# Graphs in Machine Learning

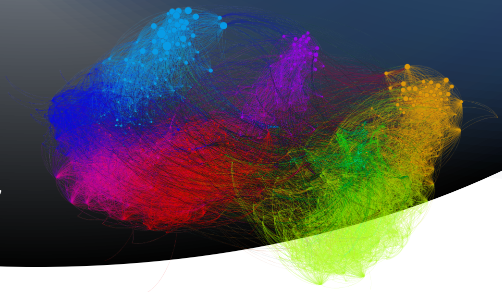
## Online SSL Introduction

Problem Setup and Challenges

Michal Valko

*Inria & ENS Paris-Saclay, MVA*

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



# 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 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**).

Source: Sweeney Todd (2007)

# 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**).

Source: Sweeney Todd (2007)

**Online learning setup**

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

# 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**).

Source: Sweeney Todd (2007)

**Online learning setup**

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

At time $t$:

  receive $\mathbf{x}_t$

# 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**).


Source: Sweeney Todd (2007)

**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 $G(\mathbf{W})$

# 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 $\mathbf{L}_t$

# 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

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

# 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 $\widehat{y}_t =$

# 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**

# 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?

# Online SSL with Graphs

**Online HFS: Straightforward solution**

```
while new unlabeled example x_t comes do
    Add x_t to graph G(W)
    Update L_t
    Infer labels
        f_u = (L_uu + γ_g I)^{-1} (W_ul f_l)
    Predict ŷ_t = sgn(f_u(t))
end while
```

where the inference step is

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

and the prediction is $\widehat{y}_t = \operatorname{sgn}\left(\mathbf{f}_u\left(t\right)\right)$.

What is wrong with this solution?

The cost and memory of the operations.

# 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 $\widehat{y}_t = \operatorname{sgn}(\mathbf{f}_u(t))$
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!

# Online SSL with Graphs

Let's keep only $k$ vertices!

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

# 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.

# 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**.

# 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 $i$.

# 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?

# 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.

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

# 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?

# 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.

# 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

# Online SSL with Graphs

**Online HFS with Graph Quantization**

1: **Input**
2: $k$ number of representative nodes
3: **Initialization**
4: $\mathbf{V}$ matrix of multiplicities with 1 on diagonal

# 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**

# 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 $G(\mathbf{VWV})$

# 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**

# Michal Valko

michal.valko@inria.fr

Inria & ENS Paris-Saclay, MVA

https://misovalko.github.io/mva-ml-graphs.html