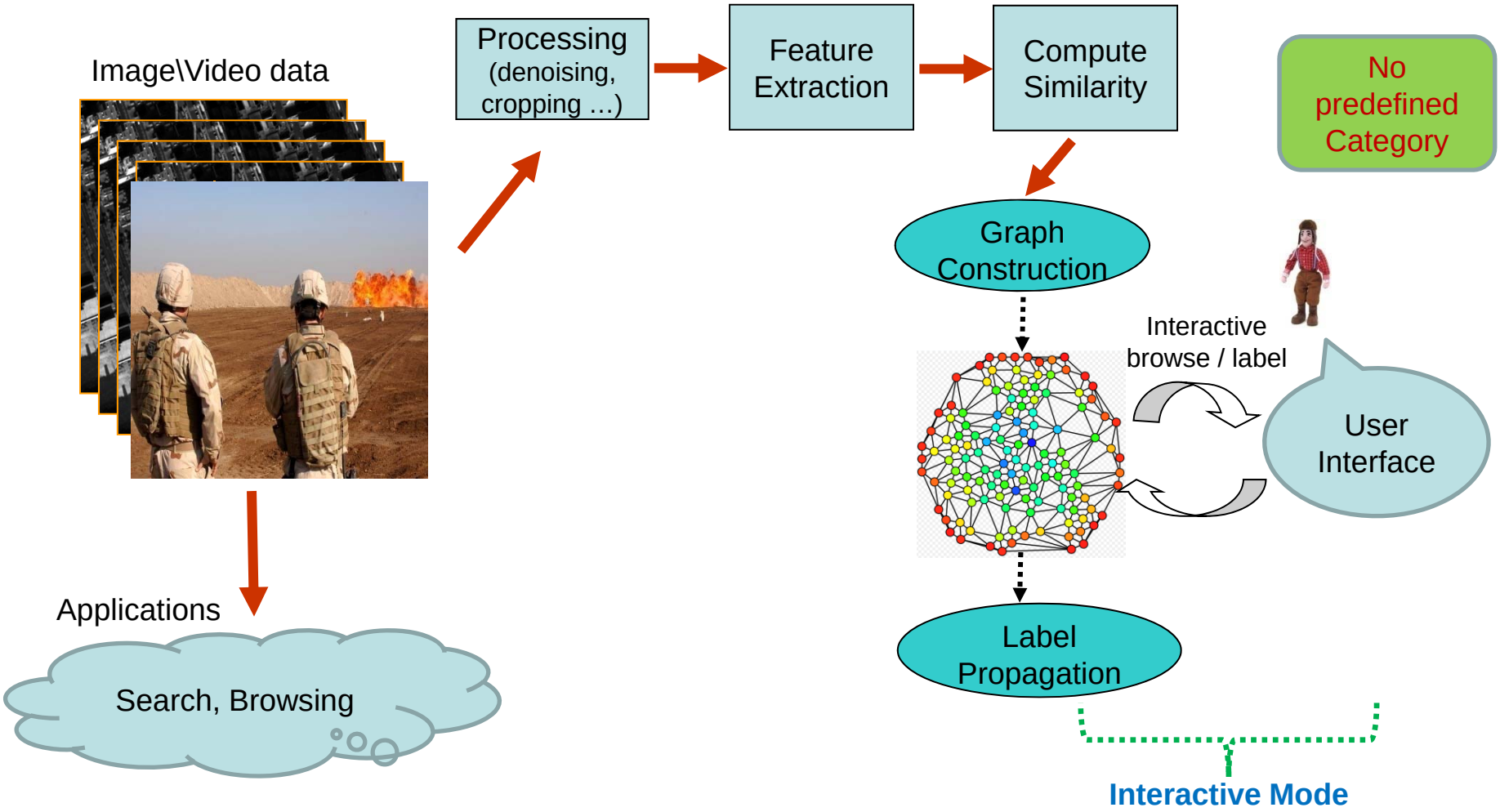
The comparison of the precision of the top 100 ranked images over different categories of images.

Method	Text	SVM	LGC	GTAM	VisualRank	LDST
Accuracy (%)	67.11	74.22	79.89	81.44	79.00	87.56

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



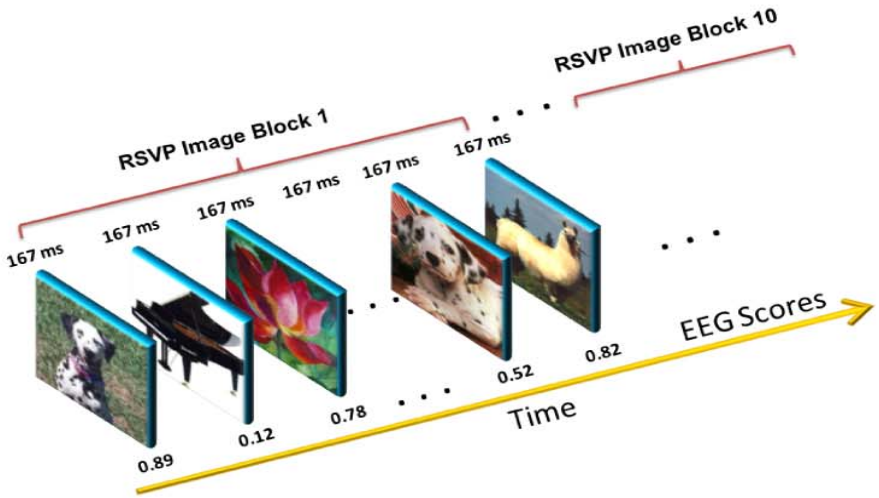
Application: Brain Machine Interface for Image Retrieval

-- denoise unreliable labels from brain signal decoding

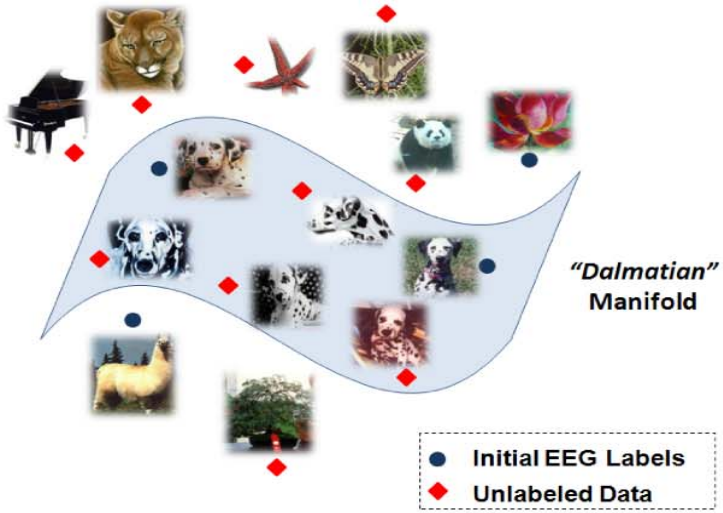
(joint work with Sajda et al, ACMMM 2009, J. of Neural Engineering, May 2011)

Use EEG brain signals to detect target of interest

Use image graph to tune & propagate information



Rapid Serial Presentation of Caltech 101 Images

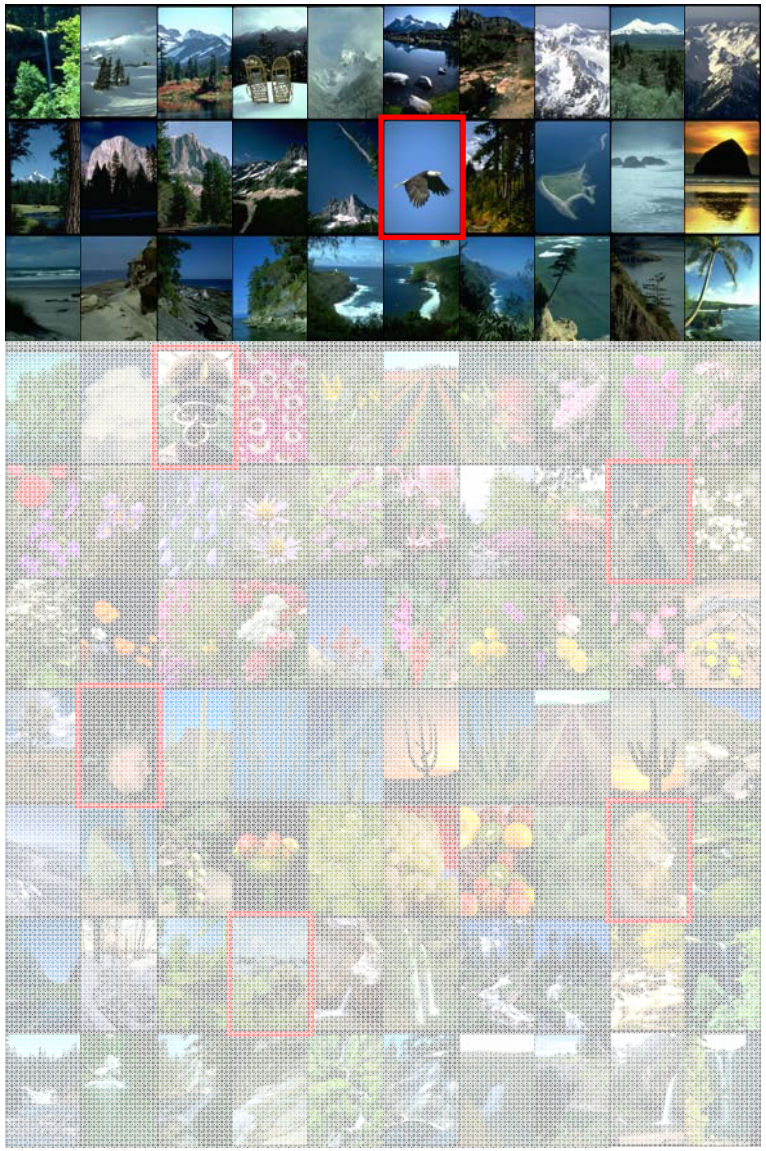


Graph-Based Visual Pattern Discovery

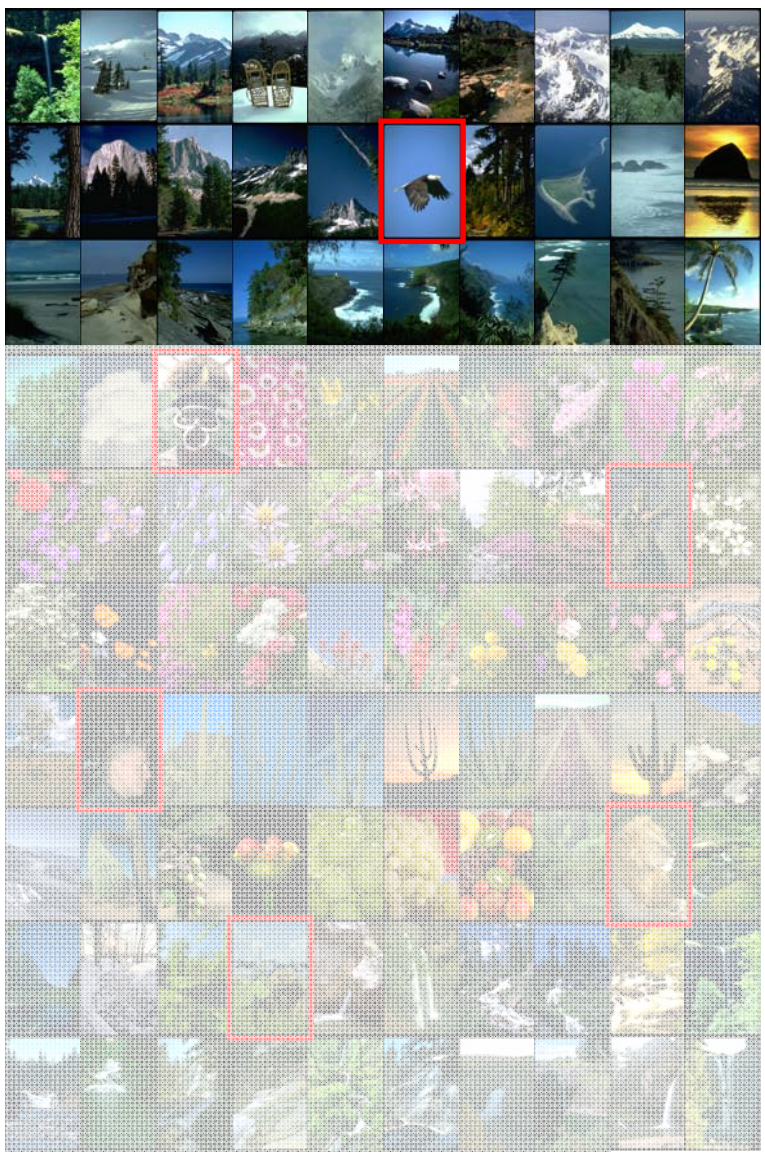
COLUMBIA UNIVERSITY

The Paradigm

Database (any target that may interest users)



The Paradigm

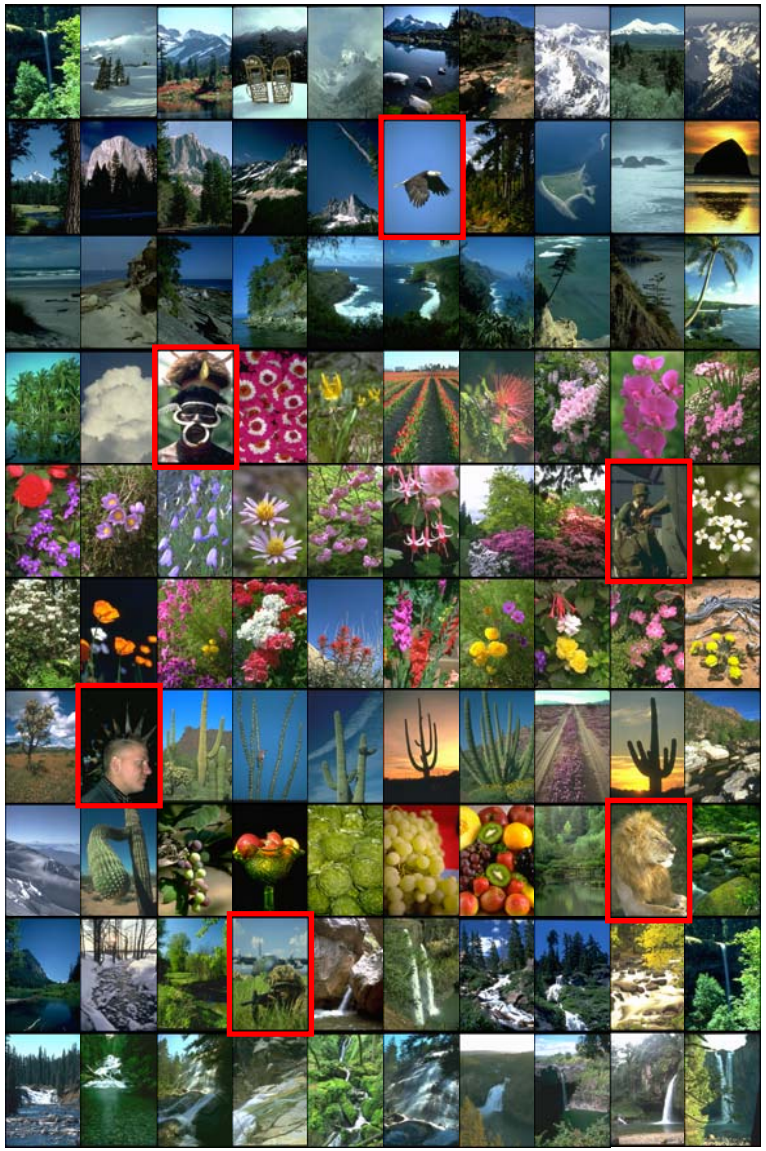


Database

Neural (EEG) decoder

EEG-scores

The Paradigm



Database

Neural (EEG) decoder

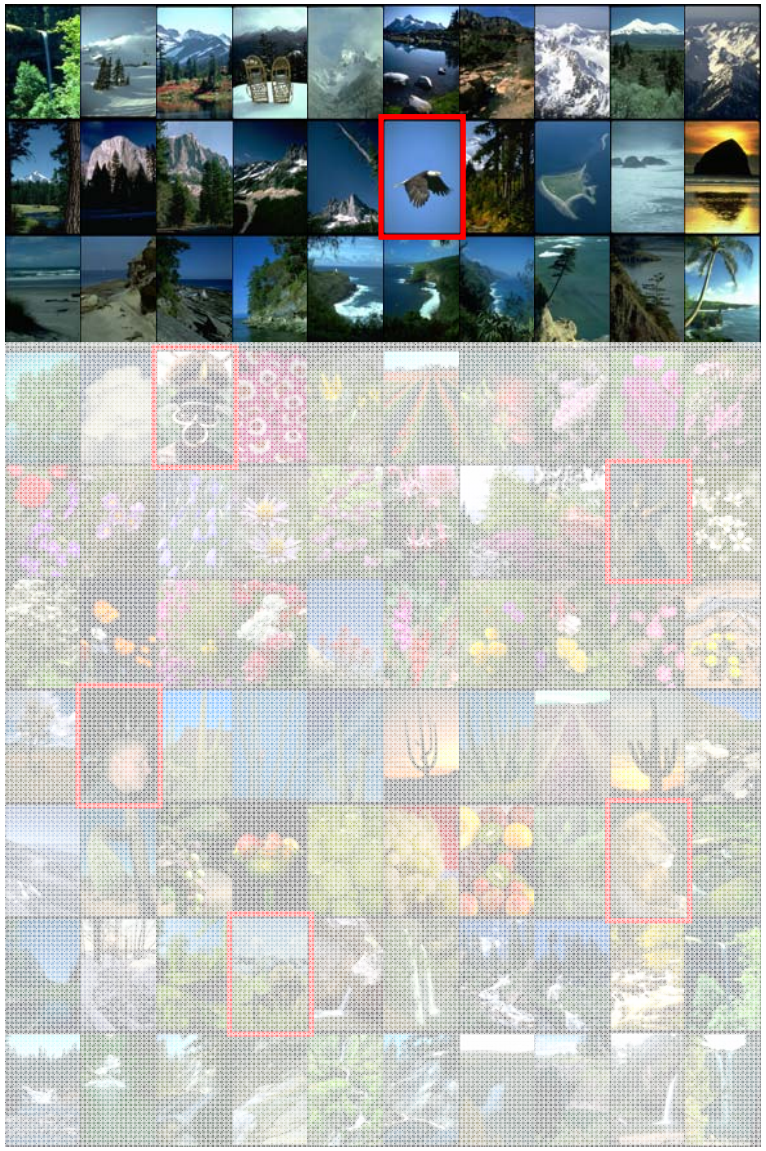
Exemplar labels (noisy)

image features

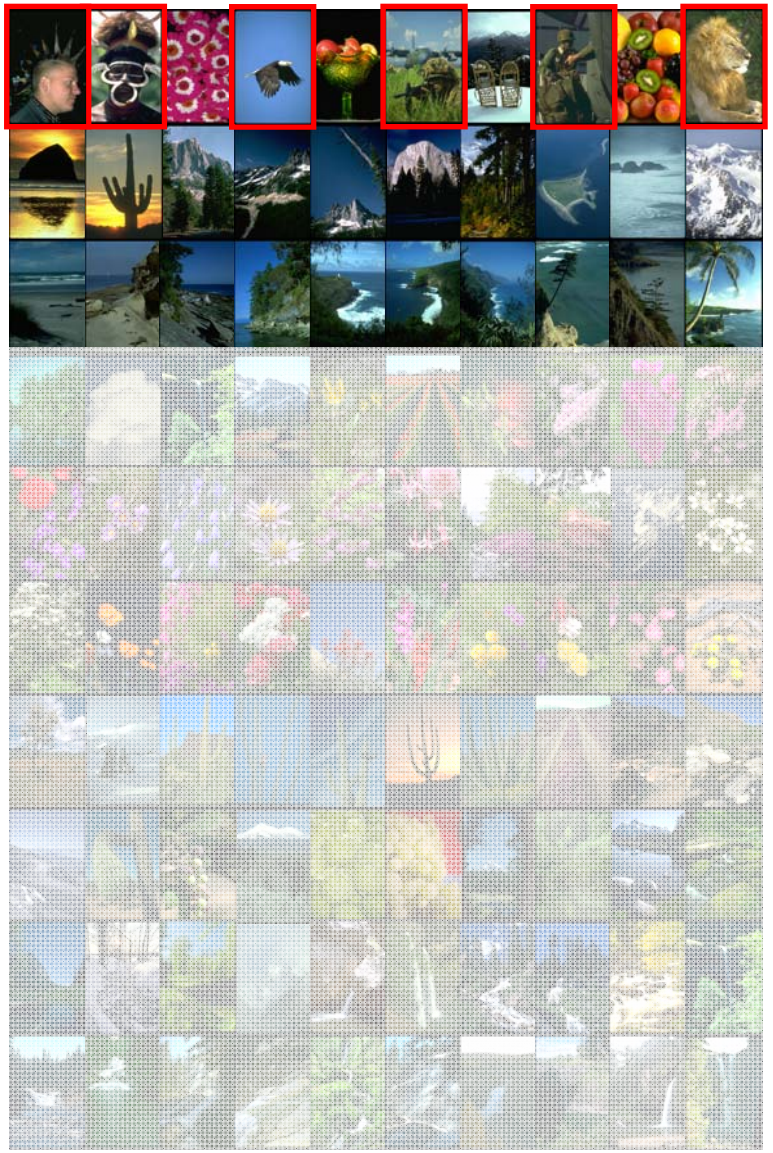
Graph-based
Semi-Supervised
Learning

prediction score

The Paradigm

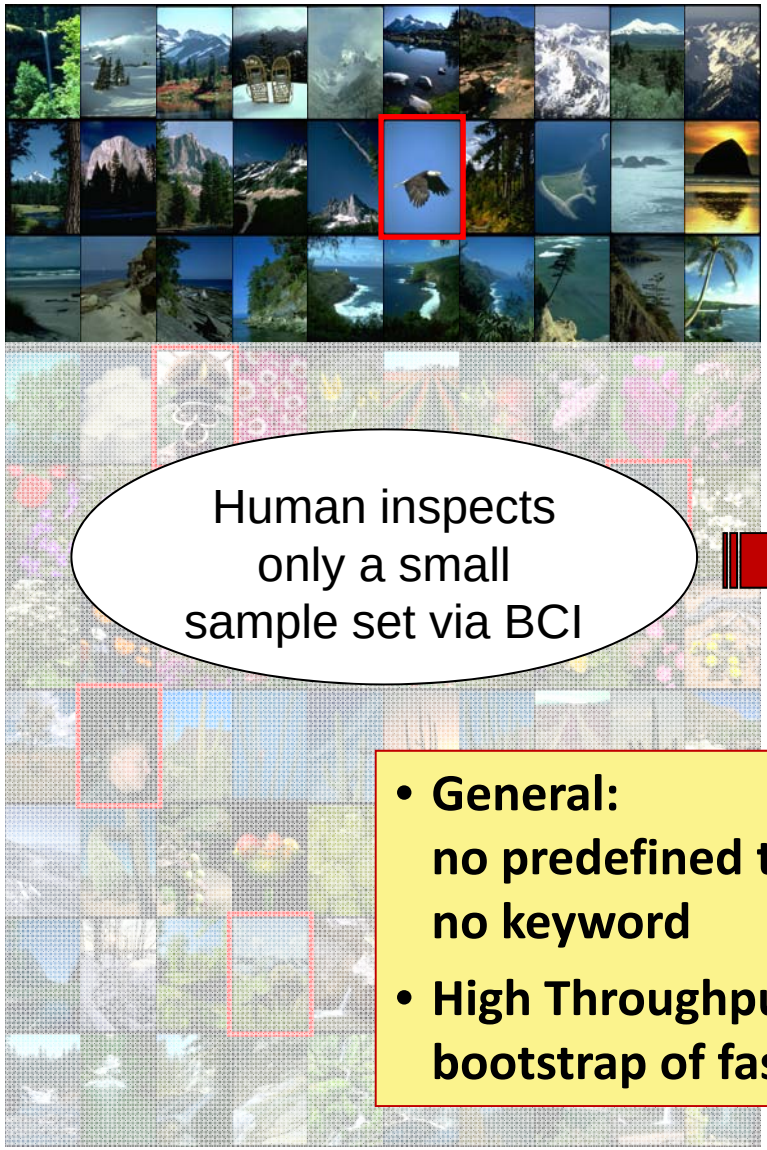


Pre-triage

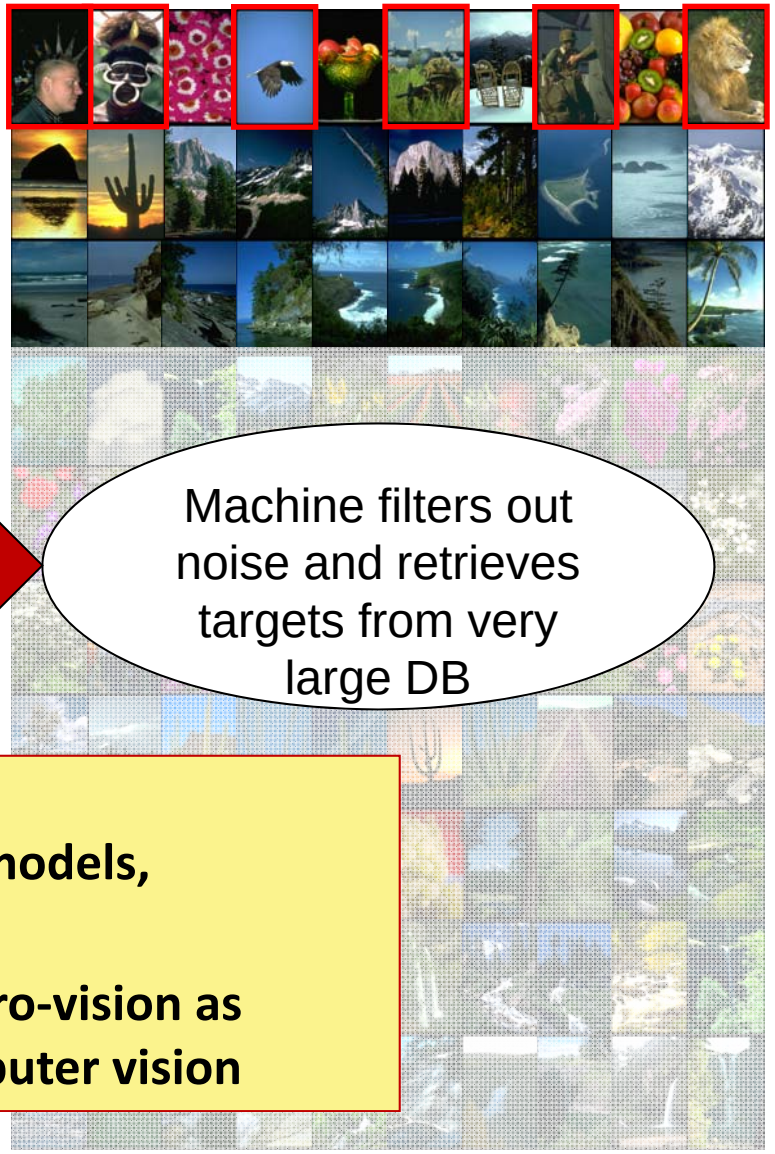


Post-triage

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**

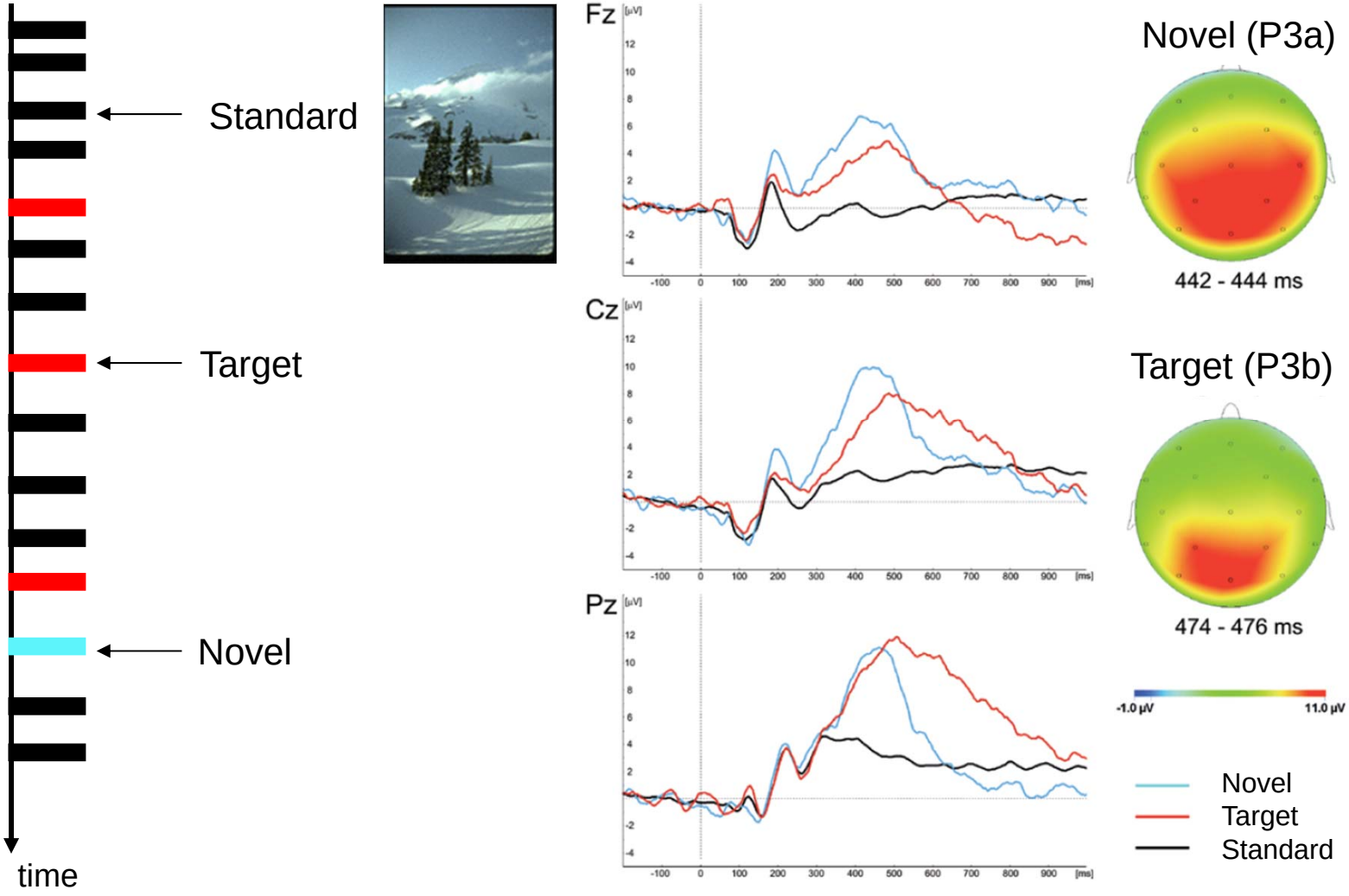
Pre-triage

Post-triage

COLUMBIA UNIVERSITY

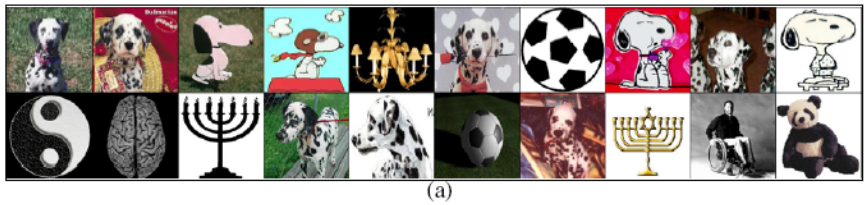
The Neural Signatures of “Recognition”

D. Linden, Neuroscientist, 2005, ***the Oddball Effect***

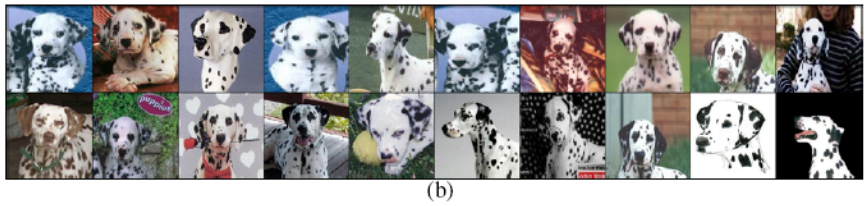


Effect of graph-based reranking (BCI test)

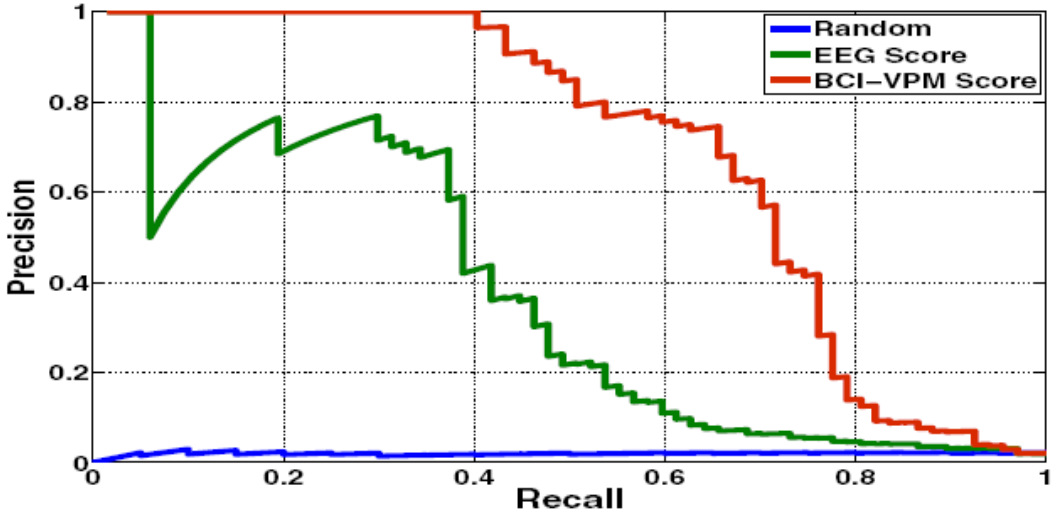
Top (noisy) results of Brain EEG signal detection



Top results after graph-based label denoising & propagation

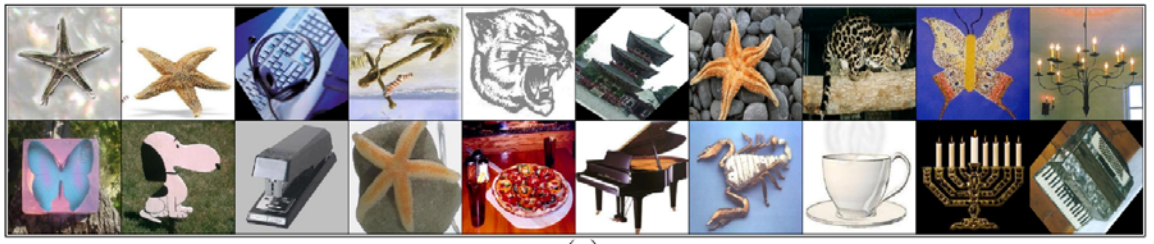


P-R curve significantly improved



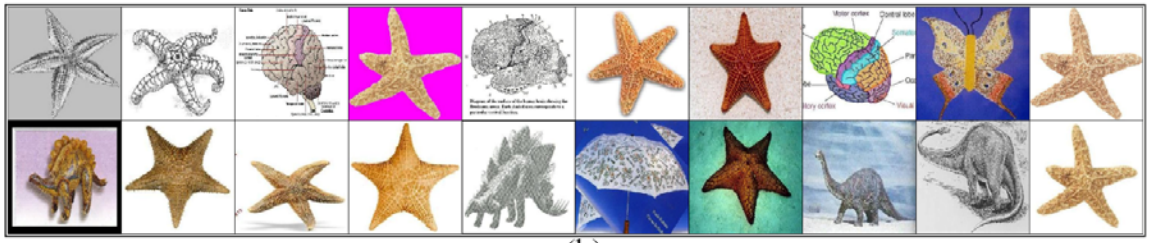
More Example Results

Top 20 results of
EEG detection



(a)

Top 20 results of
Hybrid System (BCI-VPM)



(b)

Top 20 results of
EEG detection

(a)

Top 20 results of
Hybrid System (BCI-VPM)

(b)

Graph over million points and more

- k-NN graph construction + label prediction

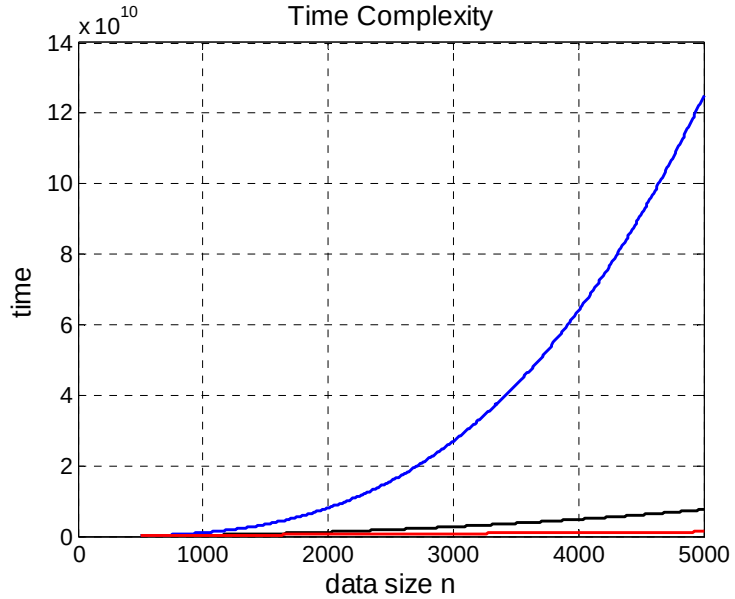
$$G(V, E, W), L = D - W$$
$$\hat{L} = D^{-1/2} L D^{-1/2}$$
$$\text{time} = O(kn^2)$$

$$\text{LGC} : F = (I + \gamma \hat{L})^{-1} Y, \gamma > 0$$
$$\text{GraphReg} : F = (J + \gamma L)^{-1} Y, \gamma > 0$$
$$\text{GFHF} : F = (J + \gamma L)^{-1} Y, \gamma \to 0$$
$$\text{time} = O(n^3)$$

- infeasible for large-scale tasks
- Idea: AnchorGraph Regularization

complexity: $O(m^2 n)$
anchors m << 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 eigenfucntions 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)

$$\mathbf{W} = \mathbf{W}_{nm}\mathbf{W}_{mm}{}^{-1}\mathbf{W}_{nm}{}^{T}$$

  - Complexity $\boldsymbol{O}(dmn)$
  - But may not be positive semidefinite -> non-convex

Idea: Build low-rank graph via anchors

(Liu, He, Chang, ICML10)

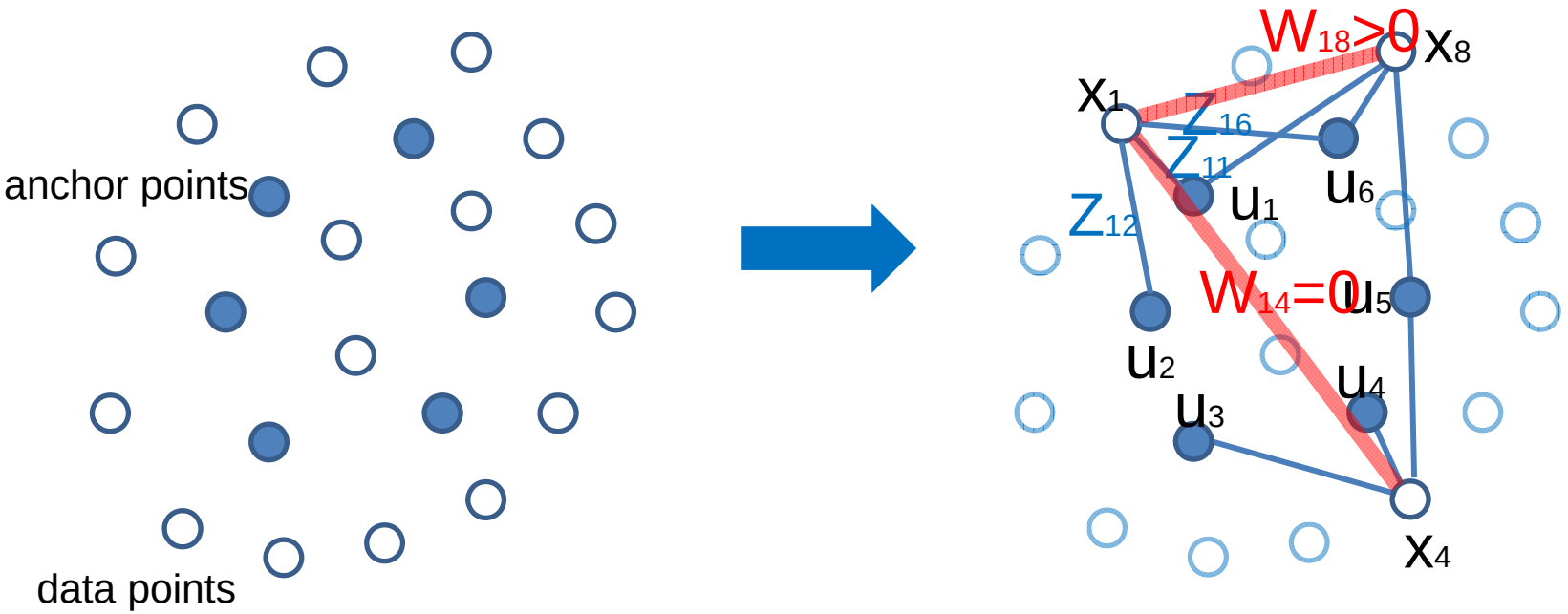
- Use anchor points to “abstract” the graph structure
- Compute data-to-anchor similarity: sparse local embedding

$$Z \in \mathbb{R}^{n \times m}, m \ll n$$

- Data-to-data similarity *W* = inner product in the embedded space

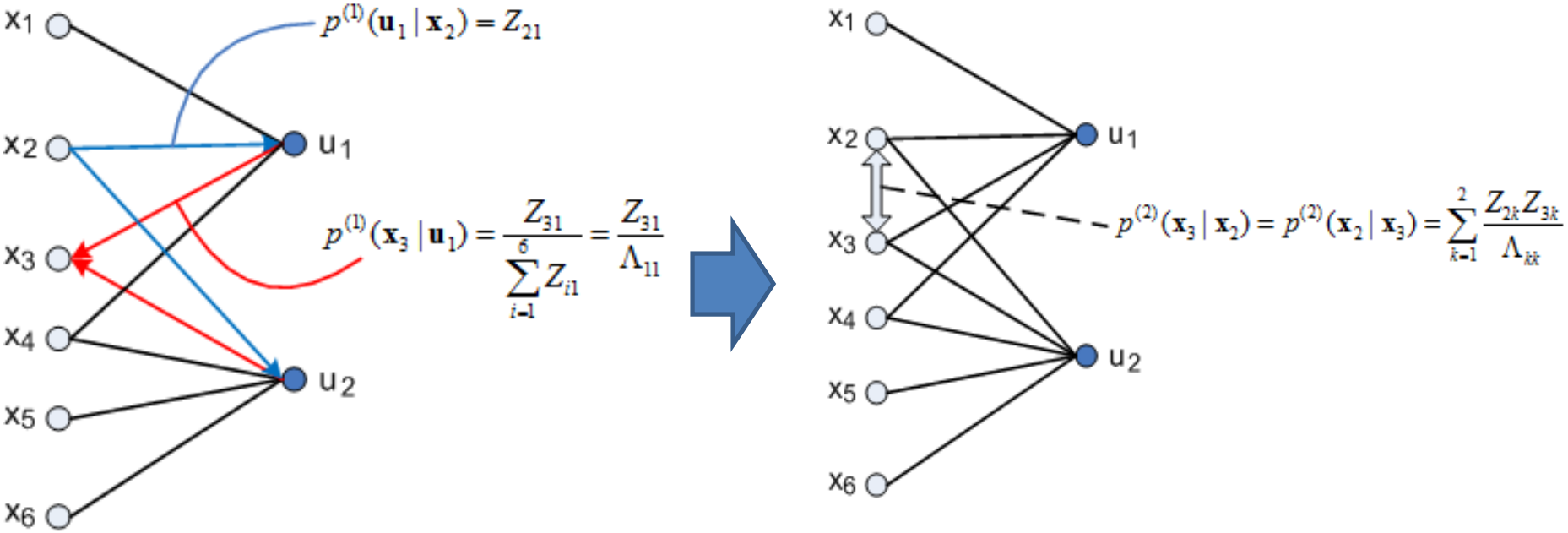
$$W_{ij} = \sum_{k=1}^{m} Z_{ik} Z_{jk} = Z_{i.} Z_{j.}^{\top}$$

$$\min_{Z_{ik}} \|x_i - \sum_{k \in \langle i \rangle} Z_{ik} u_k\|^2 \;\; s.t. \;\; \sum_{k \in \langle i \rangle} Z_{ik} = 1, Z_{ik} \geq 0$$



Probabilistic Intuition

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



$\mathbf{W} = \mathbf{Z}_{nm}\, \mathbf{\Lambda}_{mm}{}^{-1} \mathbf{Z}_{nm}{}^{T}$, where $\mathbf{\Lambda} = diag(\mathbf{1}^T \mathbf{Z})$, $m << n$

AnchorGraph: sparse, positive semi-definite

AnchorGraph Regression

- Apply the same sparse embedding principle to labels

$$f(x_i) = \sum_{k=1}^{m} Z_{ik} f(u_k), \qquad (\text{multi-class}) F = ZA$$

- The whole graph regularization process becomes low-rank

$$\min_{A \in \mathbb{R}^{m \times c}} \|Z_l A - Y_l\|_{\mathrm{F}}^2 + \gamma \mathrm{tr}\left(A^\top Z^\top (I - Z\Lambda^{-1} Z^\top) ZA\right)$$

$$A^* = \boxed{\left(Z_l^\top Z_l + \gamma Z^\top Z - \gamma (Z^\top Z)\Lambda^{-1}(Z^\top Z)\right)^{-1}} Z_l^\top Y_l$$

Small matrix inversion

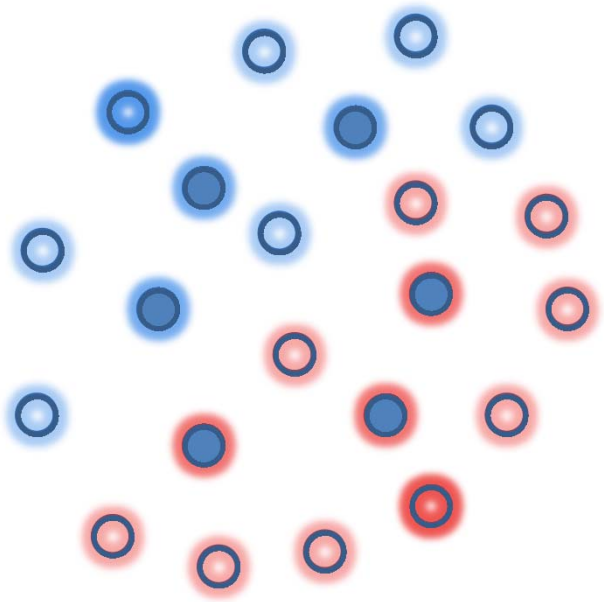
$O(m^2 n)$

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

Predicted function over graph = embedding matrix · inferred labels on anchors

Intuition: Anchor Graph SSL

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



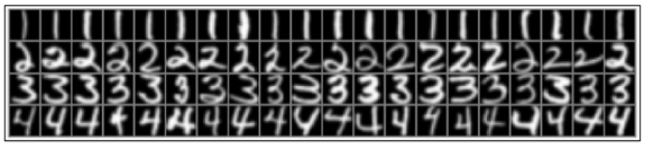
label mapping **out**

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

$O(m^2 n)$

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*

Method	Error Rate (%)	Time (seconds)
1NN	20.15	0.12
LGC with 6NN graph	8.79	403.02
GFHF with 6NN graph	5.19	413.28
AGR^0	7.40	10.20
AGR	**6.56**	16.57

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%

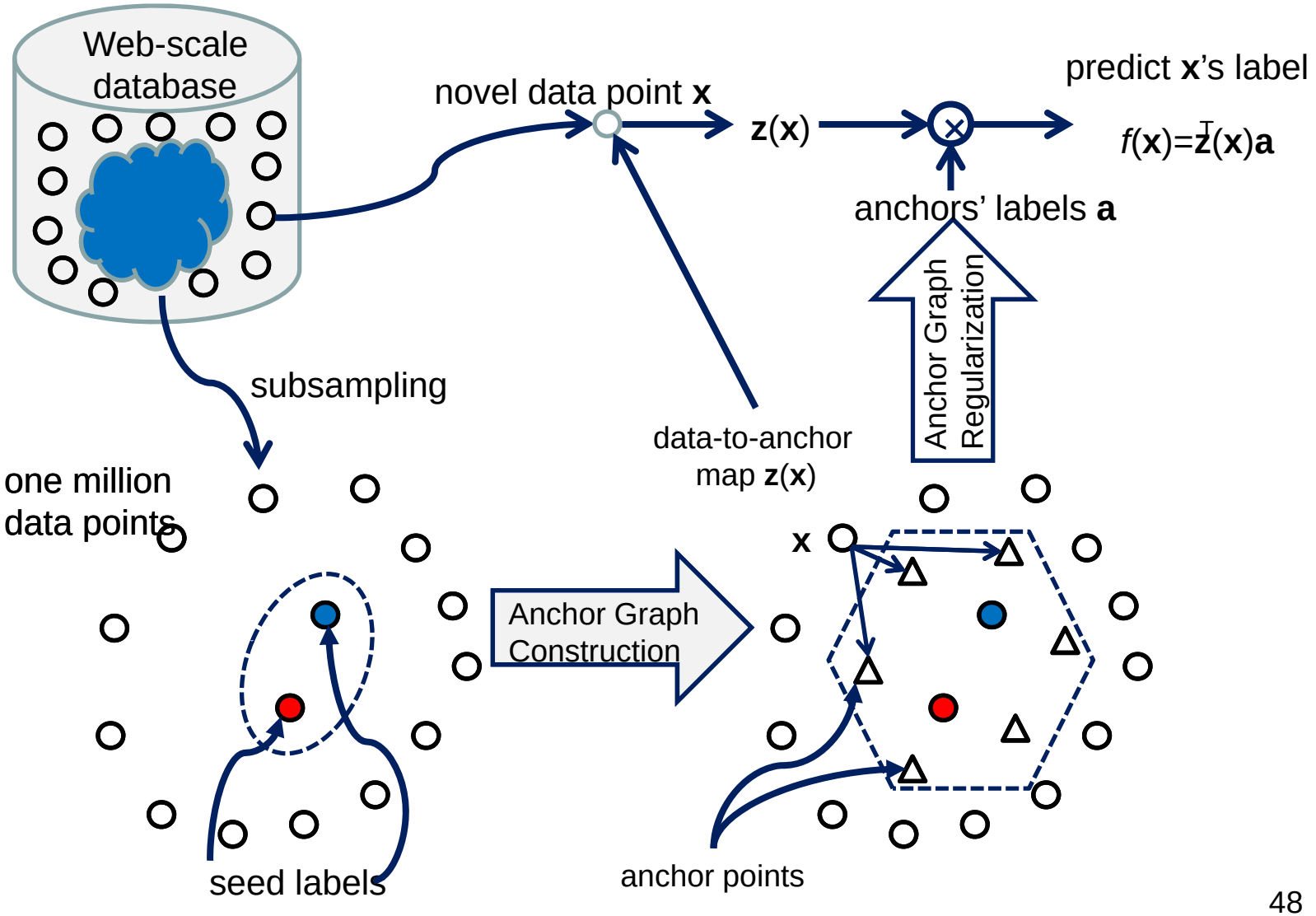
Method	Error Rate (%)	Training Time (seconds)
1NN	39.65	5.46
Eigenfunction (‘09)	36.94	44.08
PVM (‘09)	29.37	266.89
AGR^0	24.71	232.37
AGR	**19.75**	331.72

30%-50% gain

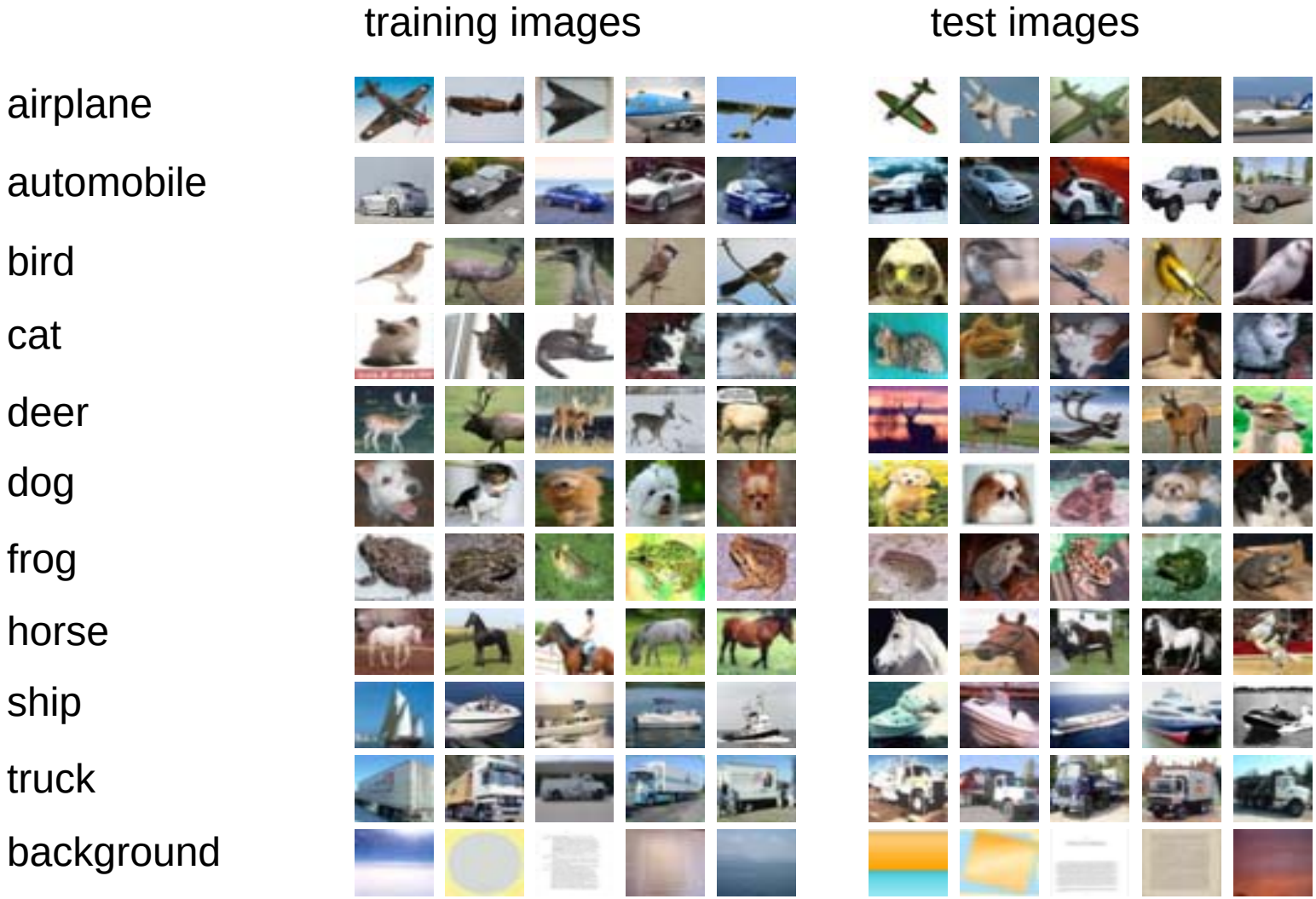
Extension to Web-Scale

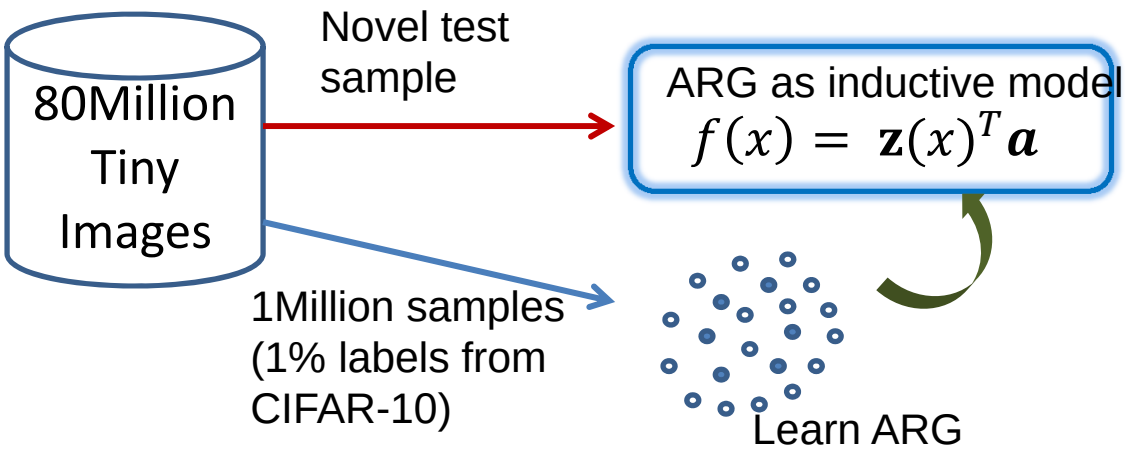
- Techniques described above not scalable to Web-scale or dynamic data sets
  - Cannot handle cases when n = ~ billions
  - For dynamic data, updating graph is expensive
- Preferred:
  learn *Inductive Models* to handle novel dynamic data

Data Subsampling & Learn Inductive Model



ARG over 80M Tiny Images + CIFAR-10





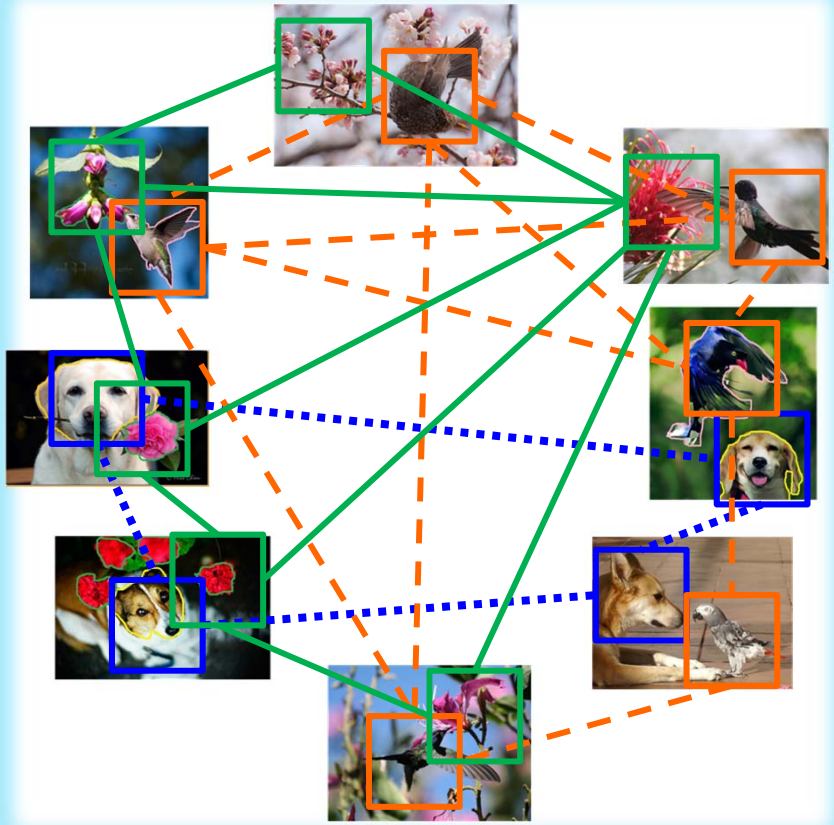
Method	1NN	Linear SVM	Eigen Function	PVM		AGR	
				1K Anchors	2k Anchors	1K Anchors	2k Anchors
Accuracy (%)	51.66±0.28	60.14±0.34	53.86±0.35	60.55±0..32	60.95±0.41	62.39±0.33	64.23±0.28
Training Time (s)	0	8.00	149.83	213.88	517.82	206.60	477.61
Test Time (s)	6.29e-4	2.66e-6	1.39e-4	5.79e-5	1.27e-4	6.20e-5	1.39e-4

Additional Issues

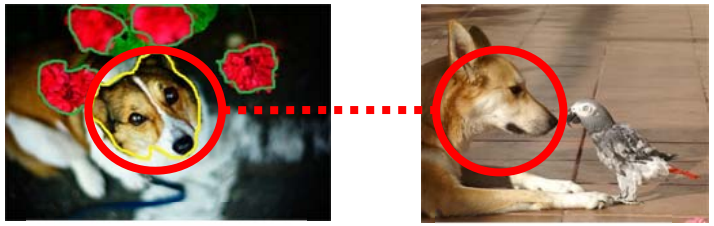
- Multi-edge Graph
  - Multiple relation edges between nodes
- Multi-feature Graph
  - Build graphs in multiple feature spaces
  - Joint optimization
- Label tuning vs. Active Learning

Image-Based Multi-Edge Graph

Liu et al, ACM Multimedia 2010



two images with the same tag

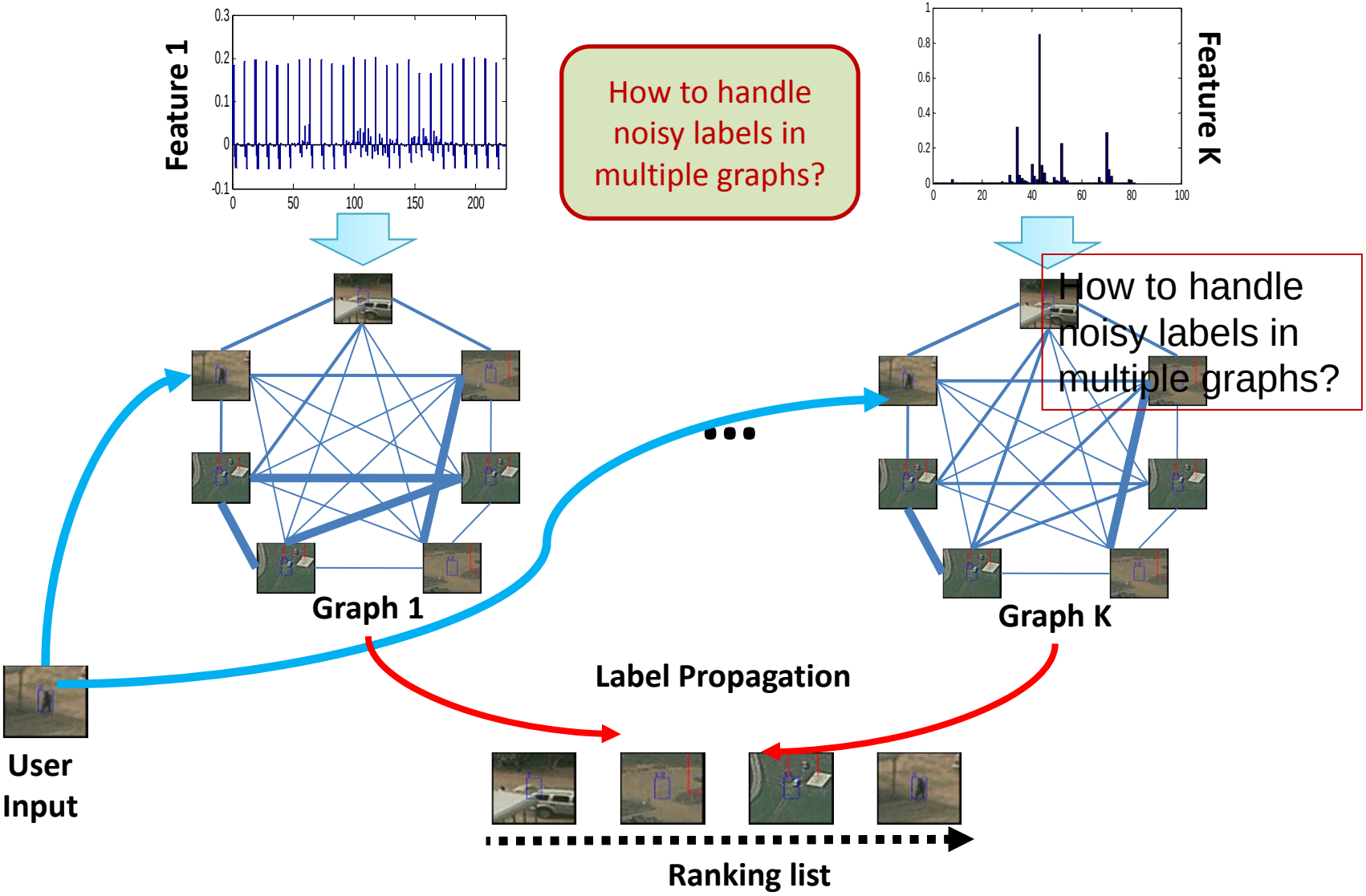


dog, flower

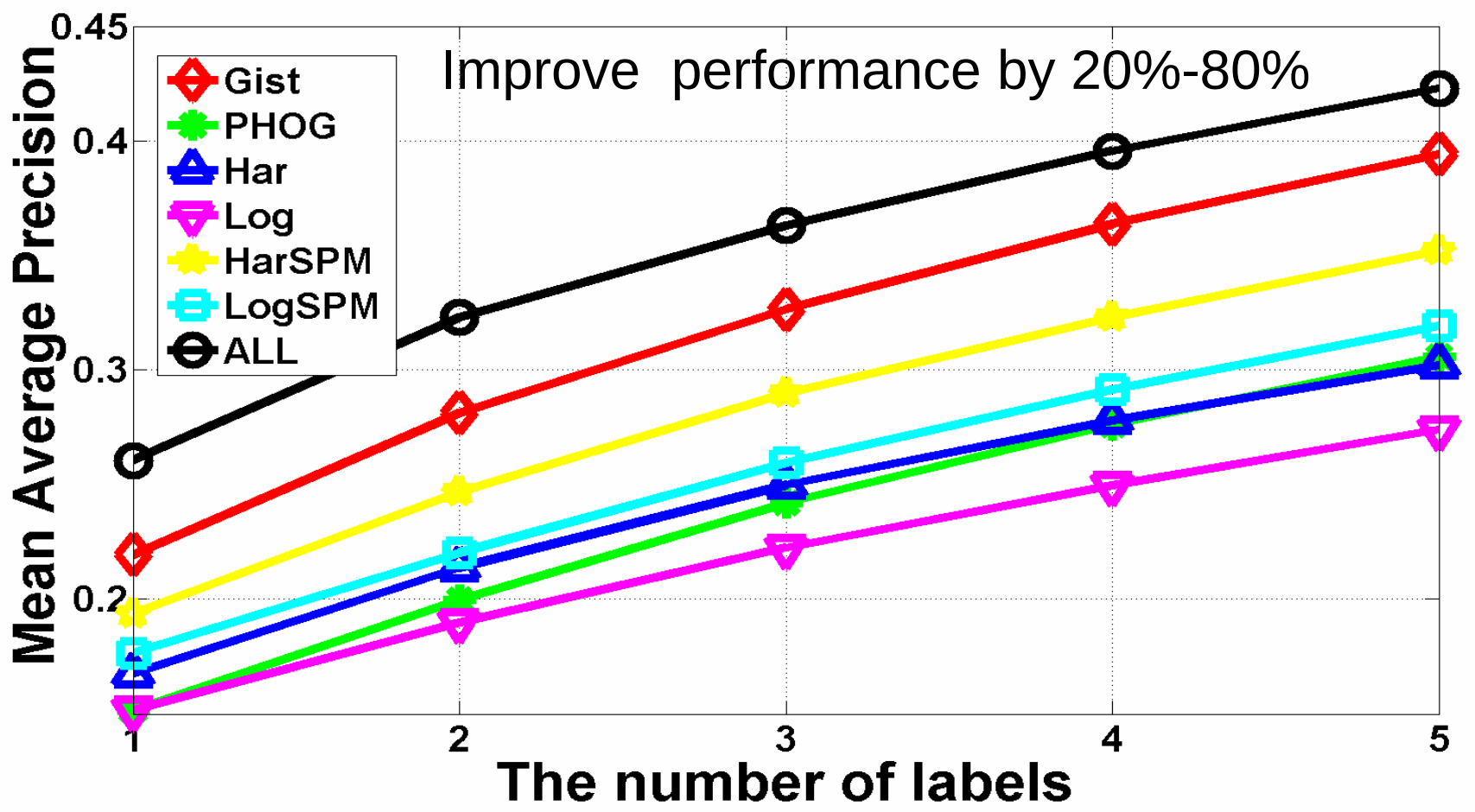
dog, bird

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

Extension to Multi-Feature Graphs



Multi-graph SSL vs. single-graph



Caltech 101 data set

References and Tools

1. X. Zhu, Z. Ghahramani, and J. D. Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. *ICML*, 2003.
2. D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Scholkopf. Learning with local and global consistency. *NIPS*, 2004.
3. W. Liu, J. He, and S.-F. Chang. **Large graph construction** for scalable semi-supervised learning. ICML, 2010. Software: http://www.ee.columbia.edu/~wliu/Anchor Graph.zip.
4. W. Liu, J. Wang, S. Kumar, and S.-F. Chang. Hashing with graphs. *ICML*, 2011.
5. J. Wang, T. Jebara, and S.-F. Chang. **Graph transduction via alternating minimization**. *ICML*, 2008.
6. J. Wang, Y.-G. Jiang, and S.-F. Chang. **Label diagnosis through self tuning** for web image search. *CVPR*, 2009.
7. W. Liu, J. Jun, and S.-F. Chang, Robust and Scalable Graph-Based Semi-Supervised Learning. In Review, *IEEE Proceedings*, 2012.
8. J. Wang, E. Pohlmeyer, B. Hanna, Y.-G. Jiang, P. Sajda, and S.-F. Chang, "**Brain State Decoding** for Rapid Image Retrieval," ACM Multimedia Conference, 2009.
9. J. Wang, A. Kumar, S.-F. Chang, "**Semi-Supervised Hashing** for Scalable Image Retrieval", CVPR 2010.