Graphs in Machine Learning

Michal Valko

INRIA Lille - Nord Europe, France

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

- resistive networks
  - recommendation score as a resistance?
  - Laplacian and resistive networks
  - computation of effective resistance

- geometry of the data and the connectivity

- spectral clustering
  - connectivity vs. compactness
  - MinCut, RatioCut, NCut
  - spectral relaxations

- manifold learning
This Lecture

- manifold learning with Laplacian Eigenmaps
- Gaussian random fields and harmonic solution
- Graph-based semi-supervised learning and manifold regularization
- Theory of Laplacian-based manifold methods
- Transductive learning
- SSL Learnability
Previous Lab Session

- 3. 2. 2015 by Daniele.Calandriello@inria.fr
- Content
  - Graph Construction
  - Test sensitivity to parameters: $\sigma$, $k$, $\varepsilon$
  - Spectral Clustering
  - Spectral Clustering vs. $k$-means
  - Image Segmentation
- Short written report (graded, each lab around 5% of grade)
- Hint: Order 2.1, 2.6 (find the bend), 2.2, 2.3, 2.4, 2.5
- Questions to Daniele.Calandriello@inria.fr
- **Deadline:** 17. 2. 2015
- [http://researchers.lille.inria.fr/~calandri/ta/graphs/td1_handout.pdf](http://researchers.lille.inria.fr/~calandri/ta/graphs/td1_handout.pdf)
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

PhD proposal at CMU and JIE

- SYSU-CMU Joint Institute of Engineering (JIE) in Guangzhou, China:
  - International environment, English working language
- Fully-funded PhD positions available at SYSU-CMU JIE:
  - Single-degree program at SYSU in Guangzhou, China
  - Double-degree program (selective)
    - 2 years at CMU, Pittsburgh
    - rest of the time at JIE in Guangzhou, China
- Fundamental research with applications in:
  - Supercomputing & Big Data
  - Biomedical applications
  - Autonomous driving
  - Smart grids and power systems
- Contact: paweng@cmu.edu
Manifold Learning: Recap

**Problem:** Definition reduction/Manifold learning

Given \( \{x_i\}_{i=1}^n \) from \( \mathbb{R}^d \) find \( \{y_i\}_{i=1}^n \) in \( \mathbb{R}^m \), where \( m \ll d \).

- What do we know about the **dimensionality reduction**
  - representation/visualization (2D or 3D)
  - an old example: globe to a map
  - often assuming \( \mathcal{M} \subset \mathbb{R}^d \)
  - feature extraction
  - linear vs. nonlinear dimensionality reduction

- What do we know about linear vs. nonlinear methods?
  - linear: ICA, PCA, SVD, ...
  - nonlinear often preserve only **local** distances
Manifold Learning: Linear vs. Non-linear
Manifold Learning: Preserving (just) local distances

\[ d(y_i, y_j) = d(x_i, x_j) \quad \text{only if} \quad d(x_i, x_j) \quad \text{is small} \]

\[ \min \sum_{ij} w_{ij} \|y_i - y_j\|^2 \]

Looks familiar?
Manifold Learning: Laplacian Eigenmaps

**Step 1:** Solve generalized eigenproblem:

\[ Lf = \lambda Df \]

**Step 2:** Assign \( m \) new coordinates:

\[ x_i \mapsto (f_2(i), \ldots, f_m(i)) \]

**Note 1:** we need to get \( m \) smallest eigenvectors

**Note 2:** \( f_1 \) is useless

http://web.cse.ohio-state.edu/~mbelkin/papers/LEM_NC_03.pdf
Manifold Learning: Laplacian Eigenmaps to 1D

Laplacian Eigenmaps 1D objective

\[
\min_{f} f^T L f \quad \text{s.t.} \quad f_i \in \mathbb{R}, \quad f^T D 1 = 0, \quad f^T D f = 1
\]

The meaning for constraints is similar as for spectral clustering:

\( f^T D f = 1 \) is for scaling

\( f^T D 1 = 0 \) is to not get \( v_1 \)

What is the solution?
Manifold Learning: Example

Semi-supervised learning: How is it possible?

This is how children learn! hypothesis
Semi-supervised learning (SSL)

**SSL problem: definition**

Given \( \{x_i\}_{i=1}^n \) 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).

**Some facts about SSL**

- assumes that the unlabeled data is useful
- works with data geometry assumptions
  - cluster assumption - low-density separation
  - manifold assumption
  - smoothness assumptions, generative models, …
- now it helps now, now it does not (sic)
  - provable cases when it helps
- inductive or transductive/out-of-sample extension

SSL: Self-Training

(a) Iteration 1

(b) Iteration 25

(c) Iteration 74

(d) Final labeling of all instances
SSL: Overview: Self-Training

SSL: **Self-Training**

**Input:** $\mathcal{L} = \{x_i, y_i\}_{i=1}^{n_l}$ and $\mathcal{U} = \{x_i\}_{i=n_l+1}^{n}$

**Repeat:**
- train $f$ using $\mathcal{L}$
- apply $f$ to (some) $\mathcal{U}$ and add them to $\mathcal{L}$

What are the properties of self-training?

- it's a wrapper method
- heavily depends on the internal classifier
- some theory exist for specific classifiers
- nobody uses it anymore
- errors propagate (unless the cluster are well separated)
SSL: Self-Training: Bad Case
SSL: Transductive SVM: S3VM
SSL: Transductive SVM: Classical SVM

Linear case: \( f = \mathbf{w}^T \mathbf{x} + b \) \( \rightarrow \) we look for \((\mathbf{w}, b)\)

**max-margin classification**

\[
\begin{align*}
\max_{\mathbf{w}, b} & \quad \frac{1}{\|\mathbf{w}\|} \\
\text{s.t.} & \quad y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 \quad \forall i = 1, \ldots, n_l
\end{align*}
\]

**max-margin classification**

\[
\begin{align*}
\min_{\mathbf{w}, b} & \quad \|\mathbf{w}\|^2 \\
\text{s.t.} & \quad y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 \quad \forall i = 1, \ldots, n_l
\end{align*}
\]
SSL: Transductive SVM: Classical SVM

max-margin classification: **separable case**

\[
\min_{\mathbf{w}, b} \quad \|\mathbf{w}\|^2 \\
\text{s.t.} \quad y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1 \quad \forall i = 1, \ldots, n_l
\]

max-margin classification: **non-separable case**

\[
\min_{\mathbf{w}, b} \quad \lambda \|\mathbf{w}\|^2 + \sum_i \xi_i \\
\text{s.t.} \quad y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i \quad \forall i = 1, \ldots, n_l \\
\xi_i \geq 0 \quad \forall i = 1, \ldots, n_l
\]
SSL: Transductive SVM: Classical SVM

**max-margin classification: non-separable case**

\[
\begin{align*}
\min_{w,b} & \quad \lambda \|w\|^2 + \sum_{i} \xi_i \\
\text{s.t.} & \quad y_i (w^T x_i + b) \geq 1 - \xi_i \quad \forall i = 1, \ldots, n_l \\
& \quad \xi_i \geq 0 \quad \forall i = 1, \ldots, n_l
\end{align*}
\]

Unconstrained formulation:

\[
\begin{align*}
\min_{w,b} & \quad \sum_{i} \max (1 - y_i (w^T x_i + b), 0) + \lambda \|w\|^2
\end{align*}
\]

**In general?**

\[
\begin{align*}
\min_{w,b} & \quad \sum_{i} V(x_i, y_i, f(x_i)) + \lambda \Omega(f)
\end{align*}
\]
SSL: Transductive SVM: Unlabeled Examples

\[
\min_{w,b} \sum_{i}^{n_l} \max (1 - y_i (w^T x_i + b), 0) + \lambda \|w\|^2
\]

How to incorporate unlabeled examples?

No \( y \)'s for unlabeled \( x \).

Prediction of \( f \) for (any) \( x \)?

\[
\hat{y} = \text{sgn} (f (x)) = \text{sgn} (w^T x + b)
\]

Pretending that \( \text{sgn} (f (x)) \) is true . . .

\[
V(x, \hat{y}, f (x)) = \max (1 - \hat{y} (w^T x + b), 0)
\]

\[
= \max (1 - \text{sgn} (w^T x + b) (w^T x + b), 0)
\]

\[
= \max (1 - |w^T x + b|, 0)
\]
SSL: Transductive SVM: Hinge and Hat Loss

What is the difference in the objectives?

Hinge loss penalizes? the margin of being on the wrong side

Hat loss penalizes? predicting in the margin

(a) the hinge loss

(b) the hat loss
SSL: Transductive SVM: S3VM

This what we wanted!
SSL: Transductive SVM: Formulation

Main SVM idea stays: penalize the margin

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

What is the loss and what is the regularizer?

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

Think of unlabeled data as the regularizers for your classifiers!

Practical hint: Additionally enforce the class balance.

Another problem: Optimization is difficult.
SSL with Graphs: Prehistory

Blum/Chawla: Learning from Labeled and Unlabeled Data using Graph Mincuts


*following some insights from vision research in 1980s
SSL with Graphs: MinCut

MinCut SSL: an idea similar to MinCut clustering

Where is the link? connected classes, not necessarily compact

What is the formal statement? We look for $f(x) \in \{\pm 1\}$

\[
\text{cut} = \sum_{i,j=1}^{n_i+n_u} w_{ij} (f(x_i) - f(x_j))^2 = \Omega(f)
\]

Why $(f(x_i) - f(x_j))^2$ and not $|f(x_i) - f(x_j)|$? It does not matter.
SSL with Graphs: MinCut

We look for \( f(x) \in \{ \pm 1 \} \)

\[
\Omega(f) = \sum_{i,j=1}^{n_l+n_u} w_{ij} (f(x_i) - f(x_j))^2
\]

Clustering was unsupervised, here we have supervised data.

Recall the general objective framework:

\[
\min_{w,b} \sum_{i} V(x_i, y_i, f(x_i)) + \lambda \Omega(f)
\]

It would be nice if we match the prediction on labeled data:

\[
V(x, y, f(x)) = \infty \sum_{i=1}^{n_l} (f(x) - y)^2
\]
SSL with Graphs: MinCut

Final objective function:

\[
\min_{f \in \{\pm 1\}^{n_l+n_u}} \sum_{i=1}^{n_l} (f(x) - y)^2 + \lambda \sum_{i,j=1}^{n_l+n_u} w_{ij} (f(x_i) - f(x_j))^2
\]

This is an integer program :(

Can we solve it? It still just MinCut. Are we happy?

There are six solutions. All equivalent.

We need a better way to reflect the confidence.
SSL with Graphs: Harmonic Functions

Zhu/Ghahramani/Lafferty: Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions


*a seminal paper that convinced people to use graphs for SSL

Idea 1: Look for a unique solution.

Idea 2: Find a smooth one. (Harmonic solution)

Harmonic SSL

1): As before we constrain \( f \) to match the supervised data:

\[
f(x_i) = y_i \quad \forall i \in \{1, \ldots, n_l\}
\]

2): We enforce the solution \( f \) to be harmonic.

\[
f(x_i) = \frac{\sum_{i \sim j} f(x_j) w_{ij}}{\sum_{i \sim j} w_{ij}} \quad \forall i \in \{n_l + 1, \ldots, n_u + n_l\}
\]
SSL with Graphs: Harmonic Functions

The harmonic solution is obtained from the mincut one . . .

\[
\min_{f \in \{-1,1\}^{n_l+n_u}} \sum_{i=1}^{n_l} (f(x_i) - y_i)^2 + \lambda \sum_{i,j=1}^{n_l+n_u} w_{ij} (f(x_i) - f(x_j))^2
\]

. . . if we just relax the integer constraints to be real . . .

\[
\min_{f \in \mathbb{R}^{n_l+n_u}} \sum_{i=1}^{n_l} (f(x_i) - y_i)^2 + \lambda \sum_{i,j=1}^{n_l+n_u} w_{ij} (f(x_i) - f(x_j))^2
\]

. . . or equivalently (note that \(f(x_i) = f_i\)) . . .

\[
\min_{f \in \mathbb{R}^{n_l+n_u}} \sum_{i,j=1}^{n_l+n_u} w_{ij} (f(x_i) - f(x_j))^2
\]

s.t. \(y_i = f(x_i)\) \(\forall i = 1, \ldots, n_l\)
Properties of the relaxation from $\pm 1$ to $\mathbb{R}$

- there is a closed form solution for $f$
- this solution is unique
- globally optimal
- it is either constant or has a maximum /minimum on a boundary
- $f(x_i)$ may not be discrete
  - but we can threshold it
- random walk interpretation
- electric networks interpretation
SSL with Graphs: Harmonic Functions

Random walk interpretation:
1) start from the vertex to label and follow
2) $P(j|i) = \frac{w_{ij}}{\sum_k w_{ik}} \equiv P = D^{-1}W$
3) finish when the labeled vertex is hit
   absorbing random walk
$f_i = \text{probability of reaching a positive labeled vertex}$
SSL with Graphs: Harmonic Functions

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

**Step 1:** Set $f(x_i) = y_i$ for $i = 1, \ldots, n_l$

**Step 2:** Propagate iteratively (only for unlabeled)

$$f(x_i) \leftarrow \frac{\sum_{i \sim j} f(x_j)w_{ij}}{\sum_{i \sim j} w_{ij}} \quad \forall i \in \{n_l + 1, \ldots, n_u + n_l\}$$

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 \( \mathbf{f} = (f(x_1), \ldots, f(x_{n_l+n_u})) = (f_1, \ldots, f_{n_l+n_u}) \)

\[
\Omega(f) = \sum_{i,j=1}^{n_l+n_u} w_{ij} (f(x_i) - f(x_j))^2 = \mathbf{f}^T \mathbf{L} \mathbf{f}
\]

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

\[
\mathbf{L} = \begin{bmatrix}
\mathbf{L}_{ll} & \mathbf{L}_{lu} \\
\mathbf{L}_{ul} & \mathbf{L}_{uu}
\end{bmatrix}
\]

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

Yes, Lagrangian multipliers are an option, but . . .
Let us compute harmonic solution using harmonic property!

How did we formalize the harmonic property of a circuit?

\[(Lf)_u = 0\]

In matrix notation

\[
\begin{bmatrix}
L_{ll} & L_{lu} \\
L_{ul} & L_{uu}
\end{bmatrix}
\begin{bmatrix}
f_l \\
f_u
\end{bmatrix}
=
\begin{bmatrix}
0_l \\
0_u
\end{bmatrix}
\]

\(f_l\) is constrained to be \(y_l\) and for \(f_u\) ........

\[L_{ul}f_l + L_{uu}f_u = 0_u\]

...from which we get

\[f_u = L_{uu}^{-1}(-L_{ul}f_l) = L_{uu}^{-1}(W_{ul}f_l).\]
SSL with Graphs: Harmonic Functions

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

\[ f_u = L_{uu}^{-1} (-L_{ul} f_l) = L_{uu}^{-1} (W_{ul} f_l) \]

Note that \( P = D^{-1} W \). Then equivalently

\[ f_u = (I - P_{uu})^{-1} P_{ul} f_l. \]

Split the equation into +ve & -ve part:

\[ f_i = (I - P_{uu})_{iu}^{-1} P_{ul} f_l \\
= \sum_{j : y_j = 1} (I - P_{uu})_{iu}^{-1} P_{uj} - \sum_{j : y_j = -1} (I - P_{uu})_{iu}^{-1} P_{uj} \]

\[ = p_i^{(+1)} - p_i^{(-1)} \]
SSL with Graphs: Regularized Harmonic Functions

\[ f_i = |f_i| \times \text{sgn}(f_i) \]

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?

$$f_u = (L_{uu} + \gamma_g I)^{-1} (W_{ul} f_l)$$

How does $\gamma_g$ influence HS?

What happens to sneaky outliers?
SSL with Graphs: Soft Harmonic Functions

Regularized HS objective with $Q = L + \gamma g I$:

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

What if we do not really believe that $f(x_i) = y_i, \forall i$?

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

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

$y \equiv$ 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

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

Closed form soft harmonic solution:

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

What are the differences between hard and soft?

Not much different in practice.

Provable generalization guarantees for soft.
SSL with Graphs: Regularized Harmonic Functions

Larger implications of random walks

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

(*) 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? \[ What \text{ if } n \to \infty? \]

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

**Solutions?** 1) \( \gamma_g \) 2) amplified commute distance 3) \( L^p \) 4) \( L^* \) …

The goal of these solutions: **make them remember!**
Michal Valko
michal.valko@inria.fr
sequel.lille.inria.fr