Learning From Data
Lecture 14: Semi-Supervised Learning

Yang Li **yangli@sz.tsinghua.edu.cn**

December 30, 2022

Today's Lecture

- What is semi-supervised learning?
- Classical approaches
  - Generative models
  - Semi-supervised SVM
  - Graph-based methods
  - Multiview learning
- Deep semi-supervised learning

Motivation: Some labels are hard to obtain

Supervised learning requires lots of labeled data

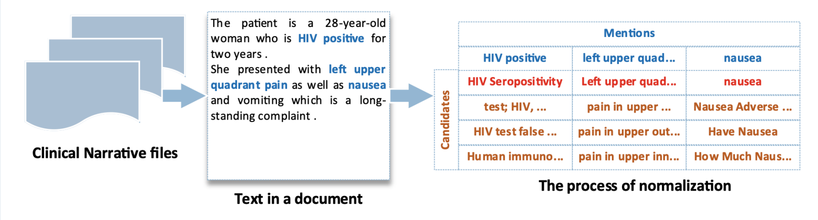
- **Labeled data**: expensive and scarce
- **Unlabeled data**: cheap (or even free)

Motivation: Some labels are hard to obtain

Supervised learning requires lots of labeled data

- **Labeled data**: expensive and scarce
- **Unlabeled data**: cheap (or even free)

e.g. Clinical concept normalization

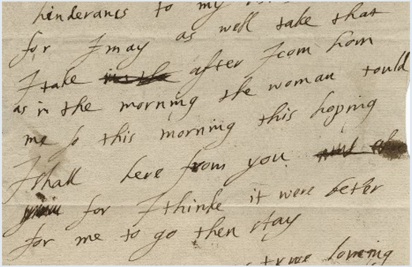


- MCN Corpus (2019): normalize clinical concepts corresponding to medical problems, treatments, and tests
- Manually annotated 3790 concepts and over 13,600 distinct concept mentions.

Motivation: Some labels are hard to obtain

e.g. letter transcription

- Shakespeares transcription


hinderances to my
for I may as well take that
I take ~~in the~~ after I com hom
as in the morning the woman tould
me so this morning this hoping
I shall here from you
for I thinke it were beter
for me to go then stay

for I may as well take that
I take ~~in the~~ after I com hom
as in the morning the woman tould
me so this morning this hoping
I shall here from you ~~and then~~
~~you~~ for I thinke it were better
for me to go then stay

What is Semi-supervised learning?

Semi-supervised learning (SSL) are supervised learning tasks that also make use of unlabeled data for training.

Notations

- Labeled data: $(X_L,\ Y_L) = \{(x^{(1)}, y^{(1)}), (x^{(l)}, y^{(l)})\}$
- Unlabeled data: $X_U = \{x^{(l+1)}, \dots, x^{(m)}\}$, $l + u = m, u \gg l$
- Hypothesis $f : \mathcal{X} \to \mathcal{Y}$

What is Semi-supervised learning?

Semi-supervised learning (SSL) are supervised learning tasks that also make use of unlabeled data for training.

Notations

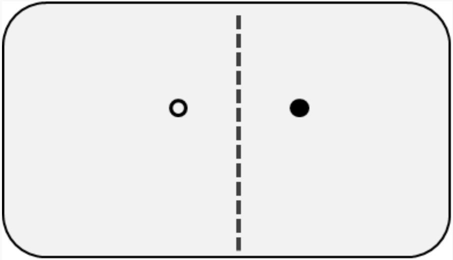
- Labeled data: $(X_L, Y_L) = \{(x^{(1)}, y^{(1)}), (x^{(l)}, y^{(l)})\}$
- Unlabeled data: $X_U = \{x^{(l+1)}, \ldots, x^{(m)}\}$, $l + u = m, u \gg l$
- Hypothesis $f : \mathcal{X} \to \mathcal{Y}$

Two types of SSL:

- **Transductive** semi-supervised learning finds the hypothesis f that best classify the unlabeled data X_U
- **Inductive** semisupervised learning learns a hypothesis f for future data (not in $X_U \cup X_L$).
  f should be better than using X_L alone.

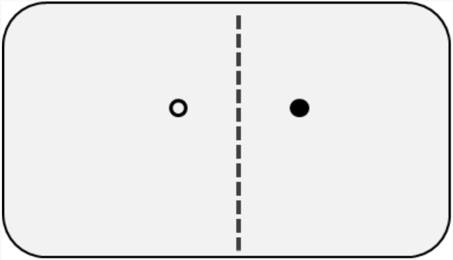
How does unlabeled data help?

Hypothesis function using labeled data:

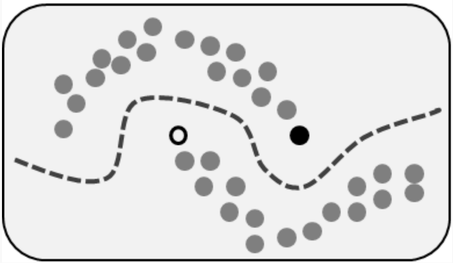


How does unlabeled data help?

Hypothesis function using labeled data:



Hypothesis function using both labeled and unlabeled data:



Semi-supervise learning assumptions

Semi-supervise learning algorithms rely on one of the following assumptions:

Semi-supervise learning assumptions

Semi-supervise learning algorithms rely on one of the following assumptions:

Smoothness assumption: If two data samples are similar, then output labels should be similar.

Cluster assumption: Samples in the same cluster are more likely to have the same label. i.e. low-density separation between classes A special case of the smoothness assumption

Semi-supervise learning assumptions

Semi-supervise learning algorithms rely on one of the following assumptions:

Smoothness assumption: If two data samples are similar, then output labels should be similar.

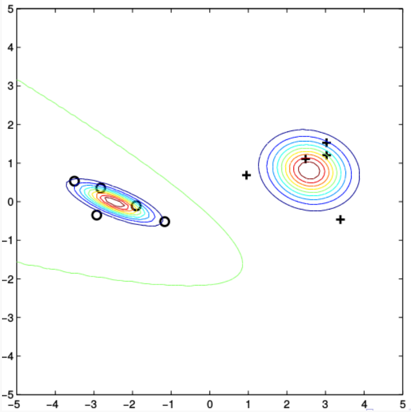
Cluster assumption: Samples in the same cluster are more likely to have the same label. i.e. low-density separation between classes A special case of the smoothness assumption

Manifold assumption: Data lie approximately on a manifold of dimension $\ll n$. This allows us to use distances on the manifold

Generative models

Using unlabeled data in generative models

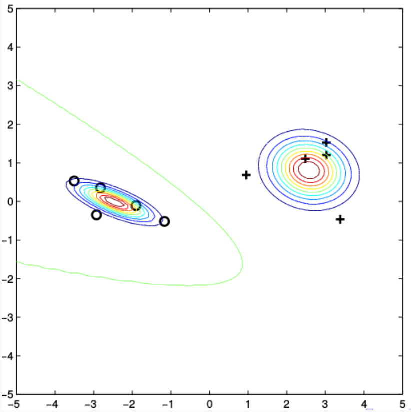
Example: guassian discriminant model



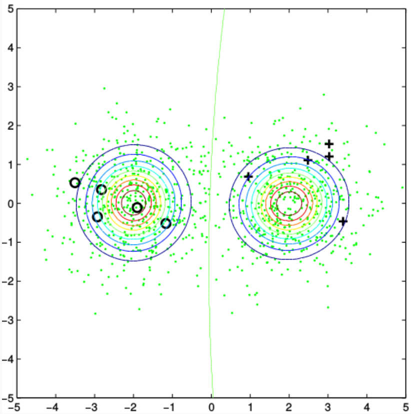
without unlabeled data

Using unlabeled data in generative models

Example: guassian discriminant model



without unlabeled data



with unlabeled data

Notice the difference in the decision boundaries

Supervised Generative Models

Given random variables $x \in \mathcal{X}$, $y \in \mathcal{Y}$, assume that

- class prior distribution $p(y;\theta)$
  e.g. $y \sim \text{Multinomial}(\phi)$
- data generating distribution $p(x|y;\theta)$
  e.g. $x|y \sim N(\mu, \Sigma)$

Supervised Generative Models

Given random variables $x \in \mathcal{X}$, $y \in \mathcal{Y}$, assume that

- class prior distribution $p(y; \theta)$
  e.g. $y \sim \text{Multinomial}(\phi)$
- data generating distribution $p(x|y; \theta)$
  e.g. $x|y \sim N(\mu, \Sigma)$

A generative model computes the joint probability as

$$p(x, y; \theta) = p(x|y; \theta)p(y; \theta)$$

Supervised Generative Models

Given random variables $x \in \mathcal{X}$, $y \in \mathcal{Y}$, assume that

- class prior distribution $p(y;\theta)$
  e.g. $y \sim \text{Multinomial}(\phi)$
- data generating distribution $p(x|y;\theta)$
  e.g. $x|y \sim N(\mu, \Sigma)$

A generative model computes the joint probability as

$$p(x, y;\theta) = p(x|y;\theta)p(y;\theta)$$

Classifier using Baye's rule:

$$\begin{aligned} p(y|x;\theta) &= \frac{p(x|y;\theta)p(y;\theta)}{p(x;\theta)} \\ &= \frac{p(x|y;\theta)p(y;\theta)}{\sum_{y'} p(x|y';\theta)p(y';\theta)} \end{aligned}$$

Training Generative Models

Given data $(x^{(1)}, y^{(1)}) \dots (x^{(m)}, y^{(m)})$, θ can be estimated using maximum likelihood:

$$\operatorname*{argmax}_{\theta} \log \prod_{i=1}^{m} p(x^{(i)}, y^{(i)}; \theta)$$

Training Generative Models

Given data $(x^{(1)}, y^{(1)}) \dots (x^{(m)}, y^{(m)})$, θ can be estimated using maximum likelihood:

$$\operatorname*{argmax}_{\theta} \log \prod_{i=1}^{m} p(x^{(i)}, y^{(i)}; \theta)$$

Alternative ways to learn θ:

- MAP estimator
- Bayesian estimator

Semi-supervised Generative Model

Given labeled data $(x^{(1)}, y^{(1)}), \ldots, (x^{(l)}, y^{(l)})$, and unlabeled data $x^{(l+1)}, \ldots, x^{(l+u)}$

Maximimum likelihood estimation of θ:

$$\operatorname*{argmax}_{\theta} \underbrace{\log \prod_{i=1}^{l} p(x^{(i)}, y^{(i)}; \theta)}_{\text{labeled data}} + \lambda \underbrace{\log \prod_{i=l+1}^{l+u} p(x^{(i)}; \theta)}_{\text{unlabeled data}}$$

Semi-supervised Generative Model

Given labeled data $(x^{(1)}, y^{(1)}), \dots, (x^{(l)}, y^{(l)})$, and unlabeled data $x^{(l+1)}, \dots, x^{(l+u)}$

Maximimum likelihood estimation of θ:

$$\underset{\theta}{\operatorname{argmax}} \underbrace{\log \prod_{i=1}^{l} p(x^{(i)}, y^{(i)}; \theta)}_{\text{labeled data}} + \lambda \underbrace{\log \prod_{i=l+1}^{l+u} p(x^{(i)}; \theta)}_{\text{unlabeled data}}$$

where

$$\log \prod_{i=l+1}^{l+u} p(x^{(i)}; \theta) = \sum_{i=l+1}^{l+u} \log p(x^{(i)}; \theta) = \sum_{i=l+1}^{l+u} \log \sum_{y \in \mathcal{Y}} p(x^{(i)}, y; \theta)$$

is typically *non-concave*. We can only find local optimal solutions.

Training semi-supervised generative model

Treat unknown labels $y^{(l)}, \dots, y^{(l+u)}$ as hidden variables.

An EM algorithm

- Initialize θ randomly
- Repeat until convergence$\{$

E-step ► Compute $Q_i(y^{(i)}) = p(y|x^{(i)};\theta)$ for all $i = l+1, \dots, l+u$

M-step ► Update θ using full data (X_l, X_u)

$\}$

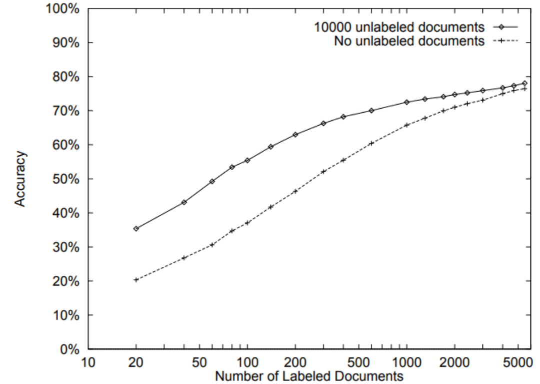
Application: Document classification

20 Newsgroup Dataset

- X_L: 10000 unlabeled documents
- X_U: 20-5000 labeled documents
- $y \in 1, \ldots, 20$ topics

Generative model

- Naive bayes model
- MAP estimator



K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. Machine Learning, 39, 2000.

Generative model assumptions

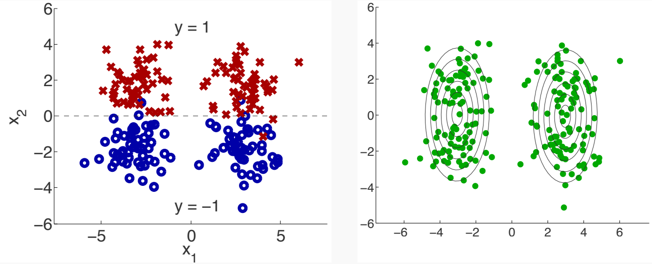
Generative model works well when the model choice is correct.
e.g. for a mixture model,

- Cluster assumption: data in the same class lie in a cluster, which is separated from other clusters
- The # of clusters = number of classes

Generative model assumptions

Generative model works well when the model choice is correct.
e.g. for a mixture model,

- Cluster assumption: data in the same class lie in a cluster, which is separated from other clusters
- The # of clusters = number of classes

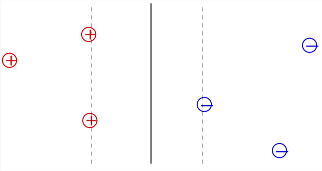


Example of incorrect assumption

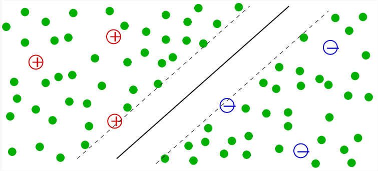
Semi-Supervised SVM

Semi-Supervised SVM

- Unlabeled data from different classes are separated by *large margin*
- *Idea: The decision boundary shouldn't lie in the regions of high density $p(x)$*



without unlabeled data



with unlabeled data

Review: Soft-Margin SVM

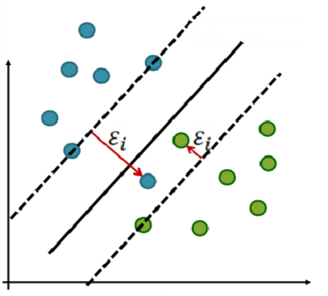
Given training data $(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})$

Train a soft-margin SVM classifier:

$$\min_{w,b,\xi} \frac{1}{2}||w||^2 + C\sum_{i=1}^{m}\xi_i$$

$$s.t.\ y^{(i)}(w^T x^{(i)} + b) \geq 1 - \xi_i$$

$$\xi_i \geq 0, i = 1, \ldots, m$$



Can be solved using quadratic programming.

Semi-Supervised SVM

Optimization variables:

- Estimated label for unlabeled data: $\{\hat{y}^{l+1}, \ldots, \hat{y}^{l+u}\}$
- Margin of labeled data: $\{\xi_1, \ldots, \xi_l\}$
- Margin of unlabeled data: $\{\hat{\xi}_{l+1}, \ldots, \hat{\xi}_{l+u}\}$

$$\min_{w,b,\{\epsilon_i\},\{\hat{\epsilon}_j\},\{\hat{y}_j\}} \frac{1}{2}||w||_2^2 + C\sum_{i=1}^{l}\xi_i + C'\sum_{j=l+1}^{l+u}\hat{\xi}_j$$

$$\begin{aligned} \text{s.t. } & (w^T x^{(i)} + b)y^{(i)} \geq 1 - \xi_i \quad \forall i = 1, \ldots, l \\ & (w^T x^{(j)} + b)\hat{y}^{(j)} \geq 1 - \hat{\xi}_j \quad \forall j = l+1, \ldots, l+u \\ & \hat{y}^{(j)} \in \{-1, 1\} \quad \forall j = l+1, \ldots, l+u \end{aligned}$$

T. Joachims. Transductive inference for text classification using support vector machines. In Proc. 16th International Conf. on Machine Learning, p200209. 1999

Semi-Supervised SVM Discussion

Numerical optimization

- Semi-supervised SVM is an integer programming problem: NP-hard
- Approximated solutions are used in practice

Low-Density Assumption

- Decision boundary should lie in a low density region
- Equivalent to the cluster assumption

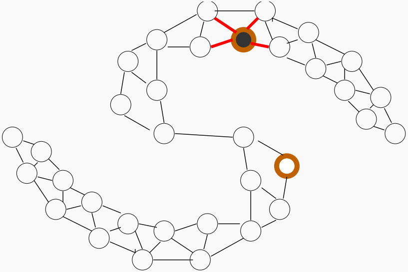
Graph-based Methods

Label propagation idea

Main idea

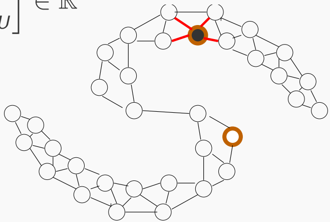
- Build a graph connecting data points $x^{(1)}, \ldots, x^{(m)}$
- Assign weights to edges according to similarity measure $s(x^{(i)}, x^{(j)})$
- Propagate labels from labeled points forward to unlabeled points

Label propagation is a **transductive** algorithm.



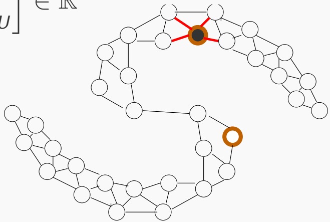
Label Propagation: Iterative Approach

Node labels: $Y = \begin{bmatrix} Y_L \\ Y_U \end{bmatrix} \in \mathbb{R}^{m \times n}$



Label Propagation: Iterative Approach

Node labels: $Y = \begin{bmatrix} Y_L \\ Y_U \end{bmatrix} \in \mathbb{R}^{m \times n}$



Define T to be the $m \times m$ transition matrix that realizes the propagation of labels:

```
1.   Initialize Y^0 = [Y_L ; 0]
2.   Repeat until convergence {
3.       Y^t = T Y^{t-1}
4.       Clamp the labeled data Y_L^t = Y_L
5.   }
```

Label propagation: analytical solution

Write the transition step as block matrices:

$$Y = TY$$
$$\begin{bmatrix} Y_L \\ Y_U \end{bmatrix} = \begin{bmatrix} T_{LL} & T_{LU} \\ T_{UL} & T_{UU} \end{bmatrix} \begin{bmatrix} Y_L \\ Y_U \end{bmatrix}$$

We can solve for the unknown labels Y_U:

$$Y_U = T_{UL} Y_L + T_{UU} Y_U$$
$$Y_U = (I - T_{UU})^{-1} T_{UL} Y_L$$

assuming that $(I - T_{UU})^{-1}$ is invertible.

How to find T?

How to find T?

Gaussian similarity:

$$W_{i,j} = \exp\left(-\frac{||x^{(i)} - x^{(j)}||_2^2}{2\sigma^2}\right) \text{ for } i,j = 1,\ldots,m$$

Let $D = diag(W\mathbf{1})$ be the degree matrix

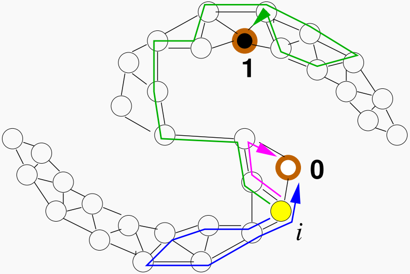
$$D = \begin{bmatrix} \sum_{j=1}^n w_{1j} & 0 & \ldots & 0 \\ 0 & \sum_{j=1}^n w_{2j} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \sum_{j=1}^n w_{mj} \end{bmatrix}$$

Define $T = D^{-1}W$ $\leftarrow$ *$I - L_{rm}$ where L_{rm} is the normalized Laplacian!*

$$T_{ij} = \frac{w_{ij}}{\sum_{l=1}^n w_{il}} \leftarrow \textit{is the transition probability from point } i \textit{ to } j$$

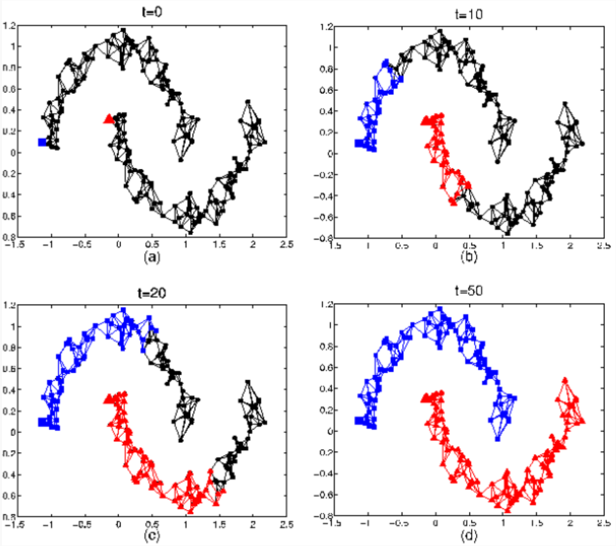
$$Y_u = (I - T_{UU})^{-1} T_{UL} Y_L = (D_U - W_{UU})^{-1} W_{UL} Y_L \tag{1}$$

Interpretation of $T = D^{-1}W$: Random Walk



- Randomly walk from unlabeled node i to j with probability $T_{ij} = \frac{w_{ij}}{\sum_{l=1}^{n} w_{il}}$
- Stop if we hit a labeled node
- The label function $Y_{ic} = Pr($ hit label $c \mid$ start from $i)$

Iterative label propagation example



Label propagation as an optimization problem

Let random vector $y_i \in R^n$ represent the label for data i
We can solve label propagation by

$$\min_{y_i, i \in U} \frac{1}{2} \sum_{i,j=1}^{m} W_{ij} ||y_i - y_j||^2$$

- Minimize the distance between class membership vectors based on weight similarity
  - W_{ij} is very large: need to ensure $||y_i - y_j||^2$ is small
  - W_{ij} is very small: $||y_i - y_j||^2$ is not constrained
- Equivalent to iterative solution $Y_u = (D_U - W_{UU})^{-1} W_{UL} Y_L$

Label Propagation

Let $L = D - W$ be the unnormalized graph laplacian of G.

Lemma 1

$\min_{y_i, i \in U} \frac{1}{2} \sum_{i,j=1}^{m} W_{ij} ||y_i - y_j||^2$ *is equivalent to* $\min_{Y_U} tr(Y^T L Y)$

Theorem 1

The optimal solution to $\min_{y_i, i \in U} \frac{1}{2} \sum_{i,j=1}^{m} W_{ij} ||y_i - y_j||^2$ *is*
$Y_u = (D_U - W_{UU})^{-1} W_{UL} Y_L$

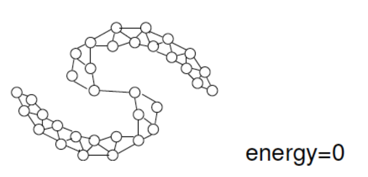
Proofs can be found in:
Bodó, Zalán, and Lehel Csató. A note on label propagation for semi-supervised learning. Acta Universitatis Sapientiae, Informatica 7, no. 1: 18-30, 2015.

Inductive semi-supervised learning

- Goal: Learn a better predictor $f : \mathcal{X} \rightarrow \mathcal{Y}$ using unlabeled data X_U
- In graph-based learning, a large W_{ij} implies a preference for $f(x^{(i)}) = f(x^{(j)})$, represented by an energy function :

$$\sum_{i,j}^{m} W_{ij}(f(x^{(i)}) - f(x^{(j)}))^2 \qquad (*)$$

Example: no labeled data



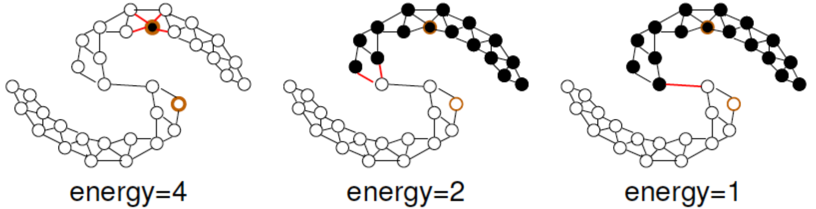
The top-ranked (smoothest) hypothesis is $f(x) = 1$ or $f(x) = 0$

Inductive semi-supervised learning

- Goal: Learn a better predictor $f : \mathcal{X} \to \mathcal{Y}$ using unlabeled data X_U
- In graph-based learning, a large W_{ij} implies a preference for $f(x^{(i)}) = f(x^{(j)})$, represented by an energy function :

$$\sum_{i,j}^{m} W_{ij}(f(x^{(i)}) - f(x^{(j)}))^2 \qquad (*)$$

Example: conditioned on labeled data



Find f that both fits the labeled data well and ranks high (being smooth on the graph or underlying manifold.

$$\operatorname*{argmin}_{f \in \mathcal{F}} \underbrace{\frac{1}{l} \sum_{i=1}^{l} \mathcal{L}(f(x^{(i)}), y^{(i)}) + \lambda_1 ||f||^2}_{\text{supervised loss}} + \underbrace{\lambda_2 \sum_{i,j=1}^{m} W_{ij}(f(x^{(i)}) - f(x^{(j)}))^2}_{\text{regularization of } X_U}$$

- $\mathcal{L}$ is a convex loss function, e.g. hinge-loss, squared loss
- This problem is convex with efficient solvers

Find f that both fits the labeled data well and ranks high (being smooth on the graph or underlying manifold.

$$\underset{f \in \mathcal{F}}{\operatorname{argmin}} \underbrace{\frac{1}{l} \sum_{i=1}^{l} \mathcal{L}(f(x^{(i)}), y^{(i)}) + \lambda_1 ||f||^2}_{\text{supervised loss}} + \underbrace{\lambda_2 \sum_{i,j=1}^{m} W_{ij}(f(x^{(i)}) - f(x^{(j)}))^2}_{\text{regularization of } X_U}$$

- $\mathcal{L}$ is a convex loss function, e.g. hinge-loss, squared loss
- This problem is convex with efficient solvers

By Lemma 1, it can be written as

$$\underset{f \in \mathcal{F}}{\operatorname{argmin}} \frac{1}{l} \sum_{i=1}^{l} \mathcal{L}(f(x^{(i)}), y^{(i)}) + \lambda_1 ||f||^2 + \lambda_2 tr(f^T L f)$$

Algorithm variations: graph min-cut, manifold regularization, etc

Further readings on inductive graph-based semi-supervised learning:

- M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. Journal of Machine Learning Research, 7:23992434, November 2006.
- A. Blum and S. Chawla. Learning from labeled and unlabeled data using graph mincuts. In Proc. 18th International Conf. on Machine Learning, 2001.

Graph-based semi-supervised learning discussion

When to use graph-based SSL?

- SSL only works well when the underlying assumptions hold on the data
- Constructing a good graph is important!

Transductive vs inductive?

- Transductive: predict labels on the unlabeled data (known at training time)
- Inductive: predict labels for future (unseen) data

Multiview Learning

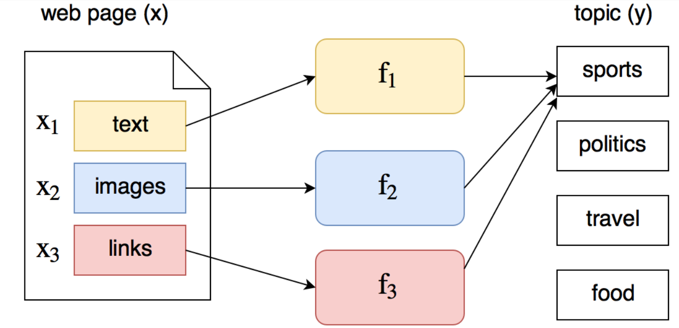
Example: Web page classification

Multiview learning assumptions:

- Multiple learners are trained on the same labeled data
- Learners agree on the unlabeled data

e.g. A web page has multiple subsets of features, or **views**

$$x = \langle x_1, x_2, x_3 \rangle$$



Multiview semi-supervised learning

Let $f_1, \ldots, f_k$ be the hypothesis function on k views.
The **disagreement** of hypothesis tuple $\langle f_1, \ldots, f_k \rangle$ on the unlabeled data can be defined as

$$\sum_{i=l+1}^{l+u} \sum_{u,v}^{k} \mathcal{L}(f_u(x^{(i)}), f_v(x^{(i)}))$$

Multiview semi-supervised learning

Let $f_1, \ldots, f_k$ be the hypothesis function on k views.
The **disagreement** of hypothesis tuple $\langle f_1, \ldots, f_k \rangle$ on the unlabeled data can be defined as

$$\sum_{i=l+1}^{l+u} \sum_{u,v}^{k} \mathcal{L}(f_u(x^{(i)}), f_v(x^{(i)}))$$

Common loss function $\mathcal{L}$

- 0-1 loss (discrete y)

$$\mathcal{L}(f_u(x^{(i)}), f_v(x^{(i)})) = \begin{cases} 1 & \text{if } f_u(x^{(i)}) = f_v(x^{(i)}) \\ 0 & \text{otherwise} \end{cases}$$

- Squared error (continuous y)

$$\mathcal{L}(f_u(x^{(i)}), f_v(x^{(i)})) = ||f_u(x^{(i)}) - f_v(x^{(i)})||^2$$

Multiview semi-supervised learning

$$\mathcal{L}(f_1, \ldots, f_k) = \sum_{u=1}^{k} \underbrace{\left(\frac{1}{l} \sum_{i=1}^{l} \mathcal{L}_u(f_u(x^{(i)}), y^{(i)}) + \lambda \Omega_u(f_u) \right)}_{\text{regularized empirical risk on labeled data}}$$
$$+ \underbrace{\sum_{i=l+1}^{l+u} \sum_{u,v}^{k} \mathcal{L}(f_u(x^{(i)}), f_v(x^{(i)}))}_{\text{disagreement on unlabeled data}}$$

where $\mathcal{L}_u$ is the loss of view u.

Multiview semi-supervised learning

$$\mathcal{L}(f_1, \dots, f_k) = \sum_{u=1}^{k} \underbrace{\left(\frac{1}{l} \sum_{i=1}^{l} \mathcal{L}_u(f_u(x^{(i)}), y^{(i)}) + \lambda \Omega_u(f_u) \right)}_{\text{regularized empirical risk on labeled data}}$$

$$+ \underbrace{\sum_{i=l+1}^{l+u} \sum_{u,v}^{k} \mathcal{L}(f_u(x^{(i)}), f_v(x^{(i)}))}_{\text{disagreement on unlabeled data}}$$

where $\mathcal{L}_u$ is the loss of view u.

To find the optimal hypothesis:

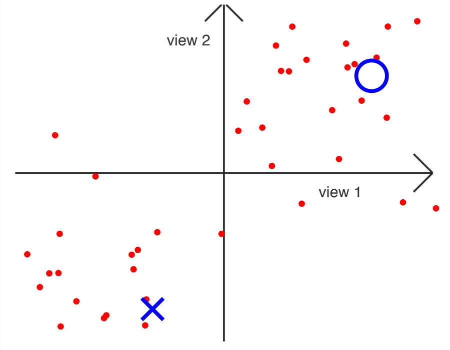
$$\operatorname*{argmin}_{f_1, \dots, f_k} \mathcal{L}(f_1, \dots, f_k)$$

When $\mathcal{L}_u, \Omega_u$ and $\mathcal{L}$ and are all convex, numerical solution can easily be obtained.

Multiview learning discussion

Independent view assumption: there exists subsets of features (views), each of which

- is independent of other views given the class
- is sufficient for classification



V. Sindhwani, P. Niyogi, and M. Belkin. A co-regularized approach to semi-supervised learning with multiple views. In Proc. of the 22nd ICML Workshop on Learning with Multiple Views, August 2005.

Deep Semi-Supervised Learning

Deep Semi-Supervised Learning

Main categories of recent deep semi-supervised methods:

- **Proxy-label method:** leverage a trained model on the labeled data to produce additional training examples by labeling unlabeled samples based on some heuristics. *e.g. self-training, pseudo-labeling*

Deep Semi-Supervised Learning

Main categories of recent deep semi-supervised methods:

- **Proxy-label method:** leverage a trained model on the labeled data to produce additional training examples by labeling unlabeled samples based on some heuristics. *e.g. self-training, pseudo-labeling*
- **Consistency regularity:** assumes that when a perturbation was applied to the unlabeled data points, the prediction should not change significantly *e.g.* Π*-Model, Mixup*

Deep Semi-Supervised Learning

Main categories of recent deep semi-supervised methods:

- **Proxy-label method:** leverage a trained model on the labeled data to produce additional training examples by labeling unlabeled samples based on some heuristics. *e.g. self-training, pseudo-labeling*
- **Consistency regularity:** assumes that when a perturbation was applied to the unlabeled data points, the prediction should not change significantly *e.g.* Π*-Model, Mixup*
- **Graph-based approaches:** use label propagation on unlabeled data with supervised deep feature embedding *e.g. GNN based methods*

Deep Semi-Supervised Learning

Main categories of recent deep semi-supervised methods:

- **Proxy-label method:** leverage a trained model on the labeled data to produce additional training examples by labeling unlabeled samples based on some heuristics. *e.g. self-training, pseudo-labeling*
- **Consistency regularity:** assumes that when a perturbation was applied to the unlabeled data points, the prediction should not change significantly *e.g. Π-Model, Mixup*
- **Graph-based approaches:** use label propagation on unlabeled data with supervised deep feature embedding *e.g. GNN based methods*
- **Generative models:** estimate the input distribution $p(x)$ from unlabeled data in addition to classification *(VAE or GAN based methods)*

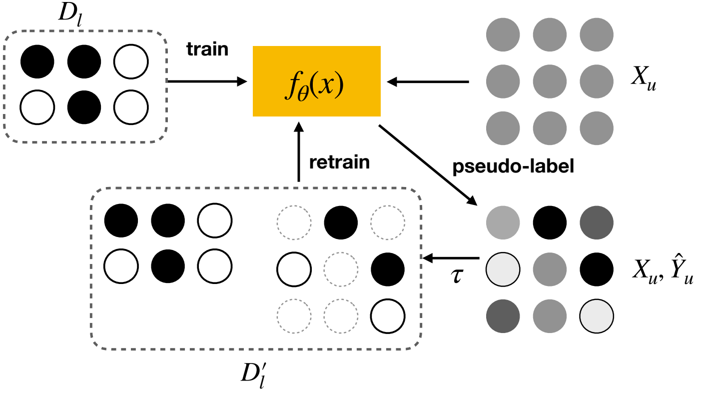
Deep Semi-Supervised Learning

Main categories of recent deep semi-supervised methods:

- **Proxy-label method:** leverage a trained model on the labeled data to produce additional training examples by labeling unlabeled samples based on some heuristics. *e.g. self-training, pseudo-labeling*
- **Consistency regularity:** assumes that when a perturbation was applied to the unlabeled data points, the prediction should not change significantly *e.g. Π-Model, Mixup*
- **Graph-based approaches:** use label propagation on unlabeled data with supervised deep feature embedding *e.g. GNN based methods*
- **Generative models:** estimate the input distribution $p(x)$ from unlabeled data in addition to classification *(VAE or GAN based methods)*
- **Hybrid approaches:** combining multiple techniques *e.g. MixMatch*

Proxy-Label Methods
Pseudo-labeling

- Use labeled data $D_l = \{X_l, Y_l\}$ to train a prediction function f_θ
- Assign pseudo-labels $\hat{y} = \text{argmax}\, f_\theta(x)$ to each unlabeled sample $x \in X_u$. *$f_\theta(x_u)$ is a probability distribution over classes $\mathcal{Y}$*
- add $(x, \hat{y})$ to D_l if $\max f_\theta(x) > \tau$ for some threshold $\tau > 0$



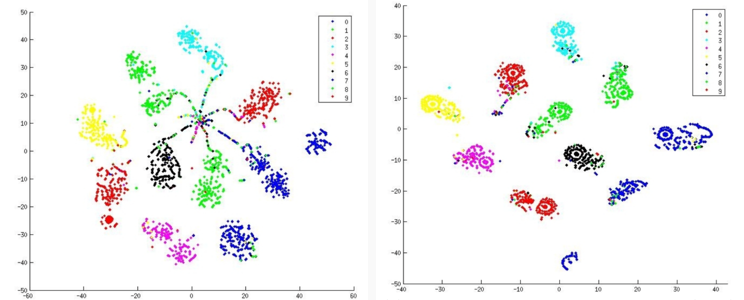
Pseudo-label example

Lee, Dong-Hyun. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In Workshop on challenges in representation learning, ICML, 2013.

Overall loss function:

$$L = \frac{1}{n}\sum_{m=1}^{n}\sum_{i=1}^{C} L(y_i^m, f_i^m) + \alpha(t)\frac{1}{n'}\sum_{m=1}^{n'}\sum_{i=1}^{C} L(y_i'^m, f_i'^m)$$

Proper scheduling of $\alpha(t)$ is important for network performance!



(a) without unlabeled data (dropNN)

(b) with unlabeled data and Pseudo-Label (+PL)

Feature embedding results on MNIST

Consistency regularization

- Favoring functions f_θ that give **consistent predictions for similar data points**. $\leftarrow$ *clustering assumption*
- Given unlabeled sample $x \in X_u$ and its perturbed version $\hat{x}$
- Minimize the distance between the two outputs $d(f_\theta(x), f_\theta(\hat{x}))$

Consistency regularization

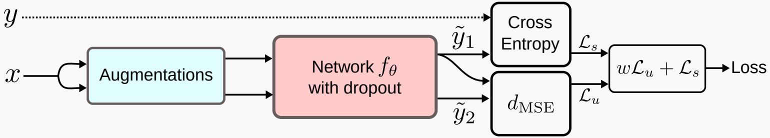
- Favoring functions f_θ that give **consistent predictions for similar data points**. $\leftarrow$ *clustering assumption*
- Given unlabeled sample $x \in X_u$ and its perturbed version $\hat{x}$
- Minimize the distance between the two outputs $d(f_\theta(x), f_\theta(\hat{x}))$
- Common distance functions:

$$d_{MSE}(f_\theta(x), f_\theta(\hat{x}) = \frac{1}{C}\sum_{j=1}^{C}(f_\theta(x)_j - f_\theta(\hat{x})_j)^2$$

$$d_{KL}(f_\theta(x), f_\theta(\hat{x})) = \frac{1}{C}\sum_{j=1}^{C} f_\theta(x)_j \log \frac{f_\theta(x)_j}{f_\theta(\hat{x})_j}$$

Consistency Regularization Example: Π-Model

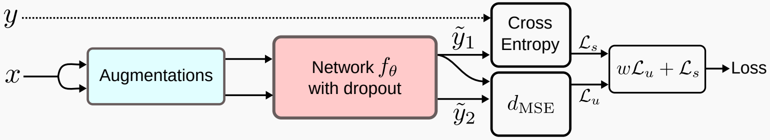
Laine, Samuli, and Timo Aila. "Temporal ensembling for semi-supervised learning." arXiv preprint arXiv:1610.02242 (2016).



- Perturb each input x by random augmentations (e.g. image translation, flipping, rotations etc) and random dropout to obtain distinct predictions $\tilde{y}_1, \tilde{y}_2$

Consistency Regularization Example: Π-Model

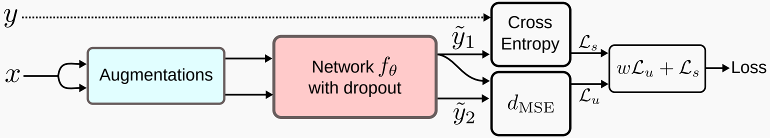
Laine, Samuli, and Timo Aila. "Temporal ensembling for semi-supervised learning." arXiv preprint arXiv:1610.02242 (2016).



- Perturb each input x by random augmentations (e.g. image translation, flipping, rotations etc) and random dropout to obtain distinct predictions $\tilde{y}_1, \tilde{y}_2$
- Enforce a consistency over two perturbed versions of x by $L_u = d_{MSE}(\tilde{y}_1 - \tilde{y}_2)$

Consistency Regularization Example: Π-Model

Laine, Samuli, and Timo Aila. "Temporal ensembling for semi-supervised learning." arXiv preprint arXiv:1610.02242 (2016).



- Perturb each input x by random augmentations (e.g. image translation, flipping, rotations etc) and random dropout to obtain distinct predictions $\tilde{y}_1, \tilde{y}_2$
- Enforce a consistency over two perturbed versions of x by $L_u = d_{MSE}(\tilde{y}_1 - \tilde{y}_2)$
- If $x \in X_l$, minimize the cross-entropy loss $\mathcal{L}_l(y, f(x))$

$$\mathcal{L} = w \frac{1}{|D_u|} \sum_{x \in D_u} d_{MSE}(\tilde{y}_1, \tilde{y}_2) + \frac{1}{|D_l|} \sum_{x,y \in D_l} \mathcal{L}_l(y, f(x))$$

w is set to zero for the first 20% training time

Semi-supervised learning summary

Approach	**Assumptions**	**Type**
Graph-based	manifold assumption	transductive , inductive
Generative model	cluster assumption	inductive
SVM	low density separation/cluster assumption	inductive
Multi-view learning	independent view assumption	inductive
Proxy-label	manifold assumption	inductive
Consistency regularization	cluster assumption	inductive

Online poster session information

- Submit your posters by **Jan 3, 2023**
- All teams will be divided into 4 tracks. Your poster will be shared online for pair-review and voting by other teams within your track, starting from Jan 5th.
- Each team will deliver a 3-min presentation for the poster on Jan 6, 2023.
- Prizes available for the best presenter, best poster and most impactful work!

Detailed grading policy will be posted later.