Optimization

Relaxation

RatioCut

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

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

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

Since $(|A|+|B|)$ is constant, $\min_{A,B} RCut(A,B) = \min_{f}\ f^T L f$, subject to $f_i \in \left\{\sqrt{\frac{|B|}{|A|}}, -\sqrt{\frac{|A|}{|B|}}\right\}$ ➡ $\boxed{|A| = |B|}$

Still NP hard…Require relaxation.

$$\|f\|^2 = \sum_i f_i^2 = |A|\frac{|B|}{|A|} + |B|\frac{|A|}{|B|} = |A| + |B| = N$$

$$|A| = |B| \rightarrow \sum_i f_i = 0 \leftrightarrow f \perp \mathbf{1}_N$$

Optimization formulation for RatioCut (same with balanced Mincut)

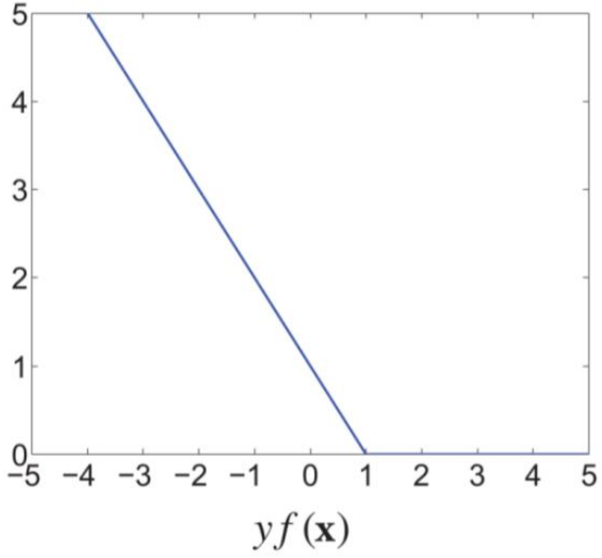
$\min_f f^T L f$ subject to $f_i \in R, f \perp \mathbf{1}_N,\ \ \|f\| = \sqrt{N}$

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

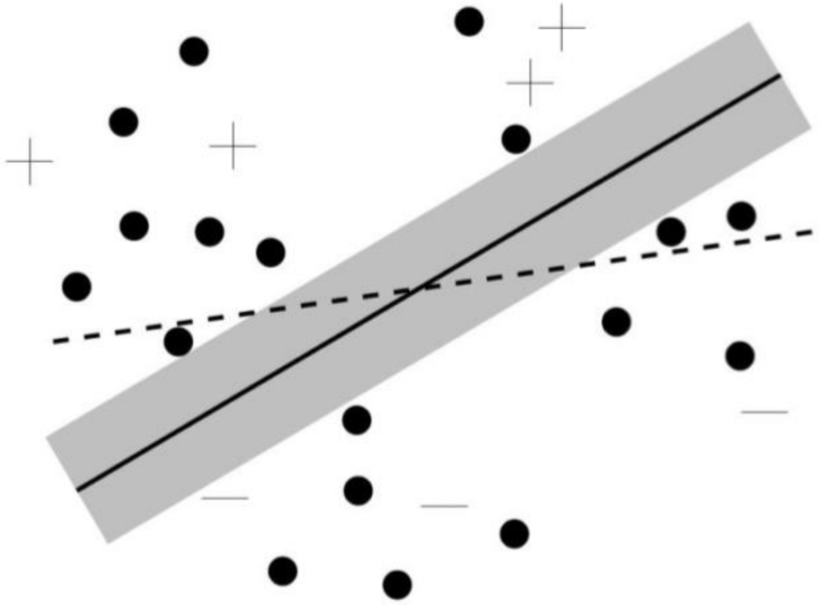
Questions of the last lecture

- What does hinge loss in SVM penalize?

→ Hinge loss penalizes the case that the classifier f(w, x) does not decide the correct class of labeled x with the score margin of y f(w, x)>=1.



hinge loss

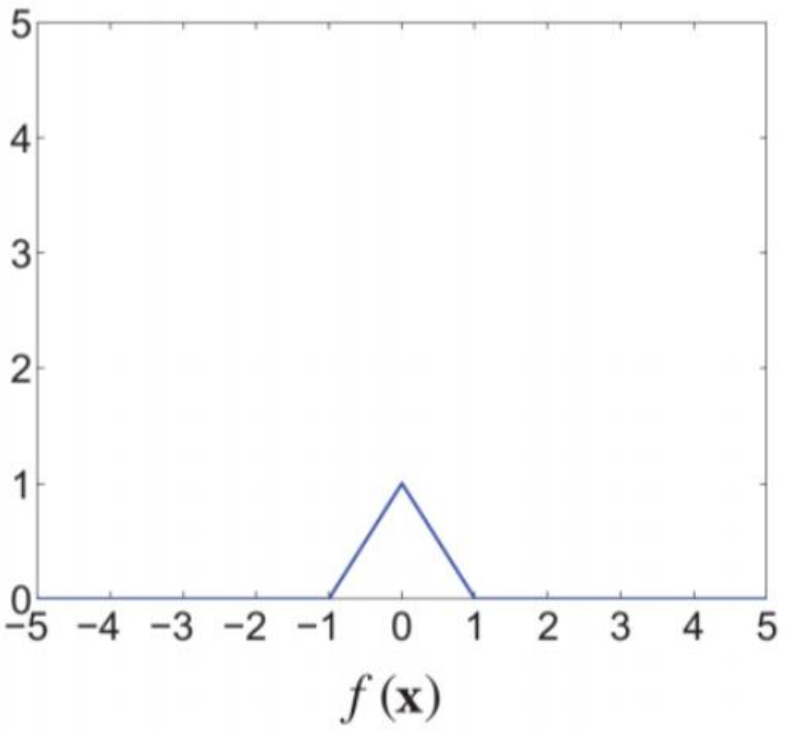


$$\Phi(\boldsymbol{x}_i, y_i, \boldsymbol{f}(\mathbf{w}, b; \boldsymbol{x}_i)) = \max(1 - y_i(\mathbf{w}^T \boldsymbol{x}_i + b), 0)$$

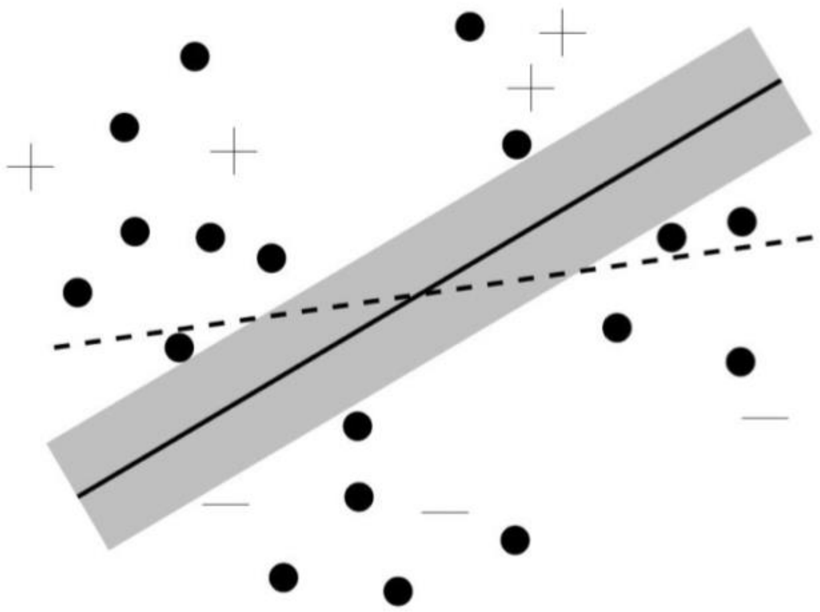
Questions of the last lecture

- What does hat loss in SVM for semi-supervised learning penalize?

→ Hat loss penalizes the case that the classifier f(w, x) does not decide any class of unlabeled x with the score margin of |f(w, x)|>=1.



(b) the hat loss



Questions of the last lecture

- Why can't we use eigenvectors to solve MinCut-based SSL in graph?

→ It is because the eigenvector solution can not consider the labeled data.

Transductive SSL with graph: fixing $f(\boldsymbol{x}_i)$ for $i \in \mathcal{L}$

$$\min_{f \in \{\pm 1\}^{n_l + n_u}} \lambda \sum_{i,j=1}^{n_l+n_u} w_{ij} \left(f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j) \right)^2 + \infty \sum_{i}^{n_l} (f(\boldsymbol{x}_i) - y_i)^2$$

Solution:

An integer program: NP hard

Can we use eigenvectors? No. Why?

We need a better way to reflect the confidence.

Questions of the last lecture

- What is the meaning of harmonic function in SSL?

$\rightarrow$ The harmonic function makes the label of a node to be harmonious(similar) with those of its neighboring nodes.

Relaxation: Transductive SSL with graph: fixing $f(\boldsymbol{x}_i)$ for $i \in \mathcal{L}$

$$\min_{f \in \boldsymbol{R}^{n_l+n_u}} \lambda \sum_{i,j=1}^{n_l+n_u} w_{ij} \left(f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j) \right)^2 + \infty \sum_{i}^{n_l} (f(\boldsymbol{x}_i) - y_i)^2$$

Naïve Solution

Right term solution: constrain f to match the supervised data

$$f(\boldsymbol{x}_i) = y_i \quad \forall\, i \in \{1, \ldots, n_l\}$$

Left term solution: enforce the solution f to be harmonic (cf. aggregation, rw)

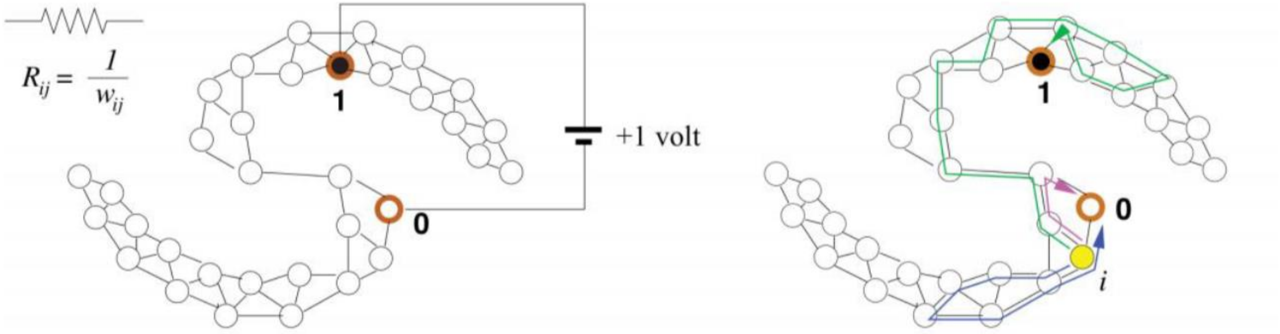
$$f(\boldsymbol{x}_i) = \frac{\sum_{ij} f(\boldsymbol{x}_j) w_{ij}}{\sum_{ij} w_{ij}} \quad \forall\, i \in \{n_l + 1, \ldots, n_l + n_u\}$$

$$\boldsymbol{f}_u = \boldsymbol{L}_{uu}^{-1}(-\boldsymbol{L}_{ul}\boldsymbol{f}_l) = \boldsymbol{L}_{uu}^{-1}(\boldsymbol{W}_{ul}\boldsymbol{f}_l)$$

Questions of the last lecture

- What is random work interpretation of harmonic function-based SSL in graph?

→ The label of a node is assigned by the average of the harmonic labels of the vertices that are hit by random works.



(a) The electric network interpretation (b) The random walk interpretation

Random walk interpretation :

1) start from the vertex you want to label and randomly walk
2) $P(j|i) = \frac{w_{ij}}{\sum_k w_{ik}} \Leftrightarrow \boldsymbol{P} = \boldsymbol{D}^{-1}\boldsymbol{W}$
3) finish when a labeled vertex is hit

$$f(\boldsymbol{x}_i) = \frac{\sum_{ij} f(\boldsymbol{x}_j) w_{ij}}{\sum_{ij} w_{ij}}$$

4) $f(\boldsymbol{x}_i)$ is assigned by average of the labels of the hit vertices.

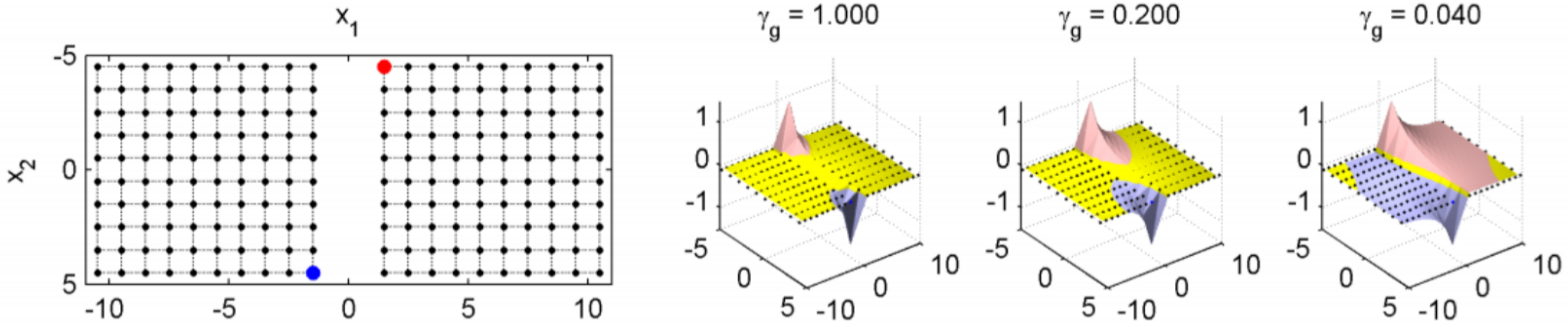
Questions of the last lecture

- What is a key point of regularized harmonic function-based SSL in graph?

→ A sink node is added to allow the random work to die at any nodes, which reduces the misleading by outliers.

$$\boldsymbol{f}_u = (\boldsymbol{L}_{uu} + \gamma_g \mathbf{I})^{-1}(\boldsymbol{W}_{ul}\boldsymbol{f}_l),$$

How does γ_g influence the solution?



Questions of the last lecture

- What is a key point of soft harmonic function-based SSL in graph?

→ The labeled data is not constrained strictly, where noisy labels may be smoothed by the soft harmonic function

Regularized HS objective with $\boldsymbol{Q} = \boldsymbol{L} + \gamma_g \mathbf{I}$:

Define $f_i \triangleq f(\boldsymbol{x}_i)$, $\boldsymbol{f} \triangleq [f_i, \dots, f_{n_l+n_u}]$

$$\min_{\boldsymbol{f} \in \mathbb{R}^{n_l+n_u}} \infty \sum_{i=1}^{n_l} (f_i - y_i)^2 + \lambda \boldsymbol{f}^T \boldsymbol{Q} \boldsymbol{f}$$

Soft constraints for $f(\boldsymbol{x}_i) = y_i, \ \forall i \in \mathcal{L}$: ∞ is replaced by finite values

$$\min_{\boldsymbol{f} \in \mathbb{R}^{n_l+n_u}} (\boldsymbol{f} - \boldsymbol{y})^T \boldsymbol{C} (\boldsymbol{f} - \boldsymbol{y}) + \boldsymbol{f}^T \boldsymbol{Q} \boldsymbol{f}$$

Outline of Lecture (1)

- 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 (out of sample extension)
  - SSL with Graph using Laplacian SVMs
  - SSL with Graph using Max-Margin Graph Cuts
  - Online SSL
  - 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

SSL with Graphs: Out of sample extension

Both MinCut and HF only inferred the labels on unlabeled data.
They are transductive.

What if a new point $x_{n_l+n_u+1}$ arrives? (called out of sample extension)

Option 1) Add it to the graph and recompute HF Solution. (Still Transductive)
Option 2) Make the algorithms inductive!

Define a classifier; $f : \mathcal{X} \rightarrow \mathbb{R}$
Make $f(\boldsymbol{x}_i)$ be smooth. Why? To deal with noise by providing reasonable interpolation for new samples.

Solution: Manifold Regularization

SSL with Graphs: Manifold Regularization

General (S)SL objective:

$$\min_{f \in \mathcal{H}} \sum_{i}^{n_l} \Phi(\boldsymbol{x}_i, y_i, f(x_i))) + \lambda\Omega(\boldsymbol{f}), \boldsymbol{f} \triangleq [\dots f(x_i) \dots]$$

Want to control f , also for the out-of-sample data, i.e., everywhere(generalization).

$$\lambda\Omega(\boldsymbol{f}) = \lambda_2 \boldsymbol{f}^T \boldsymbol{L} \boldsymbol{f} + \lambda_1 \|f\|_{\mathcal{K}}^2$$

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

For general kernels:

Smoothness for given samples

$$\min_{f \in \mathcal{H}_{\mathcal{K}}} \sum_{i}^{n_l} \Phi(\boldsymbol{x}_i, y_i, f(x_i))) + \lambda_1 \boxed{\|f\|_{\mathcal{K}}^2} + \lambda_2 \boxed{\boldsymbol{f}^T \boldsymbol{L} \boldsymbol{f}}$$

Smoothness for unknown samples

SSL with Graphs: Manifold Regularization

General (S)SL objective with kernels:

$$\min_{f \in \mathcal{H}_{\mathcal{K}}} \sum_{i}^{n_l} \Phi(\boldsymbol{x}_i, y_i, f(x_i))) + \lambda_1 \|f\|_{\mathcal{K}}^2 + \lambda_2 \boldsymbol{f}^T \boldsymbol{L} \boldsymbol{f}$$

Representer theorem for manifold regularization

The minimizer f^* has a finite expansion of the form

$$f^*(\boldsymbol{x}) = \sum_{i=1}^{n_l+n_u} \alpha_i^* \mathcal{K}(\boldsymbol{x}, \boldsymbol{x}_i)$$

LapRLS: Laplacian Regularized Least Squares

$$\Phi\big(\boldsymbol{x}, y, f(\boldsymbol{x})\big) = (y - f(\boldsymbol{x}))^2$$

LapSVM: Laplacian Support Vector Machines

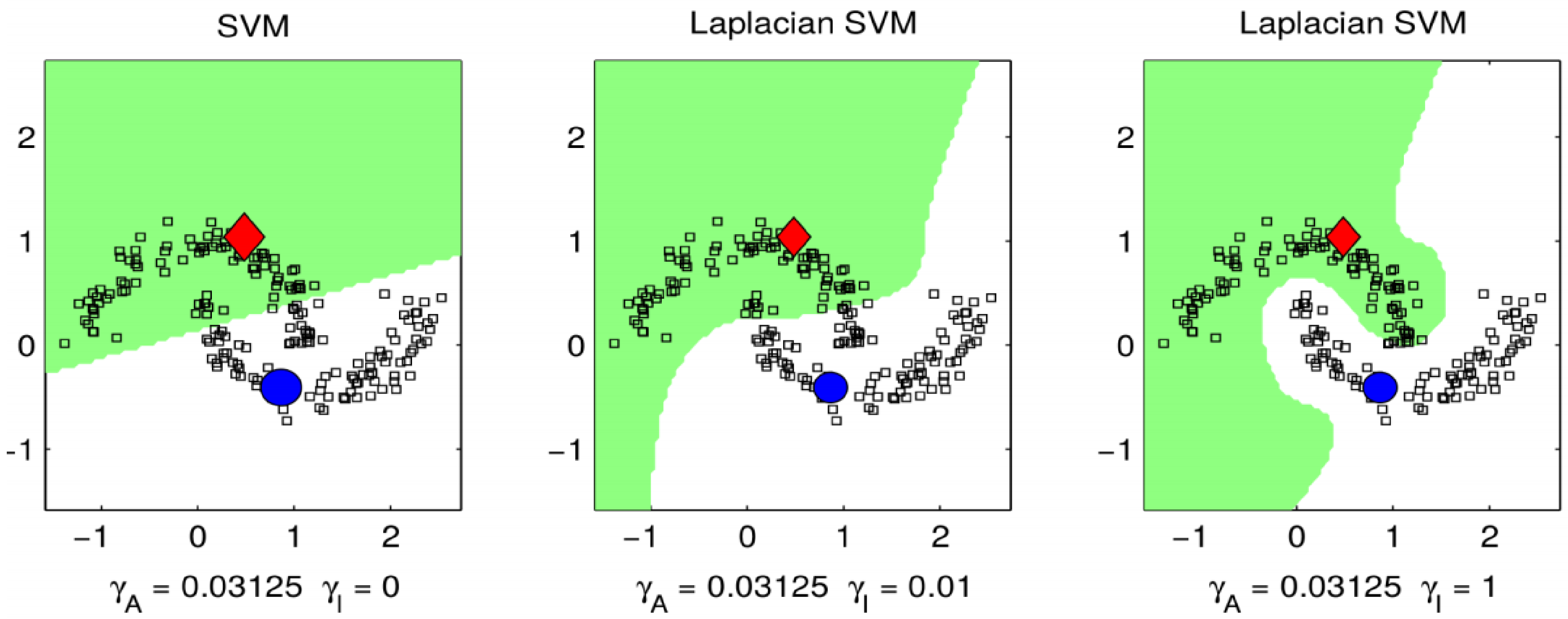
$$\Phi\big(\boldsymbol{x}, y, f(\boldsymbol{x})\big) = \max(0, 1 - yf(\boldsymbol{x}))$$

SSL with Graphs: Laplacian SVMs

General (S)SL objective with kernels:

$$\min_{f\in\mathcal{H}_{\mathcal{K}}} \sum_{i}^{n_l} \max(0, 1 - y_i f(x_i)) + \lambda_A \|f\|_{\mathcal{K}}^2 + \lambda_l \boldsymbol{f}^T \boldsymbol{L} \boldsymbol{f}$$

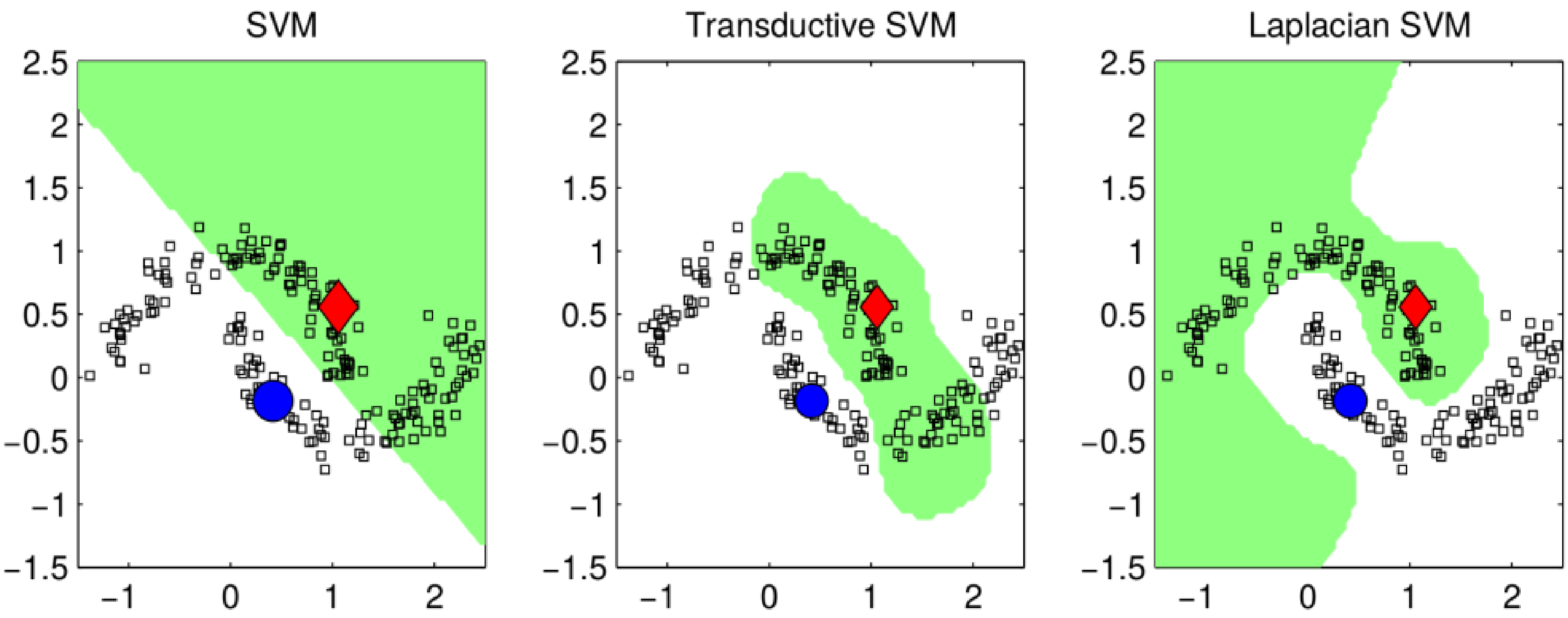
RBF kernels



SSL with Graphs: Laplacian SVMs

Formulation for **Transductive SVM** (Revisit)

$$\min_{\boldsymbol{w},b} \sum_{i=1}^{n_l} \max(1 - y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b), 0) + \lambda_1 \|\boldsymbol{w}\|^2$$
$$+\lambda_2 \sum_{i=n_l+1}^{n_l+n_u} \max(1 - |\boldsymbol{w}^T \boldsymbol{x}_i + b|, 0)$$



SSL with Graphs: Max-Margin Graph Cuts

Self-training with the confident data

$$f^* = \underset{f \in \mathcal{H}_{\mathcal{K}}}{argmin} \sum\nolimits_{i:|\ell_i^*| \geq \varepsilon} \Phi(\boldsymbol{x}_i, sgn(\ell_i^*), f(\boldsymbol{x}_i)) + \lambda \|f\|_{\mathcal{K}}^2$$

s. t. $\quad \ell^* = \underset{\ell \in \mathbb{R}^N}{argmin}\, \ell^T \left(\boldsymbol{L} + \gamma_g \mathbf{I}\right) \ell$

s. t. $\quad \ell_i = y_i, \quad \forall\, i = 1, \dots, n_l$

Representer theorem is still cool:

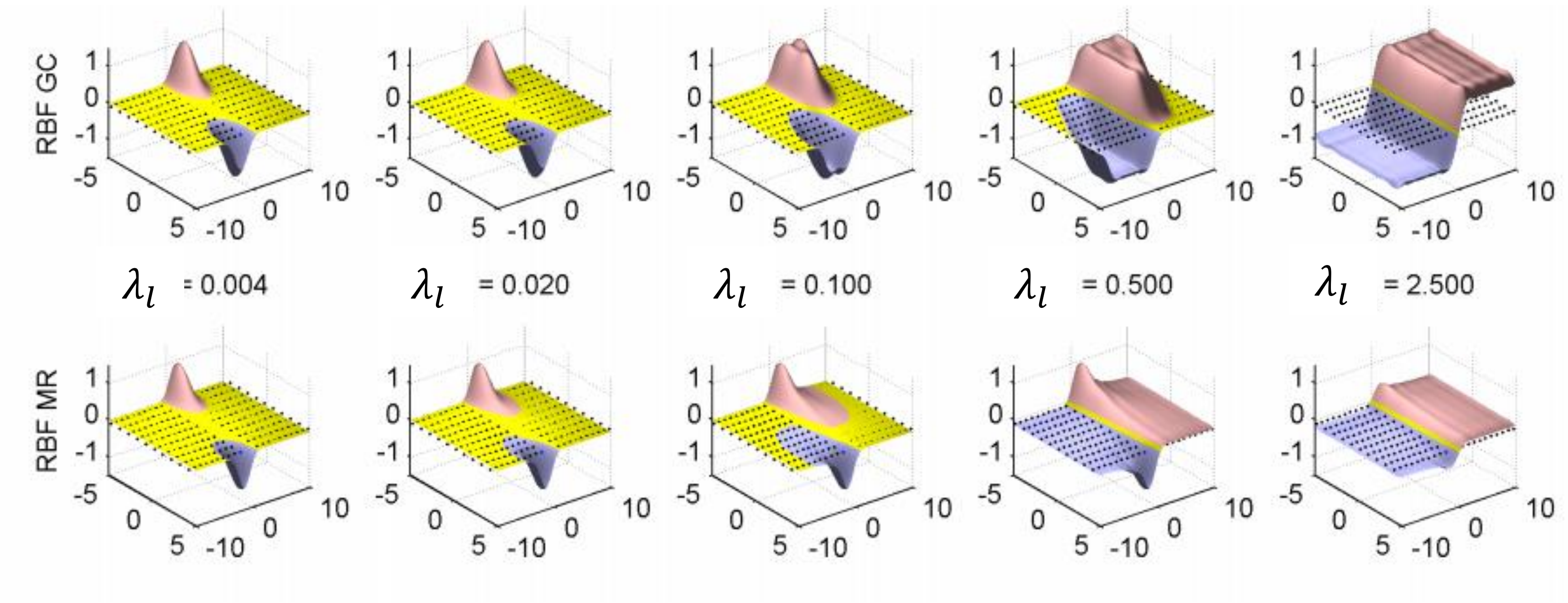
$$f^*(\boldsymbol{x}) = \sum\nolimits_{i:|\ell_i^*| \geq \varepsilon} \alpha_i^* \mathcal{K}(\boldsymbol{x}, \boldsymbol{x}_i)$$

SSL with Graphs: LapSVMs and MM Graph Cuts

$$\min_{f\in\mathcal{H}_{\mathcal{K}}} \sum_{i}^{n_l} \max(0, 1 - y_i f(x_i)) + \lambda_A \|f\|_{\mathcal{K}}^2 + \lambda_l \boldsymbol{f}^T \boldsymbol{L} \boldsymbol{f}$$

$\lambda_A = 0.1, \qquad \epsilon = 0.01$

MMGC and MR for 2D data and RBF $\mathcal{K}$



Manifold regularization of SVMs (MR), max-margin graph cuts (GC)

[MM Graph Cuts](), JMLR, 2010

$$OnlineSSL(\mathcal{G})$$

when we can't access future x

Online SSL with Graphs

Offline learning setup

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 $\{y_i | y_i = \boldsymbol{f}(\boldsymbol{x}_i), i = n+1, \dots, N\}$ well beyond that (**inductive**).

Online learning setup

At the beginning, given $\{\boldsymbol{x}_i, y_i\}_{i=1}^{n}$ from $\mathbb{R}^d$.

At time t:

receive $\boldsymbol{x}_t$

predict y_t

Revisit : out of sample expansion
Option 1) Add it to the graph and recompute HF Solution.
Option 2) Make the algorithms inductive! (not learn $\boldsymbol{x}_t$)

Online SSL with Graphs

Online HFS: Straightforward solution (option 1)

1: **while** new unlabeled example $\boldsymbol{x}_t$ comes **do**
2: Add $\boldsymbol{x}_t$ to the graph $G(\boldsymbol{W})$
3: Update $\boldsymbol{L}_t$
4: Infer labels

$$\boldsymbol{f}_u = (\boldsymbol{L}_{uu} + \gamma_g \mathbf{I})^{-1}(\boldsymbol{W}_{ul}\boldsymbol{f}_l)$$

5: Predict $\hat{y}_t = sgn(\boldsymbol{f}_{u,t})$
6: **end while**

What is wrong with this solution?

The cost and memory of the operations.

What can we do?

Online SSL with Graphs: Graph Quantization

Let's keep only k vertices!

Limit memory to k centroids with $\widetilde{\boldsymbol{W}}^q$, where $\widetilde{\boldsymbol{W}}^q_{ij}$ contains the similarity between the i-th and j-th centroids.
Each centroid represents several others.

Let $\boldsymbol{V}$ be a diagonal matrix of which
$\boldsymbol{V}_{ii}$ denotes number of points collapsed into the i-th centroid.

Can we compute it compactly? Compact harmonic solution.

$$\boldsymbol{f}_u^q = (\boldsymbol{L}_{uu}^q + \gamma_g \boldsymbol{V})^{-1}\left(\boldsymbol{W}_{ul}^q \boldsymbol{f}_l\right) \text{ where } \boldsymbol{W}^q = \boldsymbol{V}\widetilde{\boldsymbol{W}}^q\boldsymbol{V}$$

Proof and Algorithm: see http://www.bkveton.com/docs/uai2010a.pdf

Online SSL with Graphs

Online HFS with Graph Quantization

Input
    k number of representative nodes
Initialization
    V matrix of multiplicities with 1 on diagonal
while new unlabeled example $\boldsymbol{x}_t$ comes **do**
    Add $\boldsymbol{x}_t$ to graph G
    if # nodes > k **then**
        quantize G
    end if
    Update L_t of $G(\boldsymbol{VWV})$
    Infer labels
    Predict $\hat{y}_t = sgn(f_u(t))$
end while

Online SSL with Graphs: Graph Quantization

Incremental quantize G

An idea: incremental k-centers

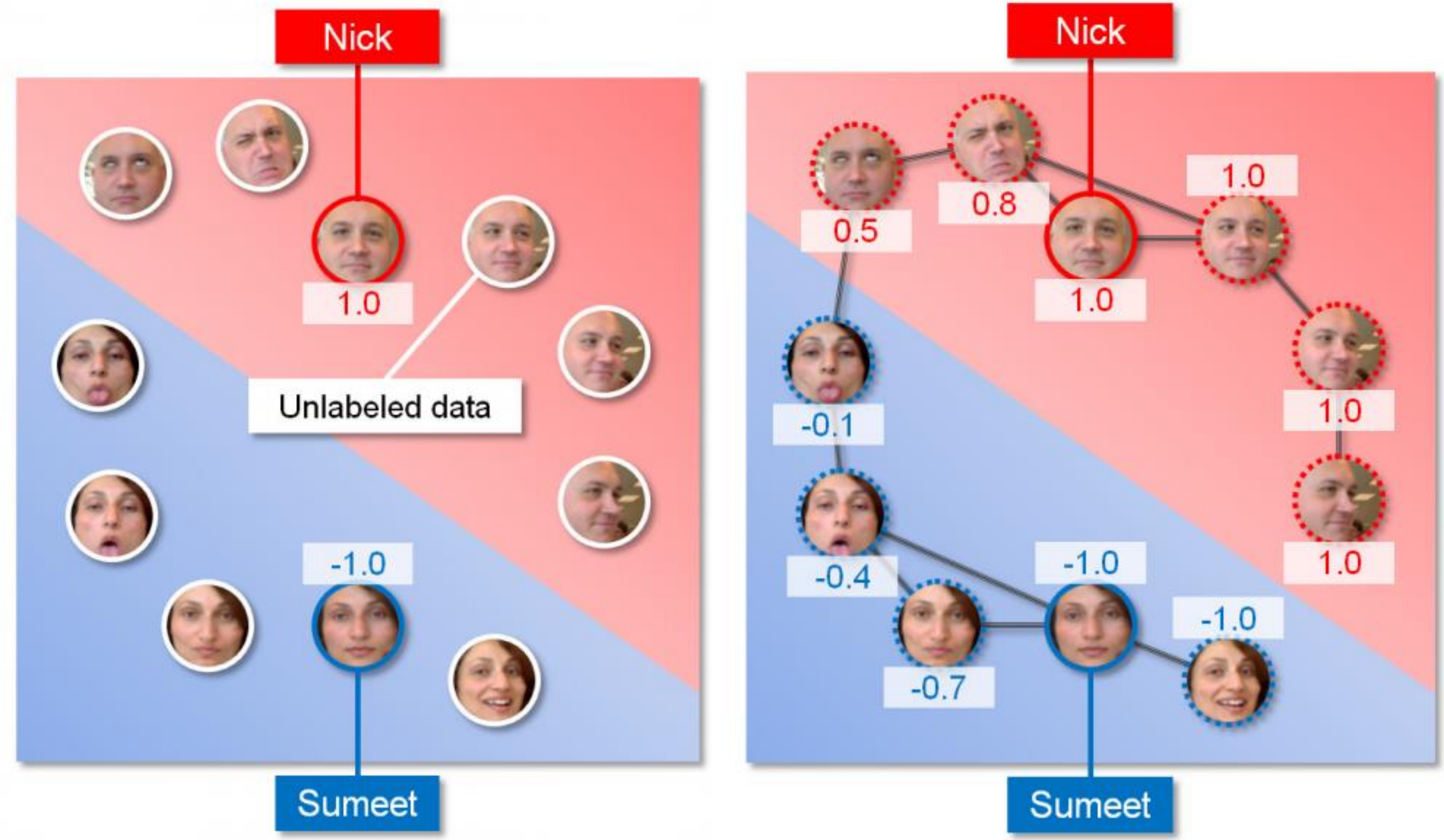
Doubling algorithm of Charikar et al. [Cha+97]

Keeps up to k centers $C_t = \{c_1, c_2, \ldots\}$ with

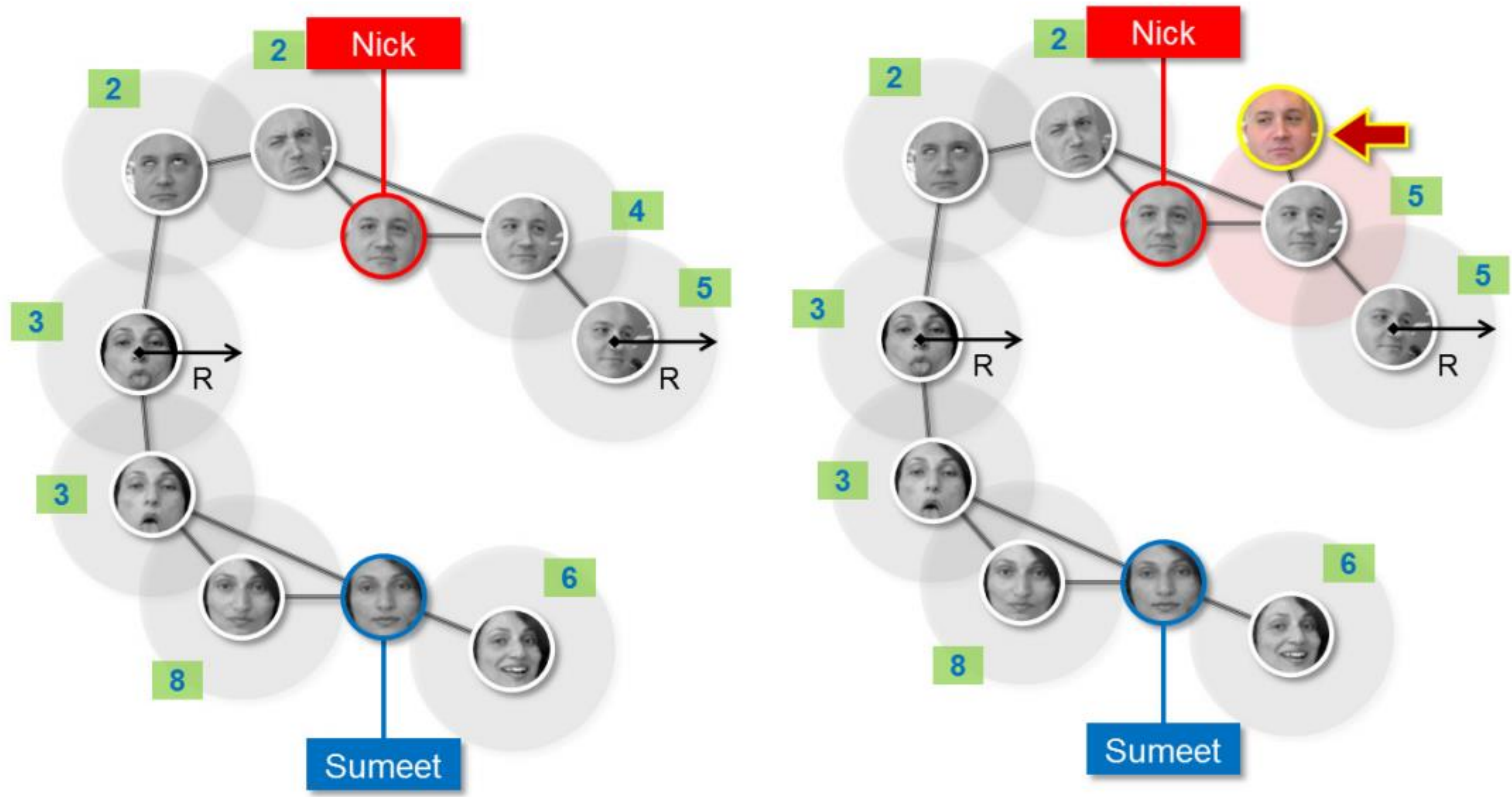
- Distance $c_i, c_j \in C_t$ is at least $\geq R$
- For each new $\boldsymbol{x}_t$, distance to some $c_i \in C_t$ is less than R.
- $|C_t| \leq k$
- if not possible, R is doubled

Online SSL with Graphs: Graph Quantization

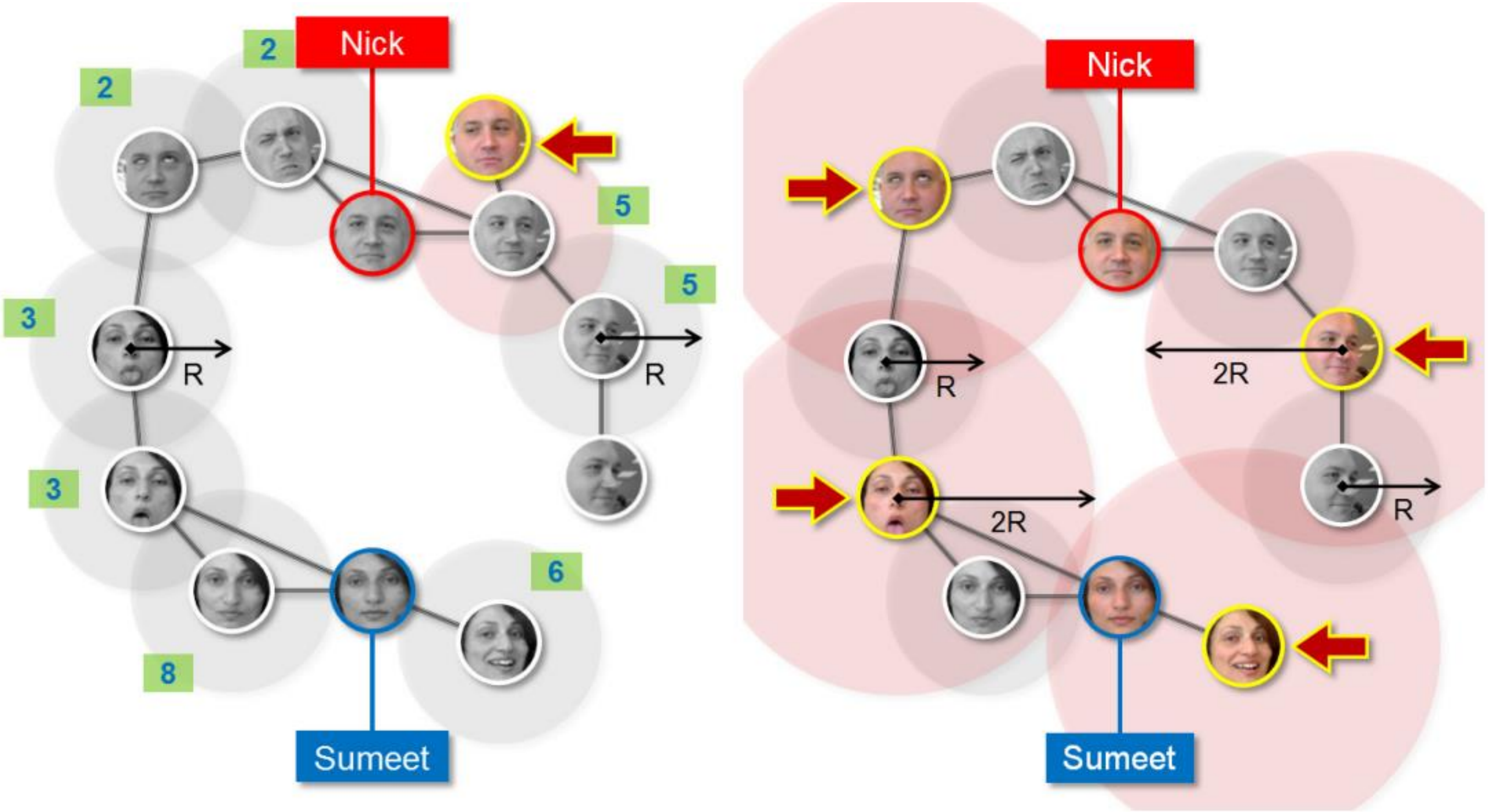
Online SSL with Graphs: Graph Quantization



Online SSL with Graphs: Graph Quantization

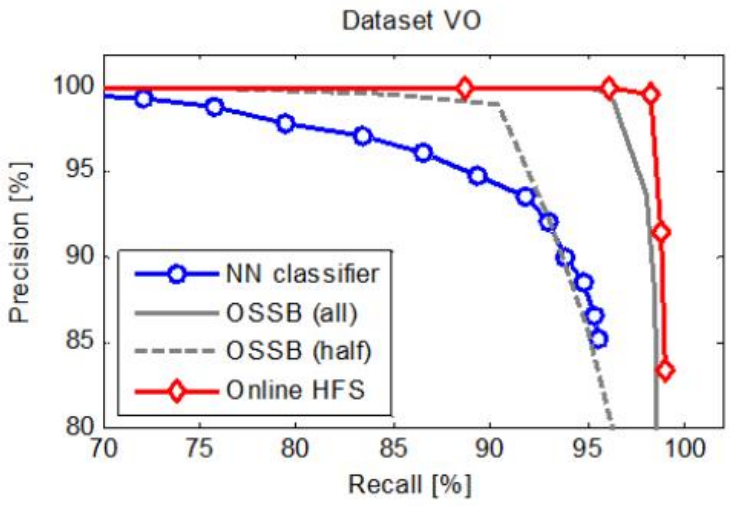


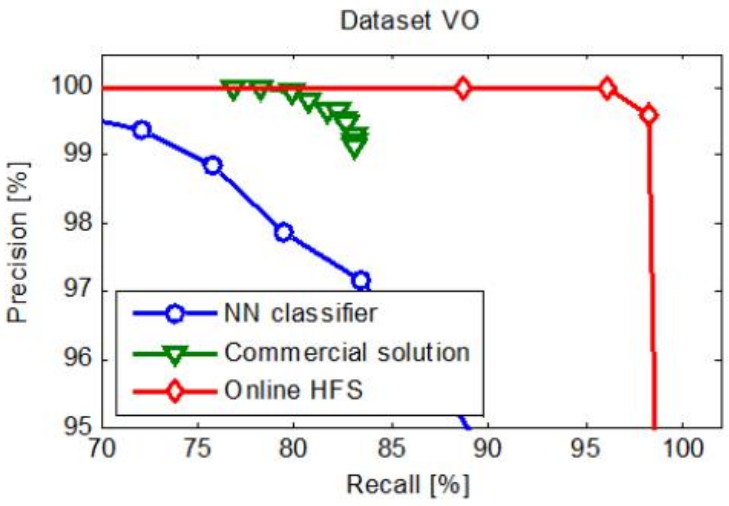
Online SSL with Graphs: Graph Quantization



Online SSL with Graphs: Some experimental results

http://www.bkveton.com/videos/Ad.mp4





Online HFS outperforms OSSB (even when the weak learners are chosen using future data)

Online HFS yields better results than a commercial solution at 20% of the computational cost

Summary Questions of the Lecture

What are the two options for out of sample extension in SSL?

Why do we have to make a classifier be smooth in inductive SSL for out of sample extension?

What is the meaning of manifold regularization?

What is the key idea of Max-Margin Graph Cuts for SSL?

What are the two options for online SSL?

What is the key idea for keeping # of representative nodes?