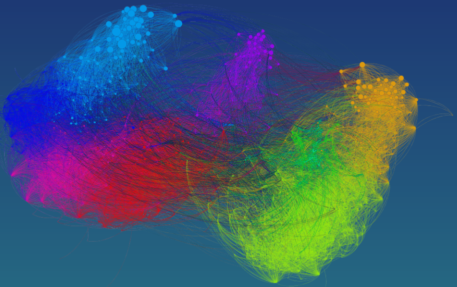
Graphs in Machine Learning

Michal Valko

DeepMind Paris and Inria Lille

TA: Omar Darwiche Domingues with the help of Pierre Perrault

Partially based on material by: Mikhail Belkin, Branislav Kveton
Rob Fergus, Nikhil Srivastava, Yiannis Koutis,
Joshua Batson, Daniel Spielman



November 5, 2019 **MVA 2019/2020**

Previous Lecture

- Regularization of harmonic solution
- Soft-harmonic solution
- Inductive and transductive semi-supervised learning
- Manifold regularization
- Max-Margin Graph Cuts
- Theory of Laplacian-based manifold methods
- Transductive learning stability based bounds
- Theory of Laplacian-based manifold methods
- Transductive learning stability based bounds

This Lecture

- Online Semi-Supervised Learning
- Online incremental *k*-centers
- Examples of applications of online SSL
- Analysis of online SSL
- SSL learnability
- When does graph-based SSL provably help?
- Scaling harmonic functions to millions of samples

Next Lab Session

- 12. 11. 2019 by Omar (and Pierre)
- Content
  - Semi-supervised learning
  - Graph quantization
  - Offline face recognizer
- Short written report
- Questions to piazza
- *Deadline:* 26. 11. 2019

Final class projects

- detailed description on the class website
- preferred option: you come up with the topic
- theory/implementation/review or a combination
- one or two people per project (exceptionally three)
- grade 60%: report + short presentation of the **team**
- deadlines
  - 19. 11. 2019 - strongly recommended DL for taking projects
  - 26. 11. 2019 - hard DL for taking projects
  - 07. 01. 2020 - submission of the project report
  - 13. 01. 2020 or later - project presentation
- list of suggested topics on piazza

OnlineSSL$(\mathcal{G})$

when we can't access future **x**

...and we want the results in real time

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

```
while new unlabeled example $\mathbf{x}_t$ comes do
    Add $\mathbf{x}_t$ to graph $G(\mathbf{W})$
    Update $\mathbf{L}_t$
    Infer labels
        $\mathbf{f}_u = (\mathbf{L}_{uu} + \gamma_g \mathbf{I})^{-1} (\mathbf{W}_{ul}\mathbf{f}_l)$
    Predict $\widehat{y}_t = \operatorname{sgn}(\mathbf{f}_u(t))$
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 $\widetilde{\mathbf{W}}^{\mathrm{q}}$ weights.

Each centroid 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.

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

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
    $\mathbf{V}$ matrix of multiplicities with 1 on diagonal
while new unlabeled example $\mathbf{x}_t$ comes do
    Add $\mathbf{x}_t$ to graph $G$
    if $\#$ nodes $> k$ then
        quantize $G$
    end if
    Update $\mathbf{L}_t$ of $G(\mathbf{VWV})$
    Infer labels
    Predict $\widehat{y}_t = \mathrm{sgn}\left(\mathbf{f}_u\left(t\right)\right)$
end while
```

Online SSL with Graphs: Graph Quantization

An idea: incremental k-centers

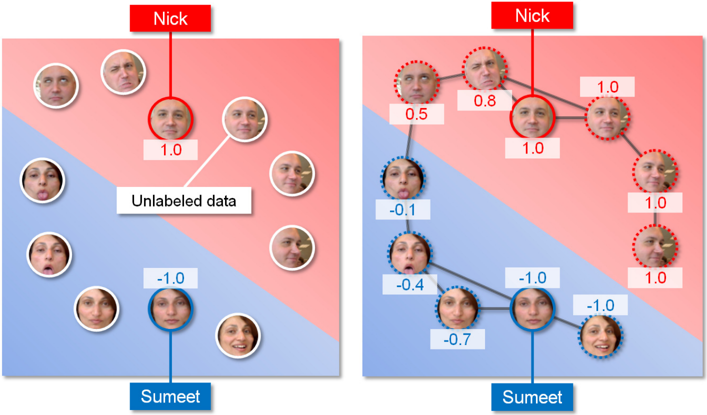
Doubling algorithm of Charikar et al. [Cha+97]

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

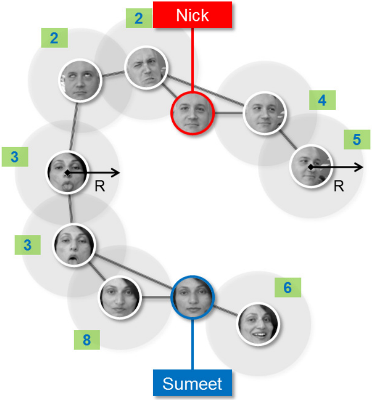
- Distance $\mathbf{c}_i, \mathbf{c}_j \in C_t$ is at least $\geq R$
- For each new $\mathbf{x}_t$, distance to some $\mathbf{c}_i \in C_t$ is less than R.
- $|C_t| \leq k$
- if not possible, R is doubled

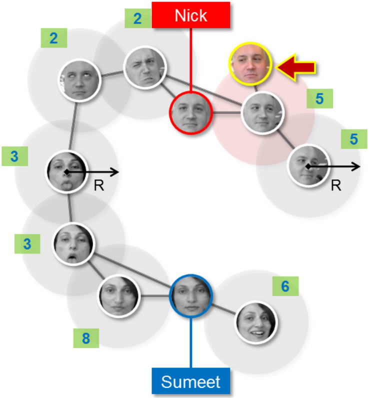
Online SSL with Graphs: Graph Quantization

Online SSL with Graphs: Graph Quantization

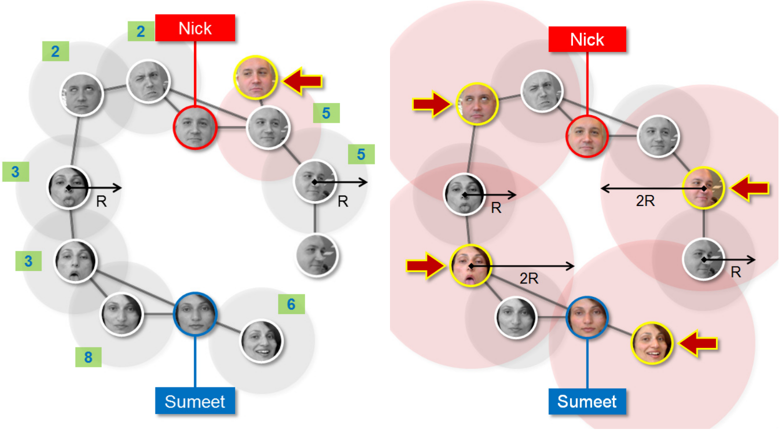


Online SSL with Graphs: Graph Quantization





Online SSL with Graphs: Graph Quantization



Online SSL with Graphs: Graph Quantization

Doubling algorithm [Cha+97]

To reduce growth of R, we use $R \leftarrow m \times R$, 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**
```

Online SSL with Graphs

Video examples

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

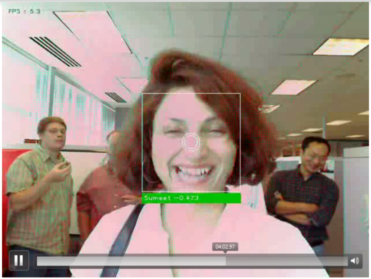
http://www.bkveton.com/videos/Ad.mp4

http://researchers.lille.inria.fr/~valko/hp/serve.php?what=publications/kveton2009nipsdemo.adaptation.mov

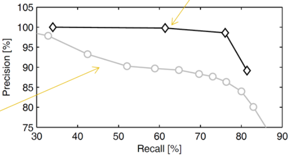
http://researchers.lille.inria.fr/~valko/hp/serve.php?what=publications/kveton2009nipsdemo.officespace.mov

http://researchers.lille.inria.fr/~valko/hp/publications/press-intel-2015-mac.mp4

SSL with Graphs: Some experimental results



- 8 people classification
- Making funny faces
- 4 faces/person are labeled

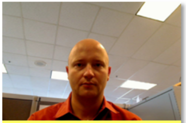


Our method

Nearest Neighbor

SSL with Graphs: Some experimental results

- One person moves among various indoor locations
- 4 labeled examples of a person in the cubicle

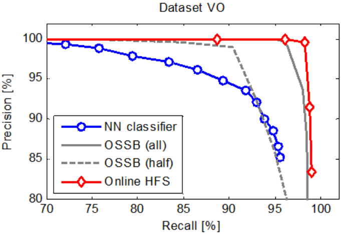

Labeled

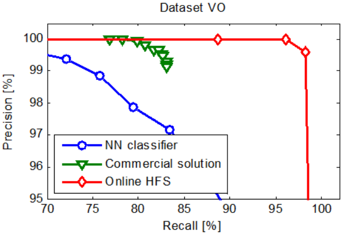
Unlabeled

Unlabeled

Unlabeled

Unlabeled





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

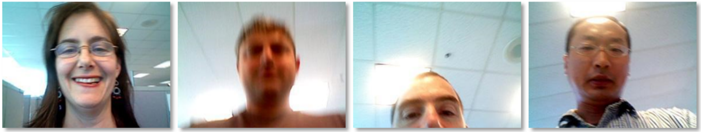
Online HFS yields better results than a commercial solution at 20% of the computational cost

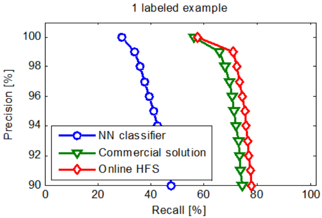
SSL with Graphs: Some experimental results

- **Logging in** with faces instead of password
- Able to **learn** and improve

SSL with Graphs: Some experimental results

- 16 people log twice into a tablet PC at 10 locations




1 labeled example

4 labeled examples

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

Online SSL with Graphs: Analysis

What can we guarantee?

Three sources of error

- generalization error — if all data: $(\ell_t^\star - y_t)^2$
- online error — data only incrementally: $(\ell_t^{\mathrm{o}}[t] - \ell_t^\star)^2$
- quantization error — memory limitation: $(\ell_t^{\mathrm{q}}[t] - \ell_t^{\mathrm{o}}[t])^2$

All together:

$$\frac{1}{N}\sum_{t=1}^{N}(\ell_t^{\mathrm{q}}[t]-y_t)^2 \leq \frac{9}{2N}\sum_{t=1}^{N}(\ell_t^\star-y_t)^2+\frac{9}{2N}\sum_{t=1}^{N}(\ell_t^{\mathrm{o}}[t]-\ell_t^\star)^2+\frac{9}{2N}\sum_{t=1}^{N}(\ell_t^{\mathrm{q}}[t]-\ell_t^{\mathrm{o}}[t])$$

Since for any a, b, c, $d \in [-1, 1]$:

$$(a-b)^2 \leq \frac{9}{2}\left[(a-c)^2+(c-d)^2+(d-b)^2\right]$$

Online SSL with Graphs: Analysis

Bounding transduction error $(\ell_t^\star - y_t)^2$

If all labeled examples l are i.i.d., $c_l = 1$ and $c_l \gg c_u$, then

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

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

holds with the probability of $1 - \delta$, where

$$R(\boldsymbol{\ell}^\star) = \frac{1}{N}\sum_t(\ell_t^\star - y_t)^2 \quad \text{and} \quad \widehat{R}(\boldsymbol{\ell}^\star) = \frac{1}{n_l}\sum_{t \in l}(\ell_t^\star - y_t)^2$$

How should we set γ_g?

Online SSL with Graphs: Analysis

Bounding online error $(\ell^{\mathrm{o}}_t[t] - \ell^\star_t)^2$

Idea: If $\mathbf{L}$ and $\mathbf{L}^{\mathrm{o}}$ are regularized, then HFSs get closer together.

since they get closer to zero

Recall $\boldsymbol{\ell} = (\mathbf{C}^{-1}\mathbf{Q} + \mathbf{I})^{-1}\mathbf{y}$, where $\mathbf{Q} = \mathbf{L} + \gamma_g \mathbf{I}$

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

$$\|\boldsymbol{\ell}\|_2 \le \frac{\|\mathbf{y}\|_2}{\lambda_m(\mathbf{C}^{-1}\mathbf{Q}+\mathbf{I})} = \frac{\|\mathbf{y}\|_2}{\frac{\lambda_m(\mathbf{Q})}{\lambda_M(\mathbf{C})}+1} \le \frac{\sqrt{n_l}}{\gamma_g + 1}$$

Difference between offline and online solutions:

$$(\ell^{\mathrm{o}}_t[t] - \ell^\star_t)^2 \le \|\boldsymbol{\ell}^{\mathrm{o}}[t] - \boldsymbol{\ell}^\star\|_\infty^2 \le \|\boldsymbol{\ell}^{\mathrm{o}}[t] - \boldsymbol{\ell}^\star\|_2^2 \le \left(\frac{2\sqrt{n_l}}{\gamma_g+1}\right)^2$$

Again, how should we set γ_g?

Online SSL with Graphs: Analysis

Bounding quantization error $\left(\ell_t^{\mathrm{q}}[t] - \ell_t^{\mathrm{o}}[t]\right)^2$

How are the quantized and full solution different?

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

In **Q**! $\mathbf{Q}^{\mathrm{o}}$ (online) vs. $\mathbf{Q}^{\mathrm{q}}$ (quantized)

We have: $\boldsymbol{\ell}^{\mathrm{o}} = (\mathbf{C}^{-1}\mathbf{Q}^{\mathrm{o}} + \mathbf{I})^{-1}\mathbf{y}$ vs. $\boldsymbol{\ell}^{\mathrm{q}} = (\mathbf{C}^{-1}\mathbf{Q}^{\mathrm{q}} + \mathbf{I})^{-1}\mathbf{y}$

Let $\mathbf{Z}^{\mathrm{q}} = \mathbf{C}^{-1}\mathbf{Q}^{\mathrm{q}} + \mathbf{I}$ and $\mathbf{Z}^{\mathrm{o}} = \mathbf{C}^{-1}\mathbf{Q}^{\mathrm{o}} + \mathbf{I}$.

$$\begin{aligned}\boldsymbol{\ell}^{\mathrm{q}} - \boldsymbol{\ell}^{\mathrm{o}} &= (\mathbf{Z}^{\mathrm{q}})^{-1}\mathbf{y} - (\mathbf{Z}^{\mathrm{o}})^{-1}\mathbf{y} = (\mathbf{Z}^{\mathrm{q}}\mathbf{Z}^{\mathrm{o}})^{-1}(\mathbf{Z}^{\mathrm{o}} - \mathbf{Z}^{\mathrm{q}})\mathbf{y}\\ &= (\mathbf{Z}^{\mathrm{q}}\mathbf{Z}^{\mathrm{o}})^{-1}\mathbf{C}^{-1}(\mathbf{Q}^{\mathrm{o}} - \mathbf{Q}^{\mathrm{q}})\mathbf{y}\end{aligned}$$

Online SSL with Graphs: Analysis

Bounding quantization error $\left(\ell_t^{\mathrm{q}}[t] - \ell_t^{\mathrm{o}}[t]\right)^2$

$$\ell^{\mathrm{q}} - \ell^{\mathrm{o}} = (\mathbf{Z}^{\mathrm{q}})^{-1}\mathbf{y} - (\mathbf{Z}^{\mathrm{o}})^{-1}\mathbf{y} = (\mathbf{Z}^{\mathrm{q}}\mathbf{Z}^{\mathrm{o}})^{-1}(\mathbf{Z}^{\mathrm{o}} - \mathbf{Z}^{\mathrm{q}})\mathbf{y}$$
$$= (\mathbf{Z}^{\mathrm{q}}\mathbf{Z}^{\mathrm{o}})^{-1}\mathbf{C}^{-1}(\mathbf{Q}^{\mathrm{o}} - \mathbf{Q}^{\mathrm{q}})\mathbf{y}$$

$$\|\ell^{\mathrm{q}} - \ell^{\mathrm{o}}\|_2 \le \frac{\lambda_M(\mathbf{C}^{-1})\|(\mathbf{Q}^{\mathrm{q}} - \mathbf{Q}^{\mathrm{o}})\mathbf{y}\|_2}{\lambda_m(\mathbf{Z}^{\mathrm{q}})\lambda_m(\mathbf{Z}^{\mathrm{o}})}$$

$|| \cdot ||_F$ and $|| \cdot ||_2$ are compatible and y_i is zero when unlabeled:

$$\|(\mathbf{Q}^{\mathrm{q}} - \mathbf{Q}^{\mathrm{o}})\mathbf{y}\|_2 \le \|\mathbf{Q}^{\mathrm{q}} - \mathbf{Q}^{\mathrm{o}}\|_F \cdot \|\mathbf{y}\|_2 \le \sqrt{n_l}\|\mathbf{Q}^{\mathrm{q}} - \mathbf{Q}^{\mathrm{o}}\|_F$$

Furthermore, $\lambda_m(\mathbf{Z}^{\mathrm{o}}) \ge \frac{\lambda_m(\mathbf{Q}^{\mathrm{o}})}{\lambda_M(\mathbf{C})} + 1 \ge \gamma_g$ and $\lambda_M\left(\mathbf{C}^{-1}\right) \le c_u^{-1}$

We get $\|\ell^{\mathrm{q}} - \ell^{\mathrm{o}}\|_2 \le \frac{\sqrt{n_l}}{c_u\gamma_g^2}\|\mathbf{Q}^{\mathrm{q}} - \mathbf{Q}^{\mathrm{o}}\|_F$

Online SSL with Graphs: Analysis

Bounding **quantization error** $\left(\ell_t^{\mathrm{q}}[t] - \ell_t^{\mathrm{o}}[t]\right)^2$

The quantization error depends on $\|\mathbf{Q}^{\mathrm{q}} - \mathbf{Q}^{\mathrm{o}}\|_F = \|\mathbf{L}^{\mathrm{q}} - \mathbf{L}^{\mathrm{o}}\|_F$.

When can we keep $\|\mathbf{L}^{\mathrm{q}} - \mathbf{L}^{\mathrm{o}}\|_F$ under control?

Charikar guarantees **distortion** error of at most $Rm/(m-1)$

For what kind of data $\{\mathbf{x}_i\}_{i=1,\ldots,n}$ is the distortion small?

Assume manifold $\mathcal{M}$

- all $\{\mathbf{x}_i\}_{i\geq 1}$ lie on a smooth s-dimensional compact $\mathcal{M}$
- with boundary of bounded geometry Def. 11 of Hein [HAL07]
  - should not intersect itself
  - should not fold back onto itself
  - has finite volume V
  - has finite surface area A

Online SSL with Graphs: Analysis

Bounding quantization error $\left(\ell_t^{\mathrm{q}}[t]-\ell_t^{\mathrm{o}}[t]\right)^2$

Bounding $\|\mathbf{L}^{\mathrm{q}}-\mathbf{L}^{\mathrm{o}}\|_F$ when $\mathbf{x}_i \in \mathcal{M}$

Consider k-sphere packing* of radius r with centers contained in $\mathcal{M}$. *only the centers are packed, not necessarily the entire ball

What is the maximum volume of this packing*?

$kc_s r^s \leq V + Ac_{\mathcal{M}} r$ with $c_s, c_{\mathcal{M}}$ depending on dimension and $\mathcal{M}$.

If k is large → $r <$ **injectivity radius** of $\mathcal{M}$ [HAL07] and $r < 1$:

$$r < \left(\left(V + Ac_{\mathcal{M}}\right) / \left(kc_s\right)\right)^{1/s} = \mathcal{O}\left(k^{-1/s}\right)$$

r-packing is a $2r$-covering:

$$\max_{i=1,\dots,N} \|\mathbf{x}_i - \mathbf{c}\|_2 \leq Rm/(m-1) \leq 2(1+\varepsilon)\mathcal{O}\left(k^{-1/s}\right) = \mathcal{O}\left(k^{-1/s}\right)$$

But what about $\|\mathbf{L}^{\mathrm{q}}-\mathbf{L}^{\mathrm{o}}\|_F$?

Online SSL with Graphs: Analysis

Bounding **quantization error** $\left(\ell_t^{\mathrm{q}}[t] - \ell_t^{\mathrm{o}}[t]\right)^2$

If similarity is M-Lipschitz, $\mathbf{L}$ is normalized, $c_{ij}^{\mathrm{o}} = \sqrt{\mathbf{D}_{ii}^{\mathrm{o}}\mathbf{D}_{jj}^{\mathrm{o}}} > c_{min}N$

$|\mathbf{W}_{ij}^{\mathrm{q}} - \mathbf{W}_{ij}^{\mathrm{o}}| < 2MRm/(m-1)$ and $|c_{ij}^{\mathrm{q}} - c_{ij}^{\mathrm{o}}| < 2nMRm/(m-1)$:

$$\begin{aligned}
\mathbf{L}_{ij}^{\mathrm{q}} - \mathbf{L}_{ij}^{\mathrm{o}} &= \frac{\mathbf{W}_{ij}^{\mathrm{q}}}{c_{ij}^{\mathrm{q}}} - \frac{\mathbf{W}_{ij}^{\mathrm{o}}}{c_{ij}^{\mathrm{o}}} \\
&\leq \frac{\mathbf{W}_{ij}^{\mathrm{q}} - \mathbf{W}_{ij}^{\mathrm{o}}}{c_{ij}^{\mathrm{q}}} + \frac{\mathbf{W}_{ij}^{\mathrm{o}}(c_{ij}^{\mathrm{o}} - c_{ij}^{\mathrm{q}})}{c_{ij}^{\mathrm{o}} c_{ij}^{\mathrm{q}}} \\
&\leq \frac{4MRm}{(m-1)c_{min}N} + \frac{4M(NMRm)}{((m-1)c_{min}N)^2} \\
&= O\left(\frac{R}{N}\right)
\end{aligned}$$

Finally, $\|\mathbf{L}^{\mathrm{q}} - \mathbf{L}^{\mathrm{o}}\|_F^2 \leq N^2\mathcal{O}(R^2/N^2) = \mathcal{O}(k^{-2/s})$.

Are the assumptions reasonable?

Online SSL with Graphs: Analysis

Bounding **quantization error** $\left(\ell_t^{\mathrm{q}}[t] - \ell_t^{\mathrm{o}}[t]\right)^2$

We showed $\|\mathbf{L}^{\mathrm{q}} - \mathbf{L}^{\mathrm{o}}\|_F^2 \leq N^2 \mathcal{O}(R^2/N^2) = \mathcal{O}(k^{-2/s}) = \mathcal{O}(1)$.

$$\frac{1}{N}\sum_{t=1}^{N}(\ell_t^{\mathrm{q}}[t] - \ell_t^{\mathrm{o}}[t])^2 \leq \frac{n_l}{c_u^2\gamma_g^4}\|\mathbf{L}^{\mathrm{q}} - \mathbf{L}^{\mathrm{o}}\|_F^2 \leq \frac{n_l}{c_u^2\gamma_g^4}$$

This converges to zero at the rate $\mathcal{O}(N^{-1/2})$ with $\gamma_g = \Omega(N^{1/8})$.

With properly setting γ_g, e.g., $\gamma_g = \Omega(N^{1/8})$, we can have

$$\frac{1}{N}\sum_{t=1}^{N}\left(\ell_t^{\mathrm{q}}[t] - y_t\right)^2 = \mathcal{O}\left(N^{-1/2}\right).$$

What does that mean?

SSL with Graphs: What is behind it?

Why and when it helps?

Can we guarantee benefit of SSL over SL?

Are there cases when **manifold** SSL is provably helpful?

Say $\mathcal{X}$ is supported on manifold $\mathcal{M}$. Compare two cases:

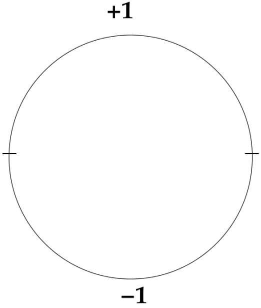
- SL: does not know about $\mathcal{M}$ and only knows $(\mathbf{x}_i, y_i)$
- SSL: perfect knowledge of $\mathcal{M} \equiv$ humongous amounts of $\mathbf{x}_i$

http://people.cs.uchicago.edu/~niyogi/papersps/ssminimax2.pdf

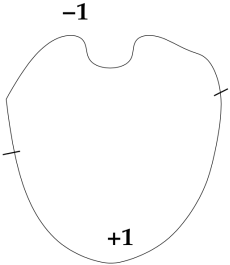
SSL with Graphs: What is behind it?

Set of learning problems - collections $\mathcal{P}$ of probability distributions:

$$\mathcal{P} = \cup_{\mathcal{M}} \mathcal{P}_{\mathcal{M}} = \cup_{\mathcal{M}} \{ p \in \mathcal{P} | p_{\mathcal{X}} \text{ is uniform on } \mathcal{M} \}$$



$\mathbf{M}_1$



$\mathbf{M}_2$

SSL with Graphs: What is behind it?

Set of problems $\mathcal{P} = \cup_{\mathcal{M}} \mathcal{P}_{\mathcal{M}} = \{p \in \mathcal{P} | p_{\mathcal{X}} \text{ is uniform on } \mathcal{M}\}$
Regression function $m_p = \mathbb{E}[y\,|x]$ when $x \in \mathcal{M}$
Algorithm A and **labeled examples** $\overline{z} = \{z_i\}_{i=1}^{n_l} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n_l}$
Minimax rate

$$R(n_l, \mathcal{P}) = \inf_{A} \sup_{p \in \mathcal{P}} \mathbb{E}_{\overline{z}}\left[\|A(\overline{z}) - m_p\|_{L^2(p_{\mathbf{X}})}\right]$$

Since $\mathcal{P} = \cup_{\mathcal{M}} \mathcal{P}_{\mathcal{M}}$

$$R(n_l, \mathcal{P}) = \inf_{A} \sup_{\mathcal{M}} \sup_{p \in \mathcal{P}_{\mathcal{M}}} \mathbb{E}_{\overline{z}}\left[\|A(\overline{z}) - m_p\|_{L^2(p_{\mathbf{X}})}\right]$$

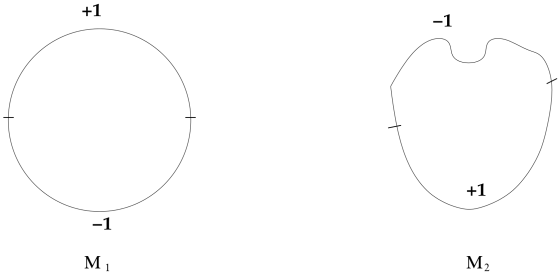
(SSL) When A is allowed to know $\mathcal{M}$

$$Q(n_l, \mathcal{P}) = \sup_{\mathcal{M}} \inf_{A} \sup_{p \in \mathcal{P}_{\mathcal{M}}} \mathbb{E}_{\overline{z}}\left[\|A(\overline{z}) - m_p\|_{L^2(p_{\mathbf{X}})}\right]$$

In which cases there is a gap between $Q(n_l, \mathcal{P})$ and $R(n_l, \mathcal{P})$?

SSL with Graphs: What is behind it?

Hypothesis space $\mathcal{H}$: half of the circle as $+1$ and the rest as -1



Case 1: $\mathcal{M}$ is known to the learner ($\mathcal{H}_{\mathcal{M}}$)

What is a VC dimension of $\mathcal{H}_{\mathcal{M}}$?

$$\text{Optimal rate } Q(n,\mathcal{P}) \le 2\sqrt{\frac{3\log n_l}{n_l}}$$

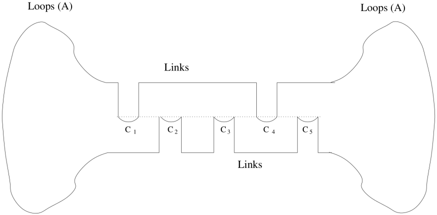
SSL with Graphs: What is behind it?

Case 2: $\mathcal{M}$ is **unknown** to the learner

$$R(n_l, \mathcal{P}) = \inf_{A} \sup_{p \in \mathcal{P}} \mathbb{E}_{\overline{z}} \left[\|A(\overline{z}) - m_p\|_{L^2(p_\mathbf{X})} \right] = \Omega(1)$$

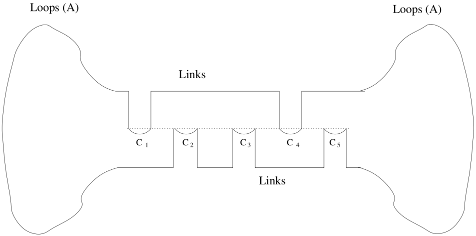
We consider 2^d manifolds of the form

$$\mathcal{M} = \text{Loops} \cup \text{Links} \cup C \text{ where } C = \cup_{i=1}^{d} C_i$$



Main idea: d segments in C, $d - l$ with no data, 2^l possible choices for labels, which helps us to lower bound $\|A(\overline{z}) - m_p\|_{L^2(p_\mathbf{X})}$

SSL with Graphs: What is behind it?



Knowing the manifold helps

- C_1 and C_4 are close
- C_1 and C_3 are far
- we also need: **target function varies smoothly**
- altogether: **closeness on manifold** $\rightarrow$ **similarity in labels**

SSL with Graphs: What is behind it?

What does it mean to **know** $\mathcal{M}$?

Different degrees of knowing $\mathcal{M}$

- set membership oracle: $\mathbf{x} \overset{?}{\in} \mathcal{M}$
- approximate oracle
- knowing the harmonic functions on $\mathcal{M}$
- knowing the Laplacian $\mathcal{L}_{\mathcal{M}}$
- knowing eigenvalues and eigen*functions*
- topological invariants, e.g., dimension
- metric information: geodesic distance

Huge $\mathcal{G}$

when $\mathcal{G}$ does not fit to memory

...or when we can't invert $\mathbf{L}$

Scaling SSL with Graphs to Millions

Semi-supervised learning with graphs

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

Let us see the same in eigenbasis of $\mathbf{L} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\mathsf{T}$, i.e., $\mathbf{f} = \mathbf{U}\boldsymbol{\alpha}$

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

What is the problem with scalability?

Diagonalization of $N \times N$ matrix

What can we do? Let's take only first k eigenvectors $\mathbf{f} = \mathbf{U}\boldsymbol{\alpha}$!

Scaling SSL with Graphs to Millions

$\mathbf{U}$ is now a $n \times k$ matrix

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

Closed form solution is $(\boldsymbol{\Lambda} + \mathbf{U}^\mathsf{T}\mathbf{C}\mathbf{U})\alpha = \mathbf{U}^\mathsf{T}\mathbf{C}\mathbf{y}$

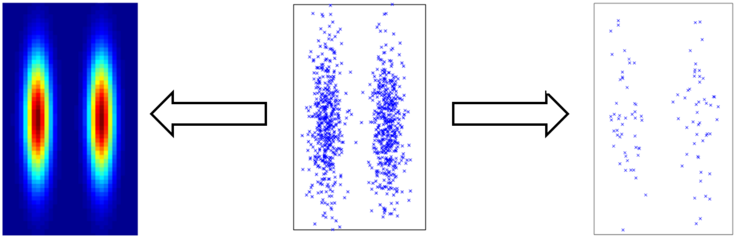
What is the size of this system of equation now?

Cool! Any problem with this approach?

Are there any reasonable assumptions when this is feasible?

Let's see what happens when $N \to \infty$!

Scaling SSL with Graphs to Millions



Density

Data

Landmarks

Limit as n → ∞

Reduce n

Linear in number of data-points

Polynomial in number of landmarks

https://cs.nyu.edu/~fergus/papers/fwt_ssl.pdf

Scaling SSL with Graphs to Millions

What happens to L when $N \to \infty$?

We have data $\mathbf{x}_i \in \mathbb{R}$ sampled from $p(\mathbf{x})$.

When $n \to \infty$, instead of vectors $\mathbf{f}$, we consider functions $F(x)$.

Instead of $\mathbf{L}$, we define $\mathcal{L}_p$ - **weighted smoothness operator**

$$\mathcal{L}_p(F) = \frac{1}{2} \int \left(F(\mathbf{x}_1) - F(\mathbf{x}_2)\right)^2 W(\mathbf{x}_1, \mathbf{x}_2) p(\mathbf{x}_1) p(\mathbf{x}_2) \,\mathrm{d}\mathbf{x}_1 \mathbf{x}_2$$

$$\text{with } W(\mathbf{x}_1, \mathbf{x}_2) = \frac{\exp\left(-\|\mathbf{x}_1 - \mathbf{x}_2\|^2\right)}{2\sigma^2}$$

$\mathbf{L}$ defined the eigenvectors of increasing smoothness.

What defines $\mathcal{L}_p$? **Eigenfunctions!**

Scaling SSL with Graphs to Millions

$$\mathcal{L}_p(F) = \frac{1}{2}\int \left(F(\mathbf{x}_1) - F(\mathbf{x}_2)\right)^2 W(\mathbf{x}_1, \mathbf{x}_2) p(\mathbf{x}_1)\, p(\mathbf{x}_2)\, \mathrm{d}x_1 x_2$$

First eigenfunction

$$\Phi_1 = \operatorname*{arg\,min}_{F: \int F^2(\mathbf{x}) p(\mathbf{x}) D(\mathbf{x})\, \mathrm{d}x = 1} \mathcal{L}_p(F)$$

where $D(\mathbf{x}) = \int_{\mathbf{x}_2} W(\mathbf{x}, \mathbf{x}_2)\, p(\mathbf{x}_2)\, \mathrm{d}\mathbf{x}_2$

What is the solution? $\Phi_1(\mathbf{x}) = 1$ because $\mathcal{L}_p(1) = 0$

How to define Φ_2? same, constraining to be orthogonal to Φ_1

$$\int F(\mathbf{x})\, \Phi_1(\mathbf{x})\, p(\mathbf{x})\, D(\mathbf{x})\, \mathrm{d}x = 0$$

Scaling SSL with Graphs to Millions

Eigenfunctions of $\mathcal{L}_p$

Φ_3 as before, orthogonal to Φ_1 and Φ_2 etc.

How to define eigenvalues? $\lambda_k = \mathcal{L}_p(\Phi_k)$

Relationship to the discrete Laplacian

$$\frac{1}{N^2}\mathbf{f}^\mathsf{T}\mathbf{L}\mathbf{f} = \frac{1}{2N^2}\sum_{ij} W_{ij}(f_i - f_j)^2 \xrightarrow[N\to\infty]{} \mathcal{L}_p(F)$$

http://www.informatik.uni-hamburg.de/ML/contents/people/luxburg/publications/

Luxburg04_diss.pdf
http://arxiv.org/pdf/1510.08110v1.pdf

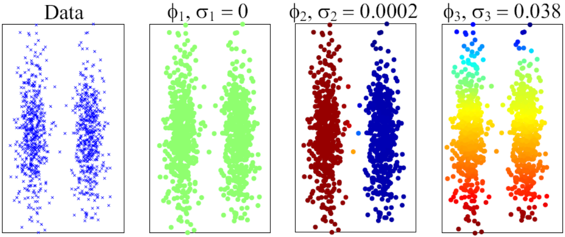
Isn't estimating eigenfunctions $p(\mathbf{x})$ more difficult?

Are there some "easy" distributions?

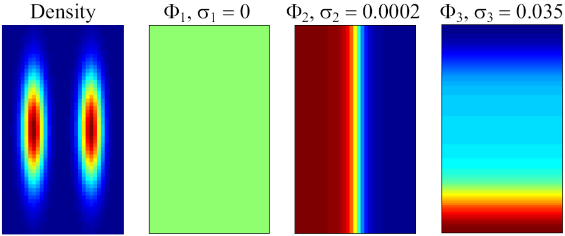
Can we compute it numerically?

Scaling SSL with Graphs to Millions

Eigenvectors



Eigenfunctions

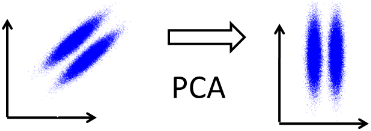


Scaling SSL with Graphs to Millions

Factorized data distribution What if

$$p\left(\mathbf{s}\right) = p\left(s_1\right)p\left(s_2\right)\dots p\left(s_d\right)$$

In general, this is not true. But we can rotate data with $\mathbf{s} = \mathbf{R}\mathbf{x}$.



Treating each factor individually

$p_k \overset{\text{def}}{=}$ marginal distribution of s_k

$\Phi_i\left(s_k\right) \overset{\text{def}}{=}$ eigenfunction of $\mathcal{L}_{p_k}$ with eigenvalue λ_i

Then: $\Phi_i\left(s\right) = \Phi_i\left(s_k\right)$ is eigenfunction of $\mathcal{L}_p$ with λ_i

We only considered single-coordinate eigenfunctions.

Scaling SSL with Graphs to Millions

How to approximate 1D density? Histograms!

Algorithm of Fergus et al. [**fergus2009semi-supervised**] for eigenfunctions

- Find $\mathbf{R}$ such that $\mathbf{s} = \mathbf{R}\mathbf{x}$
- For each "independent" s_k approximate $p(s_k)$
- Given $p(s_k)$ numerically solve for eigensystem of $\mathcal{L}_{p_k}$

$$\left(\widetilde{\mathbf{D}} - \mathbf{P}\widetilde{\mathbf{W}}\mathbf{P}\right)\mathbf{g} = \lambda\mathbf{P}\widehat{\mathbf{D}}\mathbf{g} \qquad \text{(generalized eigensystem)}$$

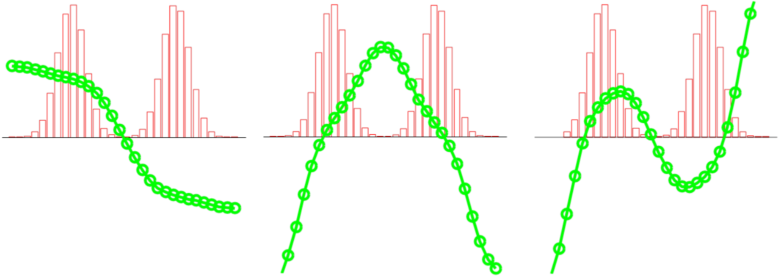
$\mathbf{g}$ - vector of length $B \equiv$ number of bins
$\mathbf{P}$ - density at discrete points
$\widetilde{\mathbf{D}}$ - diagonal sum of the columns of $\mathbf{P}\widetilde{\mathbf{W}}\mathbf{P}$
$\widehat{\mathbf{D}}$ - diagonal sum of the columns of $\mathbf{P}\widetilde{\mathbf{W}}$

- Order eigenfunctions by increasing eigenvalues

https://cs.nyu.edu/~fergus/papers/fwt_ssl.pdf

Scaling SSL with Graphs to Millions

Numerical 1D Eigenfunctions



1st Eigenfunction of $h(x_1)$

2nd Eigenfunction of $h(x_1)$

3rd Eigenfunction of $h(x_1)$

https://cs.nyu.edu/~fergus/papers/fwt_ssl.pdf

Scaling SSL with Graphs to Millions

Computational complexity for $N \times d$ dataset

Typical harmonic approach

one diagonalization of $N \times N$ system

Numerical eigenfunctions with B bins and k eigenvectors

d eigenvector problems of $B \times B$

$$\left(\widetilde{\mathbf{D}} - \mathbf{P}\widetilde{\mathbf{W}}\mathbf{P}\right)\mathbf{g} = \lambda \mathbf{P}\widehat{\mathbf{D}}\mathbf{g}$$

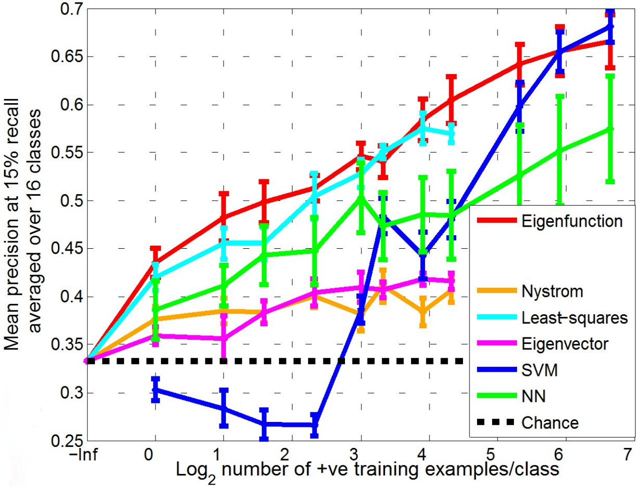
one $k \times k$ least squares problem

$$(\mathbf{\Lambda} + \mathbf{U}^\mathsf{T}\mathbf{C}\mathbf{U})\alpha = \mathbf{U}^\mathsf{T}\mathbf{C}\mathbf{y}$$

some details: several approximation, eigenvectors only linear combinations single-coordinate eigenvectors, ...

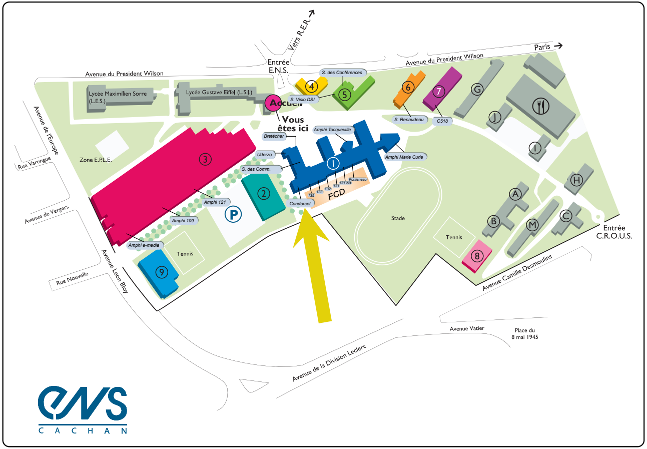
When d is not too big then N can be in millions!

Scaling SSL with Graphs to Millions



CIFAR experiments https://cs.nyu.edu/~fergus/papers/fwt_ssl.pdf

Next lecture: Tuesday, November 19th at 13:30!



Michal Valko

contact via Piazza