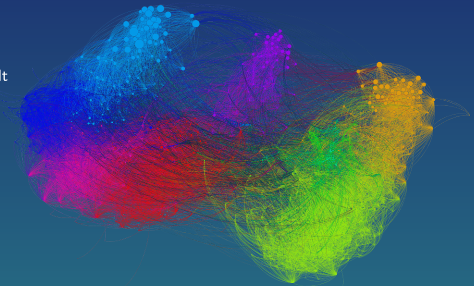
Graphs in Machine Learning

Michal Valko

DeepMind Paris and Inria Lille

TA: Omar Darwiche Domingues with the help of Pierre Perrault

Partially based on material by: Mikhail Belkin,
Jerry Zhu, Olivier Chapelle, Branislav Kveton



Previous lecture

- manifold learning with Laplacian eigenmaps
- resistive networks
  - recommendation score as a resistance?
  - Laplacian and resistive networks
  - resistance distance and random walks
- semi-supervised learning
- inductive and transductive semi-supervised learning
- SSL with self-training
- SVMs and semi-supervised SVMs = TSVMs
- Gaussian random fields and harmonic solution
- harmonic solution on graphs
- graph-based semi-supervised learning
- transductive learning

Previous lab session

- 15. 10. 2019 by Omar
- Content
  - graph construction
  - test sensitivity to parameters: σ, k, ε
  - spectral clustering
  - spectral clustering vs. k-means
  - image segmentation
- Short written report (graded, all reports around 40% of grade)
- Check the course website for the policies
- Questions to piazza
- *Deadline:* 29. 10. 2018, 23:59

This lecture

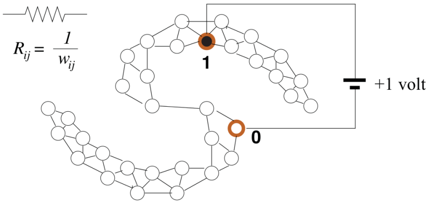
- graph-based semi-supervised learning and manifold regularization
- transductive learning
- inductive and transductive semi-supervised learning
- manifold regularization
- max-margin graph cuts
- theory of Laplacian-based manifold methods
- transductive learning stability based bounds
- online semi-supervised Learning
- online incremental k-centers

$\mathbf{SSL}(\mathcal{G})$

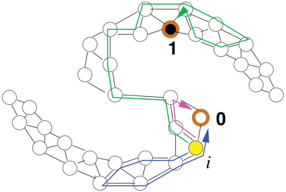
semi-supervised learning with graphs and harmonic functions

...our running example for learning with graphs

SSL with Graphs: Harmonic Functions



(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) $P(j|i) = \frac{w_{ij}}{\sum_k w_{ik}} \qquad \equiv \qquad \mathbf{P} = \mathbf{D}^{-1}\mathbf{W}$

3) finish when **a** labeled vertex is hit

absorbing random walk

f_i = probability of reaching a positive labeled vertex

SSL with Graphs: Harmonic Functions

How to compute HS? **Option A:** iteration/propagation

Step 1: Set $f(\mathbf{x}_i) = y_i$ for $i = 1, \ldots, n_l$
Step 2: Propagate iteratively (only for unlabeled)

$$f(\mathbf{x}_i) \leftarrow \frac{\sum_{i\sim j} f(\mathbf{x}_j) w_{ij}}{\sum_{i\sim j} w_{ij}} \qquad \forall i \in \{n_l + 1, \ldots, n_u + n_l\}$$

Properties:

- this will converge to the harmonic solution
- we can set the initial values for unlabeled nodes arbitrarily
- an interesting option for large-scale data

SSL with Graphs: Harmonic Functions

How to compute HS? **Option B:** Closed form solution

Define $\mathbf{f} = (f(\mathbf{x}_1), \ldots, f(\mathbf{x}_{n_l+n_u})) = (f_1, \ldots, f_{n_l+n_u})$

$$\Omega(\mathbf{f}) = \sum_{i,j=1}^{n_l+n_u} w_{ij} \left(f(\mathbf{x}_i) - f(\mathbf{x}_j)\right)^2 = \mathbf{f}^\mathsf{T}\mathbf{L}\mathbf{f}$$

$\mathbf{L}$ is a $(n_l + n_u) \times (n_l + n_u)$ matrix:

$$\mathbf{L} = \begin{bmatrix} \mathbf{L}_{ll} & \mathbf{L}_{lu} \\ \mathbf{L}_{u1} & \mathbf{L}_{uu} \end{bmatrix}$$

How to compute this **constrained** minimization problem?

SSL with Graphs: Harmonic Functions

Let us compute **harmonic** solution using **harmonic** property!

How did we formalize the harmonic property of a circuit?

$$(\mathbf{L}\mathbf{f})_u = \mathbf{0}_u$$

In matrix notation

$$\begin{bmatrix} \mathbf{L}_{ll} & \mathbf{L}_{lu} \\ \mathbf{L}_{ul} & \mathbf{L}_{uu} \end{bmatrix} \begin{bmatrix} \mathbf{f}_l \\ \mathbf{f}_u \end{bmatrix} = \begin{bmatrix} \dots \\ \mathbf{0}_u \end{bmatrix}$$

$\mathbf{f}_l$ is constrained to be $\mathbf{y}_l$ and for $\mathbf{f}_u$ ……

$$\mathbf{L}_{ul}\mathbf{f}_l + \mathbf{L}_{uu}\mathbf{f}_u = \mathbf{0}_u$$

…from which we get

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

Note that this does not depend on $\mathbf{L}_{ll}$.

SSL with Graphs: Harmonic Functions

Can we see that this calculates the probability of a random walk?

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

Note that $\mathbf{P} = \mathbf{D}^{-1}\mathbf{W}$. Then equivalently

$$\mathbf{f}_u = (\mathbf{I} - \mathbf{P}_{uu})^{-1}\mathbf{P}_{ul}\mathbf{f}_l.$$

Split the equation into +ve & -ve part:

$$\begin{aligned} f_i &= (\mathbf{I} - \mathbf{P}_{uu})_{iu}^{-1}\mathbf{P}_{ul}\mathbf{f}_l \\ &= \underbrace{\sum_{j:y_j=1} (\mathbf{I} - \mathbf{P}_{uu})_{iu}^{-1}\mathbf{P}_{uj}}_{p_i^{(+1)}} - \underbrace{\sum_{j:y_j=-1} (\mathbf{I} - \mathbf{P}_{uu})_{iu}^{-1}\mathbf{P}_{uj}}_{p_i^{(-1)}} \\ &= p_i^{(+1)} - p_i^{(-1)} \end{aligned}$$

SSL with Graphs: Regularized Harmonic Functions

$$f_i = p_i^{(+1)} - p_i^{(-1)} \qquad \Longrightarrow \quad f_i = \underbrace{|f_i|}_{\text{confidence}} \times \underbrace{\mathrm{sgn}(f_i)}_{\text{label}}$$

What if a nasty outlier sneaks in?

The prediction for the outlier can be hyperconfident :(

How to control the confidence of the inference?

Allow the random walk to **die**!

We add a **sink** to the graph.

sink = artificial label node with value 0

We connect it to every other vertex.

What will this do to our predictions?

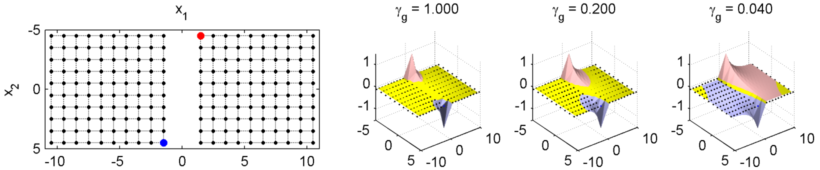
depends on the weigh on the edges

SSL with Graphs: Regularized Harmonic Functions

How do we compute this **regularized** random walk?

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

How does γ_g influence HS?



What happens to sneaky outliers?

SSL with Graphs: Harmonic Functions

Why don't we represent the sink in **L** explicitly?

Formally, to get the harmonic solution on the graph with sink ...

$$\begin{bmatrix} \mathbf{L}_{ll} + \gamma_G \mathbf{I}_{n_l} & \mathbf{L}_{lu} & -\gamma_G \\ \mathbf{L}_{ul} & \mathbf{L}_{uu} + \gamma_G \mathbf{I}_{n_u} & -\gamma_G \\ -\gamma_G \mathbf{1}_{n_l \times 1} & -\gamma_G \mathbf{1}_{n_u \times 1} & n\gamma_G \end{bmatrix} \begin{bmatrix} \mathbf{f}_l \\ \mathbf{f}_u \\ 0 \end{bmatrix} = \begin{bmatrix} \dots \\ \mathbf{0}_u \\ \dots \end{bmatrix}$$

$$\mathbf{L}_{ul}\mathbf{f}_l + \left(\mathbf{L}_{uu} + \gamma_G \mathbf{I}_{n_u}\right) \mathbf{f}_u = \mathbf{0}_u$$

...which is the same if we disregard the last column and row ...

$$\begin{bmatrix} \mathbf{L}_{ll} + \gamma_G \mathbf{I}_{n_l} & \mathbf{L}_{lu} \\ \mathbf{L}_{ul} & \mathbf{L}_{uu} + \gamma_G \mathbf{I}_{n_u} \end{bmatrix} \begin{bmatrix} \mathbf{f}_l \\ \mathbf{f}_u \end{bmatrix} = \begin{bmatrix} \dots \\ \mathbf{0}_u \end{bmatrix}$$

...and therefore we simply add γ_G to the diagonal of **L**!

SSL with Graphs: Soft Harmonic Functions

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

$$\min_{\mathbf{f} \in \mathbb{R}^{n_l + n_u}} \infty \sum_{i=1}^{n_l} (f(\mathbf{x}_i) - y_i)^2 + \lambda \mathbf{f}^\intercal \mathbf{Q} \mathbf{f}$$

What if we do not really believe that $f(\mathbf{x}_i) = y_i$, $\forall i$?

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

$\mathbf{C}$ is diagonal with $C_{ii} = \begin{cases} c_l & \text{for labeled examples} \\ c_u & \text{otherwise.} \end{cases}$

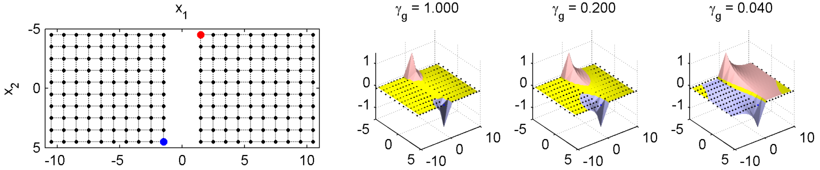
$\mathbf{y} \equiv$ pseudo-targets with $y_i = \begin{cases} \text{true label} & \text{for labeled examples} \\ 0 & \text{otherwise.} \end{cases}$

SSL with Graphs: Soft Harmonic Functions

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

Closed form **soft harmonic solution**:

$$\mathbf{f}^\star = (\mathbf{C}^{-1}\mathbf{Q}+\mathbf{I})^{-1}\mathbf{y}$$



What are the differences between hard and soft?

Not much different in practice.

Provable generalization guarantees for the soft one.

SSL with Graphs: Regularized Harmonic Functions

Larger implications of random walks

random walk relates to **commute distance** which should satisfy

($\star$) Vertices in the **same** cluster of the graph have a **small** commute distance, whereas two vertices in **different** clusters of the graph have a **large** commute distance.

Do we have this property for HS? What if $N \to \infty$?

Luxburg/Radl/Hein: *Getting lost in space: Large sample analysis of the commute distance* http://www.informatik.uni-hamburg.de/ML/contents/people/luxburg/publications/LuxburgRadlHein2010_PaperAndSupplement.pdf

Solutions? 1) γ_g 2) amplified commute distance 3) $\mathbf{L}^p$ 4) $\mathbf{L}^\star$ ...

The goal of these solutions: **make them remember!**

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**.

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

For general **kernels**:

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

SSL with Graphs: Manifold Regularization

$$f^\star = \arg\min_{f \in \mathcal{H}_\mathcal{K}} \sum_{i}^{n_l} V(\mathbf{x}_i, y_i, f) + \lambda_1 \|f\|_\mathcal{K}^2 + \lambda_2 \mathbf{f}^\mathsf{T} \mathbf{L} \mathbf{f}$$

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

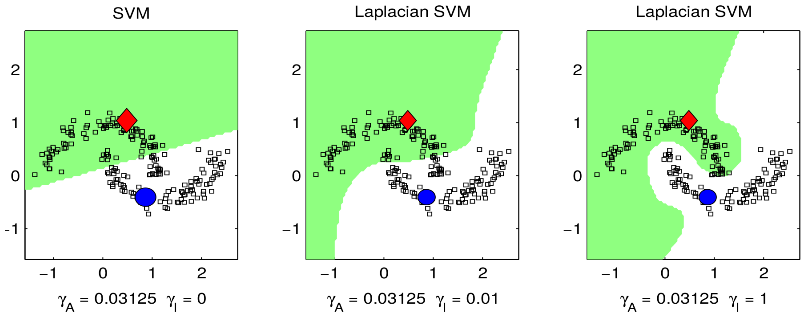
$$V(\mathbf{x}, y, f) = \max(0, 1 - yf(\mathbf{x}))$$

LapSVM Laplacian Support Vector Machines

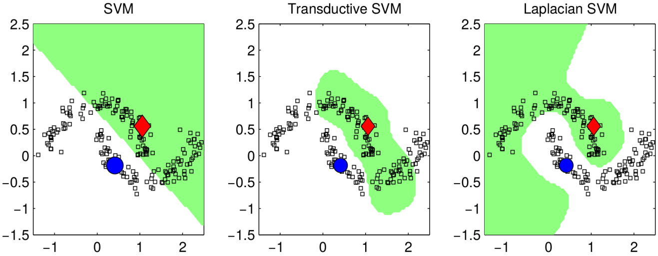
SSL with Graphs: Laplacian SVMs

$$f^\star = \underset{f \in \mathcal{H}_\mathcal{K}}{\arg\min} \sum_{i}^{n_l} \max\left(0, 1 - yf\left(\mathbf{x}\right)\right) + \gamma_A \|f\|_\mathcal{K}^2 + \gamma_I \mathbf{f}^\mathsf{T} \mathbf{L} \mathbf{f}$$

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



SSL with Graphs: Laplacian SVMs



Checkpoint 1

Semi-supervised learning with graphs:

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

Regularized harmonic Solution:

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

Checkpoint 2

Unconstrained regularization in general:

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

Out of sample extension: Laplacian SVMs

$$f^{\star} = \arg\min_{f \in \mathcal{H}_{\mathcal{K}}} \sum_{i}^{n_l} \max\left(0, 1 - y f\left(\mathbf{x}\right)\right) + \lambda_1 \|f\|_{\mathcal{K}}^{2} + \lambda_2 \mathbf{f}^{\mathsf{T}} \mathbf{L} \mathbf{f}$$

SSL with Graphs: Laplacian SVMs

$$f^\star = \underset{f \in \mathcal{H}_\mathcal{K}}{\arg\min} \sum_i^{n_l} \max\left(0, 1 - yf\left(\mathbf{x}\right)\right) + \lambda_1 \|f\|_\mathcal{K}^2 + \lambda_2 \mathbf{f}^\mathsf{T}\mathbf{L}\mathbf{f}$$

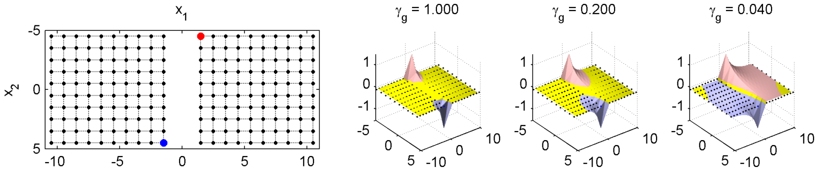
$\mathcal{H}_\mathcal{K}$ is nice and expressive.

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

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

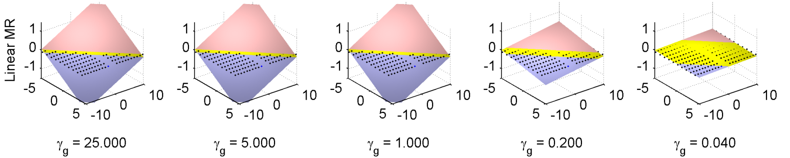
What does this objective function correspond to?

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

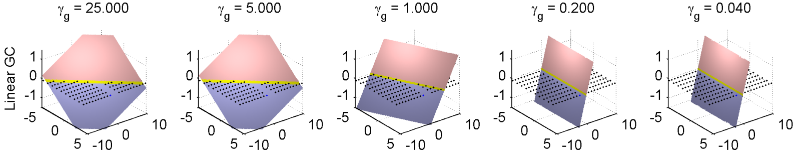
The same value of the objective as for supervised learning for some λ **without the unlabeled data**! This is not good.

SSL with Graphs: Laplacian SVMs

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



What would we like to see?



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

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 is 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

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

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

empirical risk + **error terms**

SSL with Graphs: Generalization Bounds

True risk vs. empirical risk

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

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

We look for the bound in the form

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

errors = **transductive** + **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

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

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

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

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 is 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?

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

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

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

SSL with Graphs: Generalization Bounds

Bounding transductive error

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

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

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

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

SSL with Graphs: Generalization Bounds

Bounding transductive error

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

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 $|\ell_i^\star| \leq 1$.

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

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

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

This algorithm is β-stable!

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]

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

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

holds with probability $1 - \delta$, where

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

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

SSL with Graphs: Generalization Bounds

Bounding transductive error

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

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

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

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

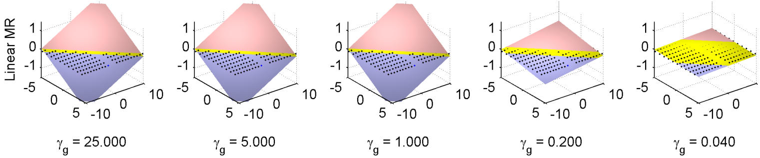
We need to account for ε. With probability $1-(\eta+\delta)$.

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

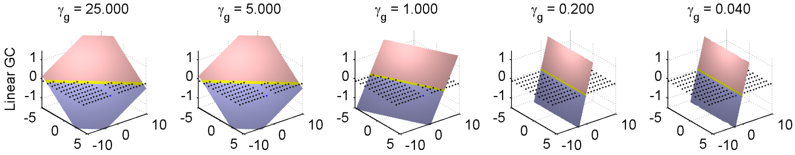
We should have $\varepsilon \leq n_l^{-1/2}$!

SSL with Graphs: LapSVMs and MM Graph Cuts

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

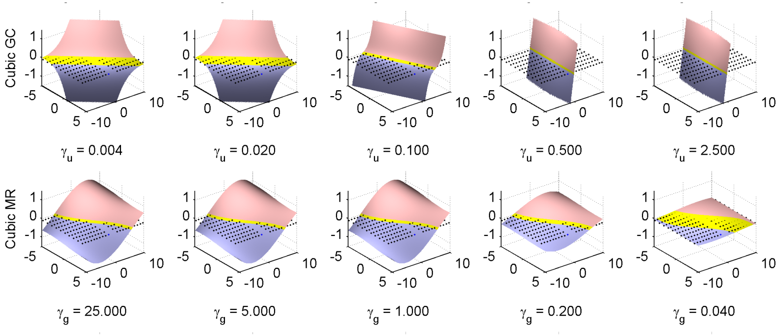


MMGC for 2D data and **linear** $\mathcal{K}$ works as we want



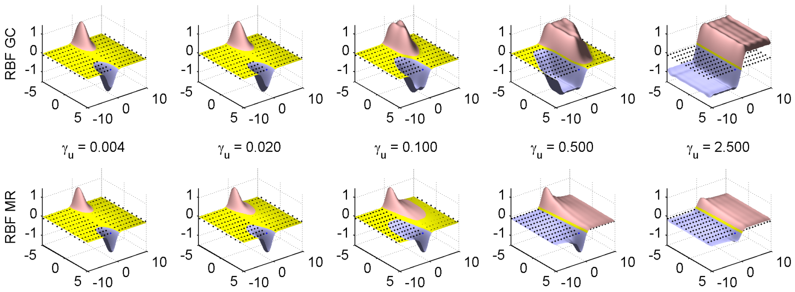
SSL with Graphs: LapSVMs and MM Graph Cuts

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

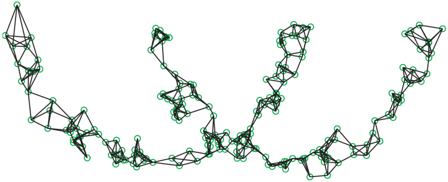


SSL with Graphs: LapSVMs and MM Graph Cuts

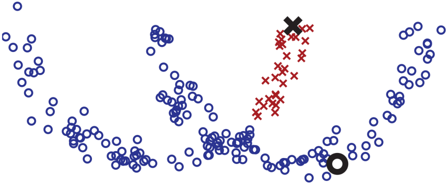
MMGC and MR for 2D data and RBF $\mathcal{K}$



SSL with Graphs



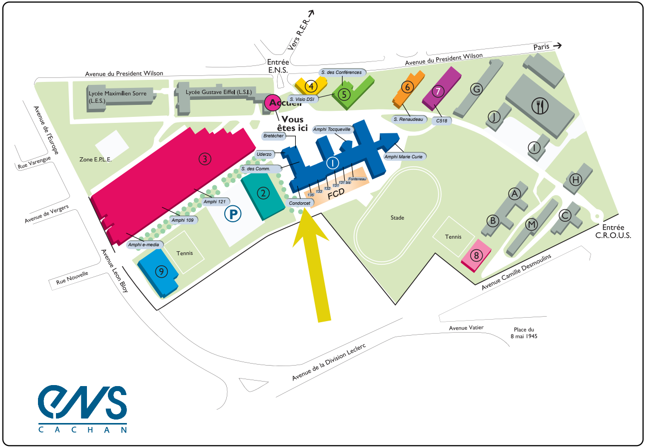
(a) 4-NN graph



(b) Harmonic function predictions

Graph-based SSL is obviously sensitive to graph construction!

Next lecture: Tuesday, November 6th at 13:30!



Michal Valko
contact via Piazza