Graphs in Machine Learning

Michal Valko

INRIA Lille - Nord Europe, France

Partially based on material by: Mikhail Belkin, Branislav Kveton

Previous Lecture

- Manifold learning with Laplacian Eigenmaps
- Semi-Supervised Learning
  - Why and when it helps?
  - Self-training
  - Semi-supervised SVMs
- Graph-based semi-supervised learning
- SSL with MinCuts
- Gaussian random fields and harmonic solution
- Regularization of harmonic solution
- Soft-harmonic solution

This Lecture

- Inductive and transductive semi-supervised learning
- Manifold regularization
- Theory of Laplacian-based manifold methods
- Transductive learning stability based bounds
- SSL Learnability
- Online Semi-Supervised Learning
- Online incremental k-centers

Previous Lab Session

- 3. 2. 2015 by Daniele.Calandriello@inria.fr
- Content
  - Graph Construction
  - Test sensitivity to parameters: σ, k, ε
  - Spectral Clustering
  - Spectral Clustering vs. k-means
  - Image Segmentation
- Short written report (graded, each lab around 5% of grade)
- Questions to Daniele.Calandriello@inria.fr
- *Deadline:* 17. 2. 2015 Today!
- http://researchers.lille.inria.fr/~calandri/ta/graphs/td1_handout.pdf

Next Lab Session

- 24. 2. 2015 by Daniele.Calandriello@inria.fr
- Content
  - Semi-supervised learning
  - Graph quantization
  - Online face recognizer
  - 3 volunteers (Linux, Max, Windows)
  - Install OpenCV (instructions: few days before the lab)
  - record a video with faces
- Short written report (graded, each lab around 5% of grade)
- Questions to Daniele.Calandriello@inria.fr
- *Deadline:* 10. 3. 2015
- http://researchers.lille.inria.fr/~calandri/ta/graphs/td2_handout.pdf

Final Class projects

- preferred option: you come up with the topic
- list of suggested topics from March
- theory/implementation/review or a combination
- one or two people per project (exceptionally three)
- grade: report + short presentation of the team
- deadlines soon

Advanced Learning for Text and Graph Data

Time: Wednesdays 8h30-11h30 — **4 lectures** and **3 Labs**
Place: Polytechnique / Amphi Sauvy
Lecturer 1: Michalis Vazirgiannis (Polytechnique)
Lecturer 2: Yassine Faihe (Hewlett-Packard - Vertica)

ALTeGraD and **Graphs in ML** run in parallel
The two graph courses are coordinated to be complementary.

Some of covered graph topics not covered in this course

- Ranking algorithms and measures (Kendal Tau, NDCG)
- Advanced graph generators
- Community mining, advanced graph clustering
- Graph degeneracy (k-core & extensions)
- Privacy in graph mining

`http://www.math.ens-cachan.fr/version-francaise/formations/master-mva/contenus-/advanced-learning-for-text-and-graph-data-altegrad--239506.kjsp?RH=1242430202531`

Where we left off

Semi-supervised learning with graphs:

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

Regularized harmonic Solution:

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

Unconstrained regularization in general:

$$\mathbf{f}^\star = \min_{\mathbf{f}\in\mathbb{R}^n} (\mathbf{f}-\mathbf{y})^\mathsf{T}\mathbf{C}(\mathbf{f}-\mathbf{y}) + \mathbf{f}^\mathsf{T}\mathbf{Q}\mathbf{f}$$

SSL with Graphs: Out of sample extension

Both **MinCut** and **HFS** only inferred the labels on unlabeled data.

They are **transductive**.

What if a new point $\mathbf{x}_{n_l+n_u+1}$ arrives? also called out-of-sample extension

Option 1) Add it to the graph and recompute HFS.

Option 2) Make the algorithms **inductive**!

Allow to be defined everywhere: $f : \mathcal{X} \mapsto \mathbb{R}$
Allow $f(\mathbf{x}_i) \neq y_i$. Why? To deal with noise.

Solution: **Manifold Regularization**

SSL with Graphs: Manifold Regularization

General (S)SL objective:

$$\min_f \sum_i^{n_l} V(\mathbf{x}_i, y_i, f(\mathbf{x}_i)) + \lambda \Omega(f)$$

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

$$\Omega(f) = \lambda_2 \mathbf{f}^\mathsf{T} \mathbf{L} \mathbf{f} + \lambda_1 \int_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x})^2 \, \mathrm{d}x$$

For general **kernels**:

$$\min_{f \in \mathcal{H}_\mathcal{K}} \sum_i^{n_l} V(\mathbf{x}_i, y_i, f(\mathbf{x}_i)) + \lambda_1 \|f\|_\mathcal{K} + \lambda_2 \mathbf{f}^\mathsf{T} \mathbf{L} \mathbf{f}$$

SSL with Graphs: Manifold Regularization

$$f^\star = \arg\min_{f \in \mathcal{H}_\mathcal{K}} \sum_i^{n_l} V(\mathbf{x}_i, y_i, f) + \lambda_1 \|f\|_\mathcal{K} + \lambda_2 \mathbf{f}^\mathsf{T}\mathbf{L}\mathbf{f}$$

Representer Theorem for Manifold Regularization

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

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

$$V(\mathbf{x}, y, f) = (y - f(\mathbf{x}))^2$$

LapRLS Laplacian Regularized Least Squares

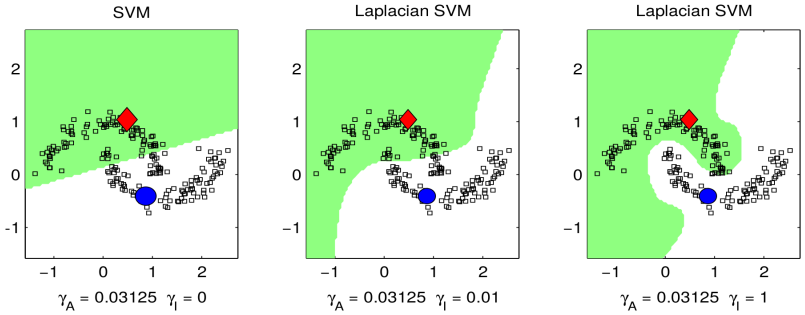
$$V(\mathbf{x}, y, f) = \max(0, 1 - yf(\mathbf{x}))$$

LapSVM Laplacian Support Vector Machines

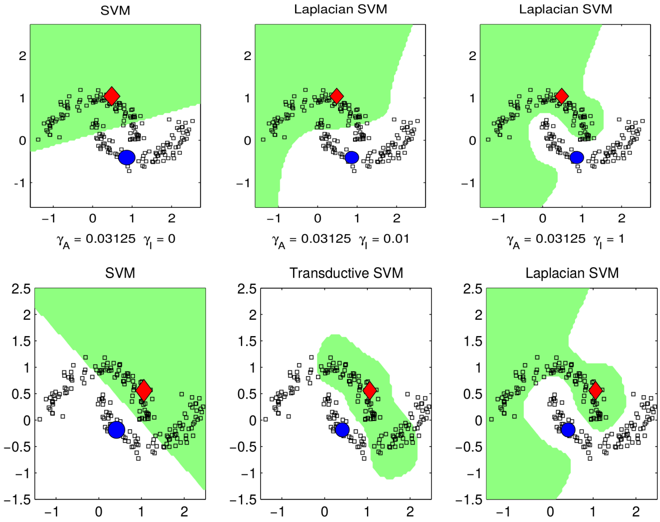
SSL with Graphs: Laplacian SVMs

$$f^{\star} = \underset{f \in \mathcal{H}_{\mathcal{K}}}{\arg\min} \sum_{i}^{n_l} \max\left(0, 1 - yf\left(\mathbf{x}\right)\right) + \lambda_1 \|f\|_{\mathcal{K}} + \lambda_2 \mathbf{f}^{\mathsf{T}} \mathbf{L} \mathbf{f}$$

Allows us to learn a function in **RKHS**, i.e., **RBF** kernels.



SSL with Graphs: Laplacian SVMs



SSL with Graphs: Laplacian SVMs

$$f^\star = \underset{f \in \mathcal{H}_\mathcal{K}}{\arg\min} \sum_i^{n_l} \max\left(0, 1 - yf\left(\mathbf{x}\right)\right) + \lambda_1 \|f\|_\mathcal{K} + \lambda_2 \mathbf{f}^\mathsf{T}\mathbf{L}\mathbf{f}$$

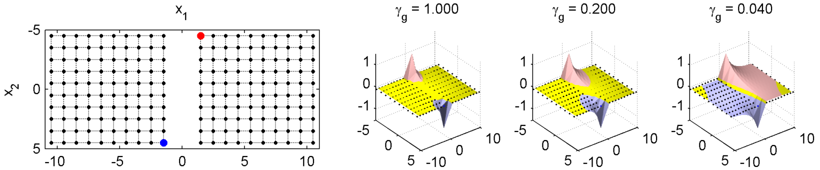
$\mathcal{H}_\mathcal{K}$ is nice and expressive.

Can there be a problem with certain $\mathcal{H}_\mathcal{K}$?

We look for f only in $\mathcal{H}_\mathcal{K}$.

If it is simple (e.g., **linear**) minimization of $\mathbf{f}^\mathsf{T}\mathbf{L}\mathbf{f}$ can perform badly.

Consider again this 2D data and linear $\mathcal{K}$.



SSL with Graphs: Laplacian SVMs

Linear $\mathcal{K} \equiv$ functions with slope α_1 and intercept α_2.

$$\min_{\alpha_1,\alpha_2} \sum_{i}^{n_l} V(f, \mathbf{x}_i, y_i) + \lambda_1 \left[\alpha_1^2 + \alpha_2^2\right] + \lambda_2 \mathbf{f}^\mathsf{T}\mathbf{L}\mathbf{f}$$

For this simple case we can write down $\mathbf{f}^\mathsf{T}\mathbf{L}\mathbf{f}$ explicitly.

$$\begin{aligned}
\mathbf{f}^\mathsf{T}\mathbf{L}\mathbf{f} &= \frac{1}{2}\sum_{i,j} w_{ij}(f(\mathbf{x}_i) - f(\mathbf{x}_j))^2 \\
&= \frac{1}{2}\sum_{i,j} w_{ij}(\alpha_1(\mathbf{x}_{i1} - \mathbf{x}_{j1}) + \alpha_2(\mathbf{x}_{i2} - \mathbf{x}_{j2}))^2 \\
&= \frac{\alpha_1^2}{2}\underbrace{\sum_{i,j} w_{ij}(\mathbf{x}_{i1} - \mathbf{x}_{j1})^2}_{\Delta=218.351} + \frac{\alpha_2^2}{2}\underbrace{\sum_{i,j} w_{ij}(\mathbf{x}_{i2} - \mathbf{x}_{j2})^2}_{\Delta=218.351}
\end{aligned}$$

SSL with Graphs: Laplacian SVMs

2D data and linear $\mathcal{K}$ objective

$$\min_{\alpha_1,\alpha_2} \sum_i^{n_l} V(f, \mathbf{x}_i, y_i) + \left(\lambda_1 + \frac{\lambda_2 \Delta}{2}\right) [\alpha_1^2 + \alpha_2^2]$$

Setting $\lambda^\star = \left(\lambda_1 + \frac{\gamma_2 \Delta}{2}\right)$:

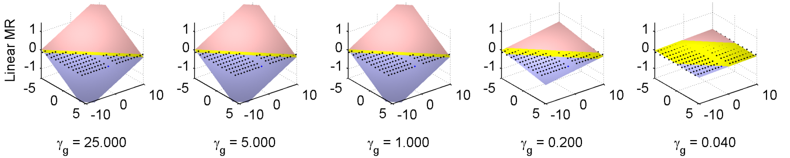
$$\min_{\alpha_1,\alpha_2} \sum_i^{n_l} V(f, \mathbf{x}_i, y_i) + \lambda^\star [\alpha_1^2 + \alpha_2^2]$$

The only influence of unlabeled data is through $\lambda^\star$.

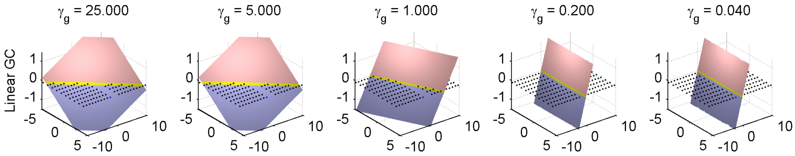
The same value of the objective as for supervised learning for some λ **without the unlabeled data**! This is not good.

SSL with Graphs: Laplacian SVMs

MR for 2D data and linear $\mathcal{K}$ only changes the slope



What would we like to see?



We use the unlabeled data **before** optimizing $\mathcal{H}_{\mathcal{K}}$!

SSL with Graphs: Max-Margin Graph Cuts

Let's take the confident data and use them as true!

$$f^\star = \min_{f \in \mathcal{H}_\mathcal{K}} \sum_{i : |\ell_i^\star| \geq \varepsilon} V(f, \mathbf{x}_i, \mathrm{sgn}(\ell_i^\star)) + \gamma \|f\|_\mathcal{K}^2$$

$$\text{s.t.} \quad \boldsymbol{\ell}^\star = \arg\min_{\boldsymbol{\ell} \in \mathbb{R}^n} \boldsymbol{\ell}^\mathsf{T}(\mathbf{L} + \gamma_g \mathbf{I})\boldsymbol{\ell}$$

$$\text{s.t. } \ell_i = y_i \text{ for all } i = 1, \dots, n_l$$

Wait, but this is what we did not like in self-training!

Will we get into the same trouble?

Representer theorem still cool:

$$f^\star(\mathbf{x}) = \sum_{i : |f_i^\star| \geq \varepsilon} \alpha_i^\star \mathcal{K}(\mathbf{x}_i, \mathbf{x})$$

SSL with Graphs: Generalization Bounds

Why is this not a witchcraft? We take GC as an example. MR or HFS are similar.

What kind of guarantees we want?

We may want to bound the **risk**

$$R_P(f) = \mathbb{E}_{P(\mathbf{x})}\left[\mathcal{L}\left(f\left(\mathbf{x}\right), y\left(\mathbf{x}\right)\right)\right]$$

for some **loss**, e.g., 0/1 loss

$$\mathcal{L}(y', y) = \mathbb{1}\{\text{sgn}(y') \neq y\}$$

What makes sense to bound $R_P(f)$ with?

empirical risk + **error terms**

SSL with Graphs: Generalization Bounds

True risk vs. empirical risk

$$R_P(\mathbf{f}^\star) = \frac{1}{n}\sum_i (f_i^\star - y_i)^2$$

$$\widehat{R}_P(\mathbf{f}^\star) = \frac{1}{n_l}\sum_{i\in l} (f_i^\star - y_i)^2$$

We look for the bound in the form

$$R_P(\mathbf{f}^\star) \leq \widehat{R}_P(\mathbf{f}^\star) + \text{errors}$$

$$\text{errors} = \textbf{transductive} + \textbf{inductive}$$

SSL with Graphs: Generalization Bounds

Bounding inductive error (using classical SLT tools)

With probability $1 - \eta$, using Equations 3.15 and 3.24 [Vap95]

$$R_P(f) \leq \frac{1}{n} \sum_i \mathcal{L}(f(\mathbf{x}_i), y_i) + \Delta_I(h, n, \eta).$$

$n \equiv$ number of samples , $h \equiv$ VC dimension of the class

$$\Delta_I(h, n, \eta) = \frac{h(\ln(2n/h) + 1) - \ln(\eta/4)}{n}$$

How to bound $\mathcal{L}(f(\mathbf{x}_i), y_i)$? For any $y_i \in \{-1, 1\}$ and $\ell_i^\star$

$$\mathcal{L}(f(\mathbf{x}_i), y_i) \leq \mathcal{L}(f(\mathbf{x}_i), \mathrm{sgn}(\ell_i^\star)) + (\ell_i^\star - y_i)^2.$$

SSL with Graphs: Generalization Bounds

Bounding transductive error (using stability analysis)

http://www.cs.nyu.edu/~mohri/pub/str.pdf

How to bound $(\ell_i^\star - y_i)^2$?

Bounding $(\ell_i^\star - y_i)^2$ for hard case difficult $\rightarrow$ we bound soft HFS:

$$\boldsymbol{\ell}^\star = \min_{\boldsymbol{\ell} \in \mathbb{R}^n} (\boldsymbol{\ell} - \mathbf{y})^\mathsf{T} \mathbf{C} (\boldsymbol{\ell} - \mathbf{y}) + \boldsymbol{\ell}^\mathsf{T} \mathbf{Q} \boldsymbol{\ell}$$

Closed form solution

$$\boldsymbol{\ell}^\star = \left(\mathbf{C}^{-1}\mathbf{Q} + \mathbf{I}\right)^{-1} \mathbf{y}$$

SSL with Graphs: Generalization Bounds

Bounding transductive **error**

$$\ell^\star = \min_{\boldsymbol{\ell}\in\mathbb{R}^n} (\boldsymbol{\ell} - \mathbf{y})^\mathsf{T}\mathbf{C}(\boldsymbol{\ell} - \mathbf{y}) + \boldsymbol{\ell}^\mathsf{T}\mathbf{Q}\boldsymbol{\ell}$$

Think about stability of this solution.

Consider two datasets differing in exactly one *labeled* point.

$\mathcal{C}_1 = \mathbf{C}_1^{-1}\mathbf{Q} + \mathbf{I}$ and $\mathcal{C}_2 = \mathbf{C}_2^{-1}\mathbf{Q} + \mathbf{I}$

What is the maximal difference in the solutions?

$$\begin{aligned}\boldsymbol{\ell}_2^\star - \boldsymbol{\ell}_1^\star &= \mathcal{C}_2^{-1}\mathbf{y}_2 - \mathcal{C}_1^{-1}\mathbf{y}_1 \\ &= \mathcal{C}_2^{-1}(\mathbf{y}_2 - \mathbf{y}_1) - \left(\mathcal{C}_2^{-1} - \mathcal{C}_1^{-1}\right)\mathbf{y}_1 \\ &= \mathcal{C}_2^{-1}(\mathbf{y}_2 - \mathbf{y}_1) - \left(\mathcal{C}_1^{-1}\left[\left(\mathbf{C}_1^{-1} - \mathbf{C}_2^{-1}\right)\mathbf{Q}\right]\mathcal{C}_2^{-1}\right)\mathbf{y}_1\end{aligned}$$

Note that $\mathbf{v} \in \mathbb{R}^{n\times 1}$, $\lambda_m(A)\|\mathbf{v}\|_2 \le \|A\mathbf{v}\|_2 \le \lambda_M(A)\|\mathbf{v}\|_2$

$$\|\boldsymbol{\ell}_2^\star - \boldsymbol{\ell}_1^\star\|_2 = \frac{\|\mathbf{y}_2 - \mathbf{y}_1\|_2}{\lambda_m(\mathcal{C}_2)} + \frac{\lambda_M(\mathbf{Q})\|\mathbf{C}_1^{-1} - \mathbf{C}_2^{-1}\|_2 \cdot \|\mathbf{y}_1\|_2}{\lambda_m(\mathcal{C}_2)\lambda_m(\mathcal{C}_1)}$$

SSL with Graphs: Generalization Bounds

Bounding transductive error

$$\boldsymbol{\ell}^\star = \min_{\boldsymbol{\ell}\in\mathbb{R}^n} (\boldsymbol{\ell} - \mathbf{y})^\mathsf{T}\mathbf{C}(\boldsymbol{\ell} - \mathbf{y}) + \boldsymbol{\ell}^\mathsf{T}\mathbf{Q}\boldsymbol{\ell}$$

$$\|\boldsymbol{\ell}_2^\star - \boldsymbol{\ell}_1^\star\|_2 = \frac{\|\mathbf{y}_2 - \mathbf{y}_1\|_2}{\lambda_m(\mathcal{C}_2)} + \frac{\lambda_M(\mathbf{Q})\|\mathbf{C}_1^{-1} - \mathbf{C}_2^{-1}\|_2 \cdot \|\mathbf{y}_1\|_2}{\lambda_m(\mathcal{C}_2)\lambda_m(\mathcal{C}_1)}$$

Using $\lambda_m(\mathcal{C}) \geq \frac{\lambda_m(\mathbf{Q})}{\lambda_M(\mathbf{C})} + 1$

$$\|\boldsymbol{\ell}_2^\star - \boldsymbol{\ell}_1^\star\|_2 = \frac{\|\mathbf{y}_2 - \mathbf{y}_1\|_2}{\frac{\lambda_m(\mathbf{Q})}{\lambda_M(\mathbf{C}_1)} + 1} + \frac{\lambda_M(\mathbf{Q})\|\mathbf{C}_1^{-1} - \mathbf{C}_2^{-1}\|_2 \cdot \|\mathbf{y}_1\|_2}{\left(\frac{\lambda_m(\mathbf{Q})}{\lambda_M(\mathbf{C}_2)} + 1\right)\left(\frac{\lambda_m(\mathbf{Q})}{\lambda_M(\mathbf{C}_1)} + 1\right)}$$

SSL with Graphs: Generalization Bounds

Bounding transductive error

$$\beta = \|\boldsymbol{\ell}_2^\star - \boldsymbol{\ell}_1^\star\|_2 = \frac{\|\mathbf{y}_2 - \mathbf{y}_1\|_2}{\frac{\lambda_m(\mathbf{Q})}{\lambda_M(\mathbf{C}_1)} + 1} + \frac{\lambda_M(\mathbf{Q})\|\mathbf{C}_1^{-1} - \mathbf{C}_2^{-1}\|_2 \cdot \|\mathbf{y}_1\|_2}{\left(\frac{\lambda_m(\mathbf{Q})}{\lambda_M(\mathbf{C}_2)} + 1\right)\left(\frac{\lambda_m(\mathbf{Q})}{\lambda_M(\mathbf{C}_1)} + 1\right)}$$

Now let us plug in the values for our problem.

Take $c_l = 1$ and $c_l > c_u$. We have $|y_i| \leq 1$ and $|\ell_i^\star| \leq 1$.

$$\beta = 2\left[\frac{\sqrt{2}}{\lambda_m(\mathbf{Q}) + 1} + \sqrt{2n_l}\frac{1 - \sqrt{c_u}}{\sqrt{c_u}}\frac{\lambda_M(\mathbf{Q})}{(\lambda_m(\mathbf{Q}) + 1)^2}\right]$$

$\mathbf{Q}$ is reg. $\mathbf{L}$: $\lambda_m(\mathbf{Q}) = \lambda_m(\mathbf{L}) + \gamma_g$ and $\lambda_M(\mathbf{Q}) = \lambda_M(\mathbf{L}) + \gamma_g$

$$\beta \;\leq\; 2\left[\frac{\sqrt{2}}{\gamma_g + 1} + \sqrt{2n_l}\frac{1 - \sqrt{c_u}}{\sqrt{c_u}}\frac{\lambda_M(\mathbf{L}) + \gamma_g}{\gamma_g^2 + 1}\right]$$

SSL with Graphs: Generalization Bounds

Bounding transductive error

http://web.cse.ohio-state.edu/~mbelkin/papers/RSS_COLT_04.pdf

By the generalization bound of Belkin [BMN04]

$$R_P^W(\boldsymbol{\ell}^\star) \;\leq\; \widehat{R}_P^W(\boldsymbol{\ell}^\star) + \underbrace{\beta + \sqrt{\frac{2\ln(2/\delta)}{n_l}}(n_l\beta + 4)}_{\text{transductive error } \Delta_T(\beta, n_l, \delta)}$$

$$\beta \;\leq\; 2\left[\frac{\sqrt{2}}{\gamma_g + 1} + \sqrt{2n_l}\frac{1-\sqrt{c_u}}{\sqrt{c_u}}\frac{\lambda_M(\mathbf{L}) + \gamma_g}{\gamma_g^2 + 1}\right]$$

holds with probability $1-\delta$, where

$$R_P^W(\boldsymbol{\ell}^\star) \;=\; \frac{1}{n}\sum_i (\ell_i^\star - y_i)^2$$

$$\widehat{R}_P^W(\boldsymbol{\ell}^\star) \;=\; \frac{1}{n_l}\sum_{i\in l} (\ell_i^\star - y_i)^2.$$

SSL with Graphs: Generalization Bounds

Bounding transductive error

$$R_P^W(\boldsymbol{\ell}^\star) \;\leq\; \widehat{R}_P^W(\boldsymbol{\ell}^\star) + \underbrace{\beta + \sqrt{\frac{2\ln(2/\delta)}{n_l}}(n_l\beta + 4)}_{\text{transductive error } \Delta_T(\beta, n_l, \delta)}$$

$$\beta \;\leq\; 2\left[\frac{\sqrt{2}}{\gamma_g + 1} + \sqrt{2n_l}\frac{1-\sqrt{c_u}}{\sqrt{c_u}}\frac{\lambda_M(\mathbf{L}) + \gamma_g}{\gamma_g^2 + 1}\right]$$

Does the bound say anything useful?

1) The error is controlled.

2) Practical when error $\Delta_T(\beta, n_l, \delta)$ decreases at rate $O(n_l^{-\frac{1}{2}})$.
Achieved when $\beta = O(1/n_l)$. That is, $\gamma_g = \Omega(n_l^{\frac{3}{2}})$.

We have an idea how to set γ_g!

SSL with Graphs: Generalization Bounds

Combining inductive + transductive error

With probability $1-(\eta+\delta)$.

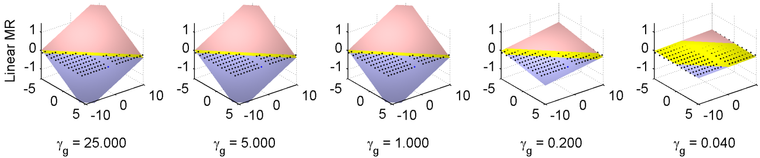
$$R_P(f) \leq \frac{1}{n}\sum_i \mathcal{L}(f(\mathbf{x}_i), \mathrm{sgn}(\ell_i^\star)) + \widehat{R}_P^W(\boldsymbol{\ell}^\star) + \Delta_T(\beta, n_l, \delta) + \Delta_I(h, n, \eta)$$

We need to account for ε. With probability $1-(\eta+\delta)$.

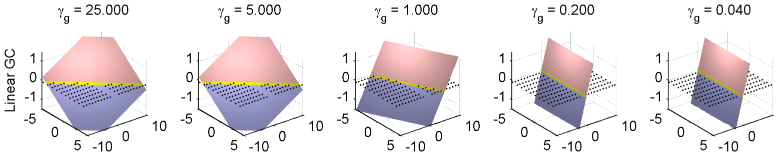
$$R_P(f) \leq \frac{1}{n}\sum_{i:|\ell_i^\star|\geq\varepsilon} \mathcal{L}(f(\mathbf{x}_i), \mathrm{sgn}(\ell_i^\star)) + \frac{2\varepsilon n_\varepsilon}{n} + \widehat{R}_P^W(\boldsymbol{\ell}^\star) + \Delta_T(\beta, n_l, \delta) + \Delta_I(h, n, \eta)$$

SSL with Graphs: LapSVMs and MM Graph Cuts

MR for 2D data and **linear** $\mathcal{K}$ only changes the slope

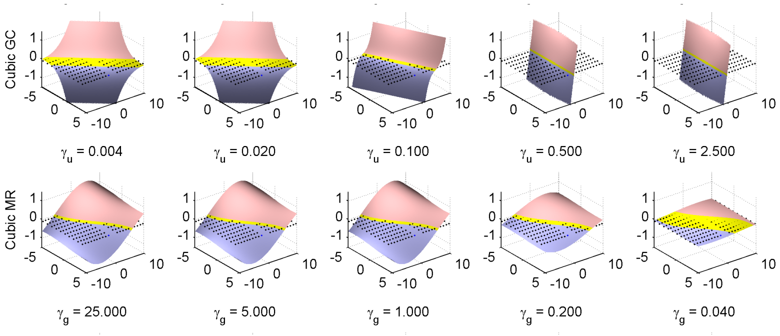


MMGC for 2D data and **linear** $\mathcal{K}$ works as we want



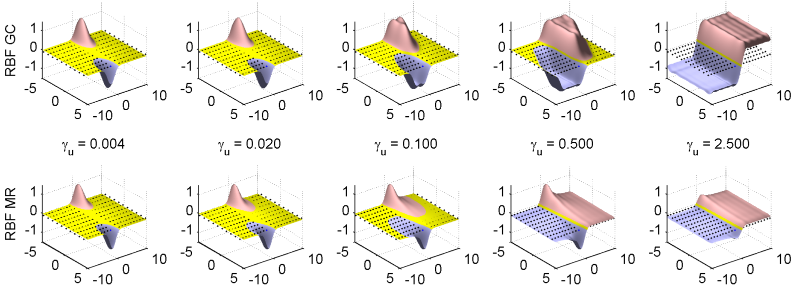
SSL with Graphs: LapSVMs and MM Graph Cuts

MR for 2D data and **cubic** $\mathcal{K}$ is also not so good

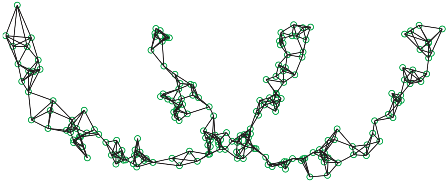


SSL with Graphs: LapSVMs and MM Graph Cuts

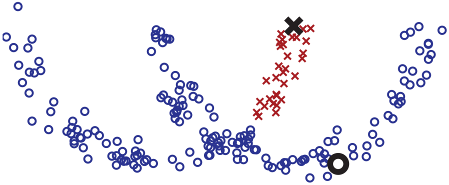
MMGC and MR for 2D data and RBF $\mathcal{K}$



SSL with Graphs



(a) 4-NN graph



(b) Harmonic function predictions

Graph-based SSL is obviously sensitive to graph construction!

Online SSL with Graphs

Offline learning setup

Given $\{\mathbf{x}_i\}_{i=1}^{n}$ from $\mathbb{R}^d$ and $\{y_i\}_{i=1}^{n_l}$, with $n_l \ll n$, find $\{y_i\}_{i=n_l+1}^{n}$ (transductive) or find f predicting y well beyond that (inductive).

Online learning setup

At the beginning: $\{\mathbf{x}_i, y_i\}_{i=1}^{n_l}$ from $\mathbb{R}^d$

At time t:

- receive $\mathbf{x}_t$
- predict y_t

Online SSL with Graphs

Online HFS: Straightforward solution

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

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

5: Predict $\widehat{y}_t = \text{sgn}(\mathbf{f}_u(t))$
6: **end while**

What is wrong with this solution?

The cost and memory of the operations.

What can we do?

Online SSL with Graphs

Let's keep only k vertices!

Limit memory to k **centroids** with $\tilde{\mathbf{W}}^{\mathrm{q}}$ weights.

Each centroids represents *several* others.

Diagonal $\mathbf{V} \equiv$ **multiplicity**. We have $\mathbf{V}_{ii}$ copies of centroid i.

Can we compute it compactly? Compact harmonic solution.

$$\boldsymbol{\ell}^{\mathrm{q}} = (\mathbf{L}^{\mathrm{q}}_{uu} + \gamma_g V)^{-1} \mathbf{W}^{\mathrm{q}}_{ul} \boldsymbol{\ell}_l \quad \text{where} \quad \mathbf{W}^{\mathrm{q}} = V \tilde{\mathbf{W}}^{\mathrm{q}} V$$

Proof? Using electric circuits.

Why do we keep the multiplicities?

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 x_t comes do
    Add x_t to graph G
    if # nodes > k then
        quantize G
    end if
    Update L_t of G(VWV)
    Infer labels
    Predict ŷ_t = sgn(f_u(t))
end while
```

Online SSL with Graphs: Graph Quantization

An idea: incremental k-centers

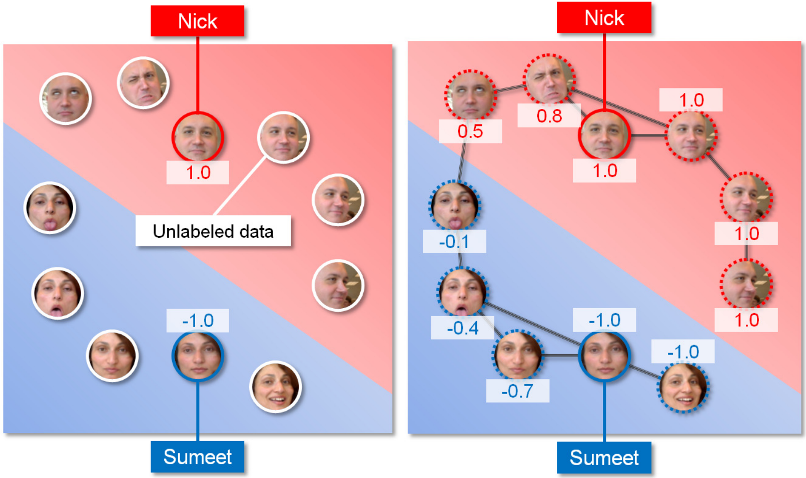
Doubling algorithm of Charikar et al. [Cha+97]

Keeps up to k centers $C_t = \{\mathbf{c}_1, \mathbf{c}_2, \dots\}$ with

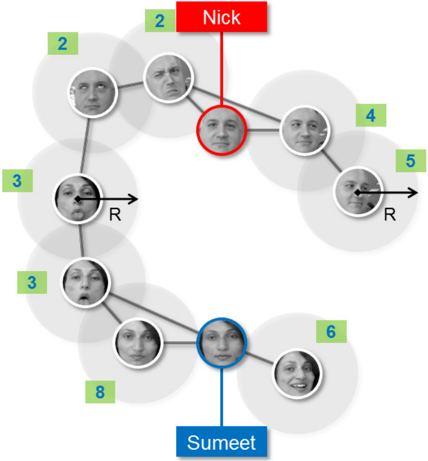
- Distance $\mathbf{c}_i, \mathbf{c}_j \in C_t$ is at least $\geq R$
- For each new $\mathbf{x}_t$, distance to some $\mathbf{c}_i \in C_t$ is less than R.
- $|C_t| \leq k$
- if not possible, R is doubled

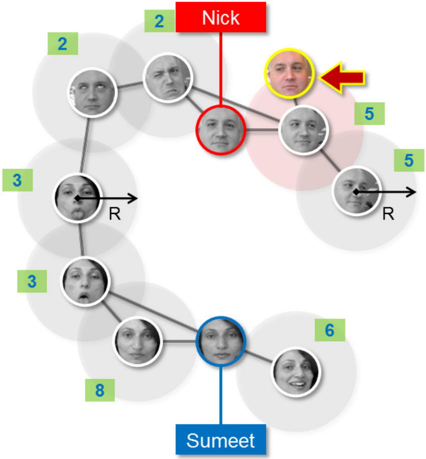
Online SSL with Graphs: Graph Quantization

Online SSL with Graphs: Graph Quantization

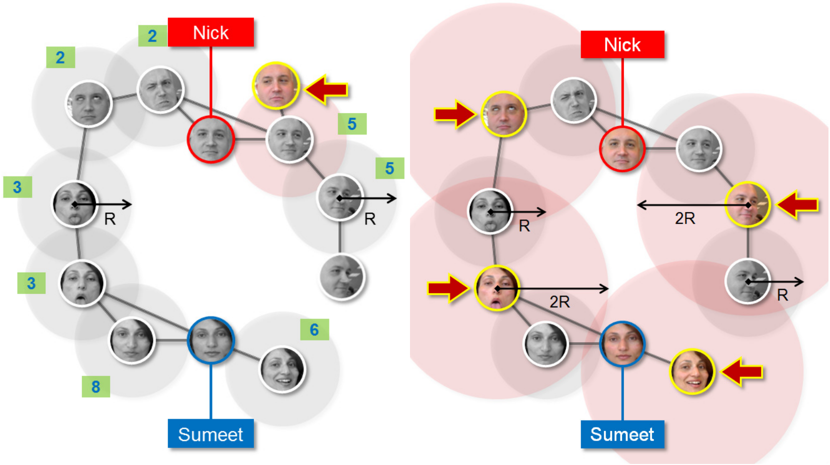


Online SSL with Graphs: Graph Quantization





Online SSL with Graphs: Graph Quantization



Online SSL with Graphs: Graph Quantization

Doubling algorithm [Cha+97]

To reduce growth of R, we use $R = m \times R$, with $m \geq 1$

C_t is changing. How far can $\mathbf{x}$ be from some $\mathbf{c}$?

$$R + \frac{R}{m} + \frac{R}{m^2} + \cdots = R\left(1 + \frac{1}{m} + \frac{1}{m^2} + \cdots\right) = \frac{Rm}{m-1}$$

Guarantees: $(1+\varepsilon)$-approximation algorithm.

Why not incremental k-means?

Online SSL with Graphs: Graph Quantization

Online k-centers

an unlabeled $\mathbf{x}_t$, a set of centroids C_{t-1}, multiplicities $\mathbf{v}_{t-1}$
if $(|C_{t-1}| = k + 1)$ **then**
    $R \leftarrow mR$
    greedily repartition C_{t-1} into C_t such that:
        no two vertices in C_t are closer than R
        for any $\mathbf{c}_i \in C_{t-1}$ exists $\mathbf{c}_j \in C_t$ such that $d(\mathbf{c}_i, \mathbf{c}_j) < R$
    update $\mathbf{v}_t$ to reflect the new partitioning
else
    $C_t \leftarrow C_{t-1}$
    $\mathbf{v}_t \leftarrow \mathbf{v}_{t-1}$
end if
if $\mathbf{x}_t$ is closer than R to any $\mathbf{c}_i \in C_t$ **then**
    $\mathbf{v}_t(i) \leftarrow \mathbf{v}_t(i) + 1$
else
    $\mathbf{v}_t(|C_t| + 1) \leftarrow 1$
end if

Michal Valko
michal.valko@inria.fr

sequel.lille.inria.fr