Semi-supervised Learning

Piyush Rai

Machine Learning (CS771A)

Oct 28, 2016

Labeled vs Unlabeled Data

- Supervised Learning models require labeled data

Labeled vs Unlabeled Data

- Supervised Learning models require labeled data
- Learning a reliable model usually requires plenty of labeled data

Labeled vs Unlabeled Data

- Supervised Learning models require labeled data
- Learning a reliable model usually requires plenty of labeled data
- Labeled Data: Expensive and Scarce (someone has to do the labeling)

Labeled vs Unlabeled Data

- Supervised Learning models require labeled data
- Learning a reliable model usually requires plenty of labeled data
- Labeled Data: Expensive and Scarce (someone has to do the labeling)
  - Often labeling is very difficult too (e.g., in speech analysis or NLP problems)

Labeled vs Unlabeled Data

- Supervised Learning models require labeled data
- Learning a reliable model usually requires plenty of labeled data
- Labeled Data: Expensive and Scarce (someone has to do the labeling)
  - Often labeling is very difficult too (e.g., in speech analysis or NLP problems)
- Unlabeled Data: Abundant and Free/Cheap

Labeled vs Unlabeled Data

- Supervised Learning models require labeled data
- Learning a reliable model usually requires plenty of labeled data
- Labeled Data: Expensive and Scarce (someone has to do the labeling)
  - Often labeling is very difficult too (e.g., in speech analysis or NLP problems)
- Unlabeled Data: Abundant and Free/Cheap
  - E.g., can easily crawl the web and download webpages/images

Labeled vs Unlabeled Data

- Supervised Learning models require labeled data
- Learning a reliable model usually requires plenty of labeled data
- Labeled Data: Expensive and Scarce (someone has to do the labeling)
    - Often labeling is very difficult too (e.g., in speech analysis or NLP problems)
- Unlabeled Data: Abundant and Free/Cheap
    - E.g., can easily crawl the web and download webpages/images

Learning with Labeled+Unlabeled Data

- Usually such problems come in one of the following two flavors

Learning with Labeled+Unlabeled Data

- Usually such problems come in one of the following two flavors
- Semi-supervised Learning
  - Training data contains both labeled $\mathcal{L} = \{\boldsymbol{x}_i, y_i\}_{i=1}^{L}$ and unlabeled data $\mathcal{U} = \{\boldsymbol{x}_j\}_{j=L+1}^{L+U}$ (usually $U \gg L$). Note: $\mathcal{U}$ isn't the test data (if it is then the problem is known as "transductive learning").

Learning with Labeled+Unlabeled Data

- Usually such problems come in one of the following two flavors
- Semi-supervised Learning
  - Training data contains both labeled $\mathcal{L} = \{\boldsymbol{x}_i, y_i\}_{i=1}^{L}$ and unlabeled data $\mathcal{U} = \{\boldsymbol{x}_j\}_{j=L+1}^{L+U}$ (usually $U \gg L$). Note: $\mathcal{U}$ isn't the test data (if it is then the problem is known as "transductive learning").
  - Want to learn a classification/regression model combining both data sources

Learning with Labeled+Unlabeled Data

- Usually such problems come in one of the following two flavors
- Semi-supervised Learning
  - Training data contains both labeled $\mathcal{L} = \{\boldsymbol{x}_i, y_i\}_{i=1}^{L}$ and unlabeled data $\mathcal{U} = \{\boldsymbol{x}_j\}_{j=L+1}^{L+U}$ (usually $U \gg L$). Note: $\mathcal{U}$ isn't the test data (if it is then the problem is known as "transductive learning").
  - Want to learn a classification/regression model combining both data sources
- Semi-Unsupervised Learning
  - We are given unlabeled data $\mathcal{D} = \{\boldsymbol{x}_i\}_{i=1}^{N}$

Learning with Labeled+Unlabeled Data

- Usually such problems come in one of the following two flavors
- Semi-supervised Learning
  - Training data contains both labeled $\mathcal{L} = \{\boldsymbol{x}_i, y_i\}_{i=1}^{L}$ and unlabeled data $\mathcal{U} = \{\boldsymbol{x}_j\}_{j=L+1}^{L+U}$ (usually $U \gg L$). Note: $\mathcal{U}$ isn't the test data (if it is then the problem is known as "transductive learning").
  - Want to learn a classification/regression model combining both data sources
- Semi-Unsupervised Learning
  - We are given unlabeled data $\mathcal{D} = \{\boldsymbol{x}_i\}_{i=1}^{N}$
  - Additionally, we're given supervision in form of some constraints on data (e.g., points $\boldsymbol{x}_n$ and $\boldsymbol{x}_m$ belong to the same cluster) or "labels" of some points.

Learning with Labeled+Unlabeled Data

- Usually such problems come in one of the following two flavors
- Semi-supervised Learning
    - Training data contains both labeled $\mathcal{L} = \{\boldsymbol{x}_i, y_i\}_{i=1}^{L}$ and unlabeled data $\mathcal{U} = \{\boldsymbol{x}_j\}_{j=L+1}^{L+U}$ (usually $U \gg L$). Note: $\mathcal{U}$ isn't the test data (if it is then the problem is known as "transductive learning").
    - Want to learn a classification/regression model combining both data sources
- Semi-Unsupervised Learning
    - We are given unlabeled data $\mathcal{D} = \{\boldsymbol{x}_i\}_{i=1}^{N}$
    - Additionally, we're given supervision in form of some constraints on data (e.g., points $\boldsymbol{x}_n$ and $\boldsymbol{x}_m$ belong to the same cluster) or "labels" of some points.
    - Want to learn an unsupervised learning model combining both data sources

Learning with Labeled+Unlabeled Data

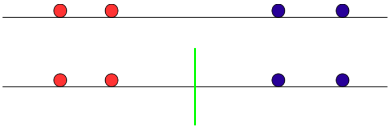
- Usually such problems come in one of the following two flavors
- Semi-supervised Learning
  - Training data contains both labeled $\mathcal{L} = \{\boldsymbol{x}_i, y_i\}_{i=1}^{L}$ and unlabeled data $\mathcal{U} = \{\boldsymbol{x}_j\}_{j=L+1}^{L+U}$ (usually $U \gg L$). Note: $\mathcal{U}$ isn't the test data (if it is then the problem is known as "transductive learning").
  - Want to learn a classification/regression model combining both data sources
- Semi-Unsupervised Learning
  - We are given unlabeled data $\mathcal{D} = \{\boldsymbol{x}_i\}_{i=1}^{N}$
  - Additionally, we're given supervision in form of some constraints on data (e.g., points $\boldsymbol{x}_n$ and $\boldsymbol{x}_m$ belong to the same cluster) or "labels" of some points.
  - Want to learn an unsupervised learning model combining both data sources
- Here, we will focus on Semi-supervised Learning (SSL)

Why/How Might Unlabeled Data Help?

- Red: $+1$, Dark Blue: -1

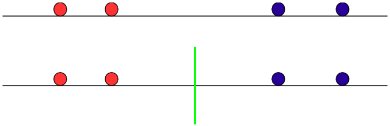
Why/How Might Unlabeled Data Help?

- Red: $+1$, Dark Blue: -1



Why/How Might Unlabeled Data Help?

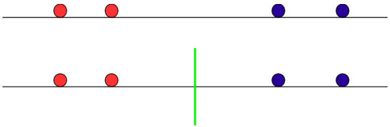
- Red: $+1$, Dark Blue: -1



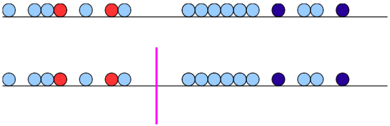
- Let's include some additional unlabeled points (Light Blue points)

Why/How Might Unlabeled Data Help?

- Red: + 1, Dark Blue: -1

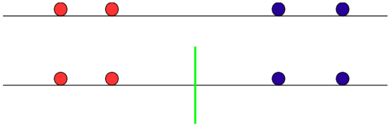


- Let's include some additional unlabeled points (Light Blue points)

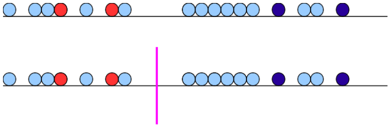


Why/How Might Unlabeled Data Help?

- Red: + 1, Dark Blue: -1



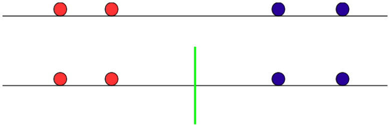
- Let's include some additional unlabeled points (Light Blue points)



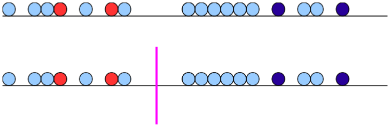
- Assumption: Examples from the same class are clustered together

Why/How Might Unlabeled Data Help?

- Red: $+1$, Dark Blue: -1

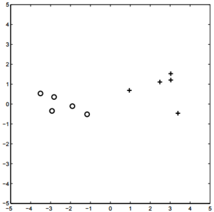


- Let's include some additional unlabeled points (Light Blue points)



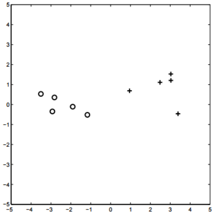
- Assumption: Examples from the same class are clustered together
- Assumption: Decision boundary lies in the region where data has low density

Why/How Might Unlabeled Data Help?

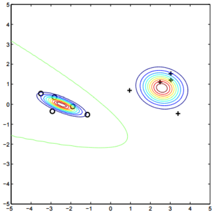


labeled data

Why/How Might Unlabeled Data Help?

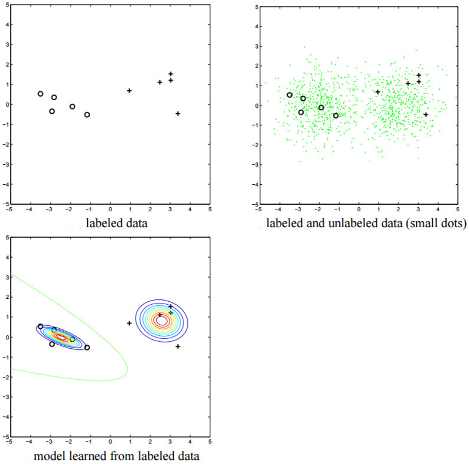


labeled data

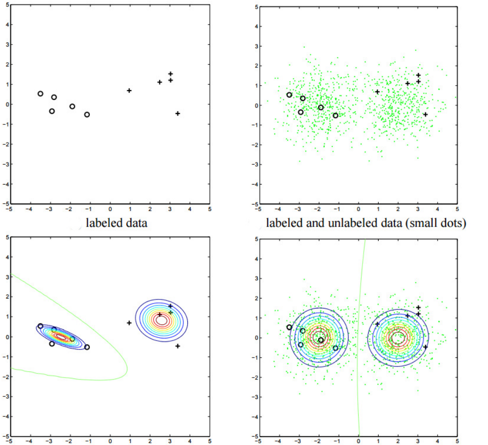


model learned from labeled data

Why/How Might Unlabeled Data Help?



Why/How Might Unlabeled Data Help?



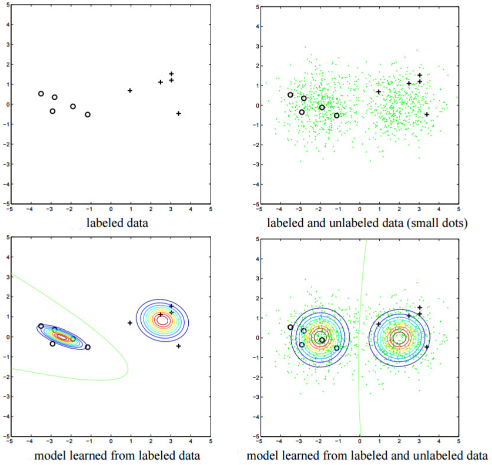
labeled data

labeled and unlabeled data (small dots)

model learned from labeled data

model learned from labeled and unlabeled data

Why/How Might Unlabeled Data Help?



In general, having some idea of the distribution of the data may be useful (even if we might not know the labels of all data points)

Some Basic Assumptions used in SSL

- Nearby points (may) have the same label
  - Smoothness assumption

Some Basic Assumptions used in SSL

- Nearby points (may) have the same label
  - Smoothness assumption
- Points in the same cluster (may) have same label
  - Cluster assumption

Some Basic Assumptions used in SSL

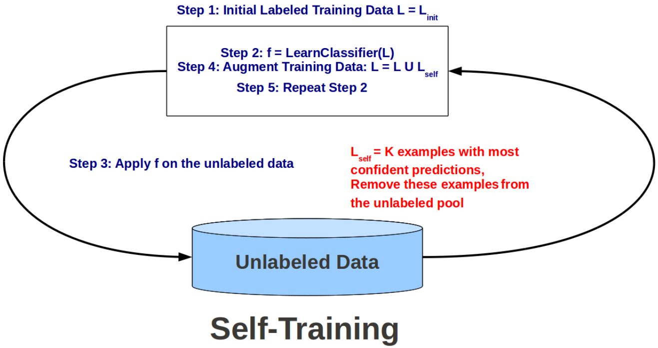
- Nearby points (may) have the same label
  - Smoothness assumption
- Points in the same cluster (may) have same label
  - Cluster assumption
- Decision boundary lies in areas where data has low density
  - Low-density separation

SSL using Self-Training

- **Given:** Small amount of initial labeled training data

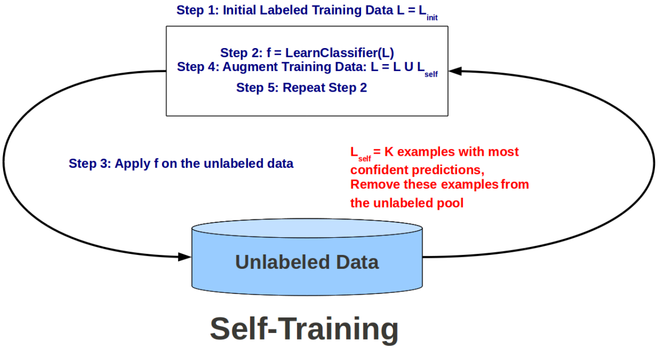
SSL using Self-Training

- **Given:** Small amount of initial labeled training data
- **Idea:** Train, predict, re-train using your own (best) predictions, repeat



SSL using Self-Training

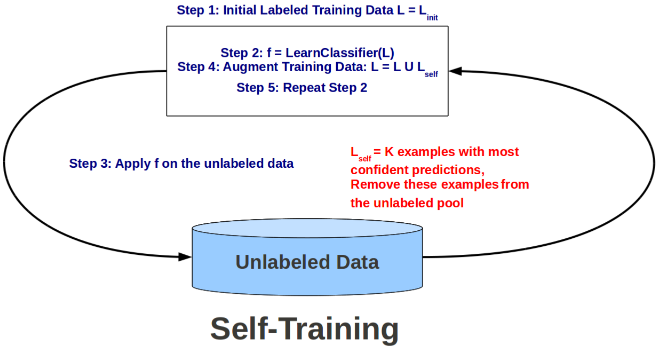
- **Given:** Small amount of initial labeled training data
- **Idea:** Train, predict, re-train using your own (best) predictions, repeat



- Can be used with any supervised learner. Often works well in practice

SSL using Self-Training

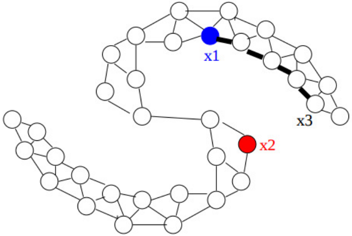
- **Given:** Small amount of initial labeled training data
- **Idea:** Train, predict, re-train using your own (best) predictions, repeat



- Can be used with any supervised learner. Often works well in practice
- **Caution:** Prediction mistakes can get reinforced

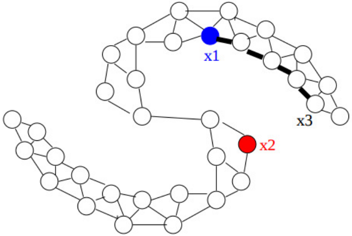
SSL using Graph-based Regularization

- Based on constructing a graph between all the examples



SSL using Graph-based Regularization

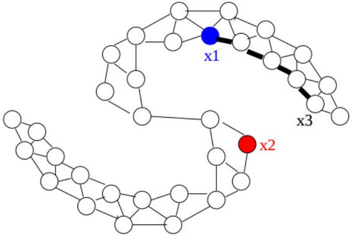
- Based on constructing a graph between all the examples



- The graph can be constructed using several ways

SSL using Graph-based Regularization

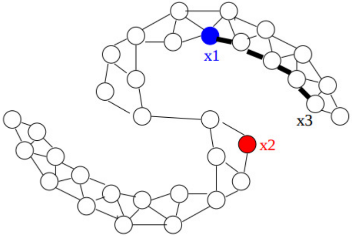
- Based on constructing a graph between all the examples



- The graph can be constructed using several ways
  - Connecting every example with its top k neighbors (labeled/unlabeled)

SSL using Graph-based Regularization

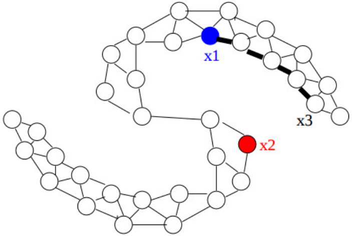
- Based on constructing a graph between all the examples



- The graph can be constructed using several ways
  - Connecting every example with its top k neighbors (labeled/unlabeled)
  - Constructing an all-connected weighted graph

SSL using Graph-based Regularization

- Based on constructing a graph between all the examples



- The graph can be constructed using several ways
  - Connecting every example with its top k neighbors (labeled/unlabeled)
  - Constructing an all-connected weighted graph
  - Weighted case: weight of edge connecting examples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$

$$a_{ij} = \exp(-||\boldsymbol{x}_i - \boldsymbol{x}_j||^2/\sigma^2)$$

.. where $\mathbf{A}$ is the $(L+U) \times (L+U)$ matrix of pairwise similarities

SSL using Graph-based Regularization

- Suppose we want to learn a function f using labeled+unlabeld data $\mathcal{L} \cup \mathcal{U}$
- Suppose f_i denotes the prediction on example $\boldsymbol{x}_i$

SSL using Graph-based Regularization

- Suppose we want to learn a function f using labeled+unlabeld data $\mathcal{L} \cup \mathcal{U}$
- Suppose f_i denotes the prediction on example $\boldsymbol{x}_i$
- Graph-based regularization assumes that the function f is smooth
  - Similar examples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ (thus high a_{ij}) should have similar f_i and f_j

SSL using Graph-based Regularization

- Suppose we want to learn a function f using labeled+unlabeld data $\mathcal{L} \cup \mathcal{U}$
- Suppose f_i denotes the prediction on example $\boldsymbol{x}_i$
- Graph-based regularization assumes that the function f is smooth
  - Similar examples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ (thus high a_{ij}) should have similar f_i and f_j
  - Graph-based regularization optimizes the following objective:

$$\min_{\boldsymbol{f}} \underbrace{\sum_{i \in \mathcal{L}} \ell(y_i, f_i)}_{\text{loss term}} + \underbrace{\lambda R(f)}_{\text{usual regularizer}} + \underbrace{\gamma \sum_{i,j \in \mathcal{L},\mathcal{U}} a_{ij}(f_i - f_j)^2}_{\text{graph-based regularizer}}$$

SSL using Graph-based Regularization

- Suppose we want to learn a function f using labeled+unlabeld data $\mathcal{L} \cup \mathcal{U}$
- Suppose f_i denotes the prediction on example $\boldsymbol{x}_i$
- Graph-based regularization assumes that the function f is smooth
  - Similar examples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ (thus high a_{ij}) should have similar f_i and f_j
  - Graph-based regularization optimizes the following objective:

$$\min_{\boldsymbol{f}} \underbrace{\sum_{i \in \mathcal{L}} \ell(y_i, f_i)}_{\text{loss term}} + \underbrace{\lambda R(f)}_{\text{usual regularizer}} + \underbrace{\gamma \sum_{i,j \in \mathcal{L},\mathcal{U}} a_{ij}(f_i - f_j)^2}_{\text{graph-based regularizer}}$$

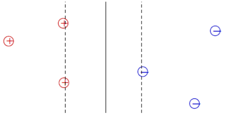
- For linear models, i.e., $\boldsymbol{f} = \mathbf{X}\boldsymbol{w}$, the objective becomes:

$$\min_{\boldsymbol{w}} \sum_{i \in \mathcal{L}} \ell(y_i, \boldsymbol{w}^\top \boldsymbol{x}_i) + \lambda \boldsymbol{w}^\top \boldsymbol{w} + \gamma \boldsymbol{w}^\top \mathbf{X}^\top \mathbf{L} \mathbf{X} \boldsymbol{w}$$

.. where $\mathbf{X}$ is the $(L + U) \times D$ feature matrix, $\mathbf{L} = \mathbf{D} - \mathbf{A}$ denotes the graph Laplacian and $\mathbf{D}$ is diagonal matrix with $D_{nn} = \sum_{m=1}^{L+U} a_{nm}$

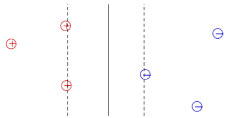
SSL using Transductive SVM

SVMs

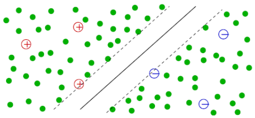


SSL using Transductive SVM

SVMs

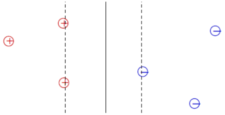


Semi-supervised SVMs (S3VMs) = Transductive SVMs (TSVMs)

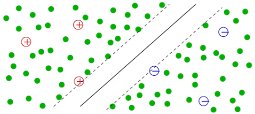


SSL using Transductive SVM

SVMs



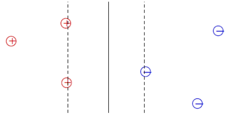
Semi-supervised SVMs (S3VMs) = Transductive SVMs (TSVMs)



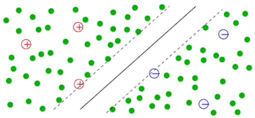
- Unlabeled data from different classes are separated by large margin

SSL using Transductive SVM

SVMs



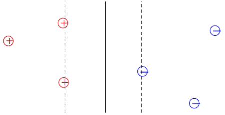
Semi-supervised SVMs (S3VMs) = Transductive SVMs (TSVMs)



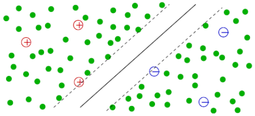
- Unlabeled data from different classes are separated by large margin
- Idea: The decision boundary shouldn't lie in the regions of high density

SSL using Transductive SVM

SVMs



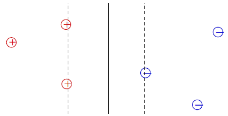
Semi-supervised SVMs (S3VMs) = Transductive SVMs (TSVMs)



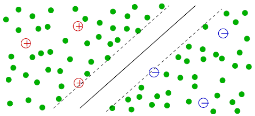
- Unlabeled data from different classes are separated by large margin
- Idea: The decision boundary shouldn't lie in the regions of high density
- Note: Transductive SVM is inductive (can use for future test data as well)

SSL using Transductive SVM

SVMs



Semi-supervised SVMs (S3VMs) = Transductive SVMs (TSVMs)



- Unlabeled data from different classes are separated by large margin
- Idea: The decision boundary shouldn't lie in the regions of high density
- Note: Transductive SVM is inductive (can use for future test data as well)

SSL using Transductive SVM

- The loss function now contains both labeled and unlabeled examples

$$\min_{\mathbf{w},b,(y_j),(\xi_k)} \quad \frac{1}{2}\langle \mathbf{w},\mathbf{w}\rangle + C\sum_i \xi_i + C^*\sum_j \xi_j$$

$$s.t. \quad \begin{array}{ll} y_i(\langle \mathbf{w},\mathbf{x}_i\rangle + b) \geq 1-\xi_i & \xi_i \geq 0 \\ y_j(\langle \mathbf{w},\mathbf{x}_j\rangle + b) \geq 1-\xi_j & \xi_j \geq 0 \end{array}$$

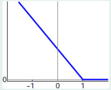
Effective Loss Functions

$$\xi_i = \max\{1 - y_i(\langle \mathbf{w},\mathbf{x}_i\rangle + b), 0\}$$

$$\xi_j = \max_{y_j\in\{+1,-1\}}\{1 - y_j(\langle \mathbf{w},\mathbf{x}_j\rangle + b), 0\}$$

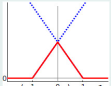
loss functions

ξ_i



$y_i(\langle \mathbf{w},\mathbf{x}_i\rangle + b)$

ξ_j



$\langle \mathbf{w},\mathbf{x}_j\rangle + b$

- Also need to optimize w.r.t. the unknown labels
- Results in a non-convex loss function but there are ways to optimize it

Generative Classification

- We've seen generative models for unsupervised learning (GMM, PPCA/FA)

Generative Classification

- We've seen generative models for unsupervised learning (GMM, PPCA/FA)
- Can also use these for Generative Classification

Generative Classification

- We've seen generative models for unsupervised learning (GMM, PPCA/FA)
- Can also use these for Generative Classification
  - Assume a class conditional data distribution $p(\boldsymbol{x}|y)$ (one for each class)

Generative Classification

- We've seen generative models for unsupervised learning (GMM, PPCA/FA)
- Can also use these for Generative Classification
  - Assume a class conditional data distribution $p(\boldsymbol{x}|y)$ (one for each class)
  - Learn $p(\boldsymbol{x}|y)$ and $p(y)$ from labeled data

Generative Classification

- We've seen generative models for unsupervised learning (GMM, PPCA/FA)
- Can also use these for Generative Classification
  - Assume a class conditional data distribution $p(\boldsymbol{x}|y)$ (one for each class)
  - Learn $p(\boldsymbol{x}|y)$ and $p(y)$ from labeled data
  - Compute $p(y|\boldsymbol{x}) \propto p(\boldsymbol{x}|y)p(y)$. Find most likely label $\hat{y}$ using $p(y|\boldsymbol{x})$

Generative Classification

- We've seen generative models for unsupervised learning (GMM, PPCA/FA)
- Can also use these for Generative Classification
  - Assume a class conditional data distribution $p(\boldsymbol{x}|y)$ (one for each class)
  - Learn $p(\boldsymbol{x}|y)$ and $p(y)$ from labeled data
  - Compute $p(y|\boldsymbol{x}) \propto p(\boldsymbol{x}|y)p(y)$. Find most likely label $\hat{y}$ using $p(y|\boldsymbol{x})$
  - This is different from discriminative classification models (like logistic reg. or SVM) which don't model $\boldsymbol{x}$ and model y directly as a function of $\boldsymbol{x}$

Generative Classification

- We've seen generative models for unsupervised learning (GMM, PPCA/FA)
- Can also use these for Generative Classification
  - Assume a class conditional data distribution $p(\boldsymbol{x}|y)$ (one for each class)
  - Learn $p(\boldsymbol{x}|y)$ and $p(y)$ from labeled data
  - Compute $p(y|\boldsymbol{x}) \propto p(\boldsymbol{x}|y)p(y)$. Find most likely label $\hat{y}$ using $p(y|\boldsymbol{x})$
  - This is different from discriminative classification models (like logistic reg. or SVM) which don't model $\boldsymbol{x}$ and model y directly as a function of $\boldsymbol{x}$
- Discriminative vs Generative Classification: Which is better?

Generative Classification

- We've seen generative models for unsupervised learning (GMM, PPCA/FA)
- Can also use these for Generative Classification
  - Assume a class conditional data distribution $p(\boldsymbol{x}|y)$ (one for each class)
  - Learn $p(\boldsymbol{x}|y)$ and $p(y)$ from labeled data
  - Compute $p(y|\boldsymbol{x}) \propto p(\boldsymbol{x}|y)p(y)$. Find most likely label $\hat{y}$ using $p(y|\boldsymbol{x})$
  - This is different from discriminative classification models (like logistic reg. or SVM) which don't model $\boldsymbol{x}$ and model y directly as a function of $\boldsymbol{x}$
- Discriminative vs Generative Classification: Which is better?
  - Depends. With lots of labeled data, discriminative models tend to work better. In small labeled data regimes, generative models often work better.

Generative Classification

- We've seen generative models for unsupervised learning (GMM, PPCA/FA)
- Can also use these for Generative Classification
  - Assume a class conditional data distribution $p(\boldsymbol{x}|y)$ (one for each class)
  - Learn $p(\boldsymbol{x}|y)$ and $p(y)$ from labeled data
  - Compute $p(y|\boldsymbol{x}) \propto p(\boldsymbol{x}|y)p(y)$. Find most likely label $\hat{y}$ using $p(y|\boldsymbol{x})$
  - This is different from discriminative classification models (like logistic reg. or SVM) which don't model $\boldsymbol{x}$ and model y directly as a function of $\boldsymbol{x}$
- Discriminative vs Generative Classification: Which is better?
  - Depends. With lots of labeled data, discriminative models tend to work better. In small labeled data regimes, generative models often work better.
  - In some cases, generative models are incredibly easy and fast to train (e.g., naïve Bayes classification in which $p(\boldsymbol{x}|y)$ has a very simple form $\prod_d^D p(x_d|y)$)

Generative Classification

- We've seen generative models for unsupervised learning (GMM, PPCA/FA)
- Can also use these for Generative Classification
  - Assume a class conditional data distribution $p(\boldsymbol{x}|y)$ (one for each class)
  - Learn $p(\boldsymbol{x}|y)$ and $p(y)$ from labeled data
  - Compute $p(y|\boldsymbol{x}) \propto p(\boldsymbol{x}|y)p(y)$. Find most likely label $\hat{y}$ using $p(y|\boldsymbol{x})$
  - This is different from discriminative classification models (like logistic reg. or SVM) which don't model $\boldsymbol{x}$ and model y directly as a function of $\boldsymbol{x}$
- Discriminative vs Generative Classification: Which is better?
  - Depends. With lots of labeled data, discriminative models tend to work better. In small labeled data regimes, generative models often work better.
  - In some cases, generative models are incredibly easy and fast to train (e.g., naïve Bayes classification in which $p(\boldsymbol{x}|y)$ has a very simple form $\prod_d^D p(x_d|y)$)
  - Generative classification models can be easily made semi-supervised

SSL using EM based Generative Classification

- Suppose data $\mathcal{D}$ is labeled $\mathcal{L} = \{\boldsymbol{x}_i, y_i\}_{i=1}^{L}$ and unlabeled $\mathcal{U} = \{\boldsymbol{x}_j\}_{j=L+1}^{L+U}$

SSL using EM based Generative Classification

- Suppose data $\mathcal{D}$ is labeled $\mathcal{L} = \{\boldsymbol{x}_i, y_i\}_{i=1}^{L}$ and unlabeled $\mathcal{U} = \{\boldsymbol{x}_j\}_{j=L+1}^{L+U}$
- Assume the following model

$$p(\mathcal{D}|\theta) = \underbrace{\prod_{i=1}^{L} p(\boldsymbol{x}_i, y_i|\theta)}_{\text{labeled}} \underbrace{\prod_{j=L+1}^{L+U} p(\boldsymbol{x}_j|\theta)}_{\text{unlabeled}} = \prod_{i=1}^{L} p(\boldsymbol{x}_i, y_i|\theta) \prod_{j=L+1}^{L+U} \sum_{y_j} p(\boldsymbol{x}_j, y_j|\theta)$$

- The unknowns are $\{y_j\}_{j=L+1}^{L+U}$ (latent variables) and θ (parameters)

SSL using EM based Generative Classification

- Suppose data $\mathcal{D}$ is labeled $\mathcal{L} = \{\boldsymbol{x}_i, y_i\}_{i=1}^{L}$ and unlabeled $\mathcal{U} = \{\boldsymbol{x}_j\}_{j=L+1}^{L+U}$
- Assume the following model

$$p(\mathcal{D}|\theta) = \underbrace{\prod_{i=1}^{L} p(\boldsymbol{x}_i, y_i|\theta)}_{\text{labeled}} \underbrace{\prod_{j=L+1}^{L+U} p(\boldsymbol{x}_j|\theta)}_{\text{unlabeled}} = \prod_{i=1}^{L} p(\boldsymbol{x}_i, y_i|\theta) \prod_{j=L+1}^{L+U} \sum_{y_j} p(\boldsymbol{x}_j, y_j|\theta)$$

- The unknowns are $\{y_j\}_{j=L+1}^{L+U}$ (latent variables) and θ (parameters)
- We can use EM to estimate the unknowns

SSL using EM based Generative Classification

- Suppose data $\mathcal{D}$ is labeled $\mathcal{L} = \{\boldsymbol{x}_i, y_i\}_{i=1}^{L}$ and unlabeled $\mathcal{U} = \{\boldsymbol{x}_j\}_{j=L+1}^{L+U}$
- Assume the following model

$$p(\mathcal{D}|\theta) = \underbrace{\prod_{i=1}^{L} p(\boldsymbol{x}_i, y_i|\theta)}_{\text{labeled}} \underbrace{\prod_{j=L+1}^{L+U} p(\boldsymbol{x}_j|\theta)}_{\text{unlabeled}} = \prod_{i=1}^{L} p(\boldsymbol{x}_i, y_i|\theta) \prod_{j=L+1}^{L+U} \sum_{y_j} p(\boldsymbol{x}_j, y_j|\theta)$$

- The unknowns are $\{y_j\}_{j=L+1}^{L+U}$ (latent variables) and θ (parameters)
- We can use EM to estimate the unknowns
  - Given $\theta = \hat{\theta}$, E step will compute expected labels for unlabeled examples

$$\mathbb{E}[y_j] = +1 \times P(y_j = +1|\hat{\theta}, \boldsymbol{x}_j) + (-1) \times P(y_j = -1|\hat{\theta}, \boldsymbol{x}_j)$$

SSL using EM based Generative Classification

- Suppose data $\mathcal{D}$ is labeled $\mathcal{L} = \{\boldsymbol{x}_i, y_i\}_{i=1}^{L}$ and unlabeled $\mathcal{U} = \{\boldsymbol{x}_j\}_{j=L+1}^{L+U}$
- Assume the following model

$$p(\mathcal{D}|\theta) = \underbrace{\prod_{i=1}^{L} p(\boldsymbol{x}_i, y_i|\theta)}_{\text{labeled}} \underbrace{\prod_{j=L+1}^{L+U} p(\boldsymbol{x}_j|\theta)}_{\text{unlabeled}} = \prod_{i=1}^{L} p(\boldsymbol{x}_i, y_i|\theta) \prod_{j=L+1}^{L+U} \sum_{y_j} p(\boldsymbol{x}_j, y_j|\theta)$$

- The unknowns are $\{y_j\}_{j=L+1}^{L+U}$ (latent variables) and θ (parameters)
- We can use EM to estimate the unknowns
  - Given $\theta = \hat{\theta}$, E step will compute expected labels for unlabeled examples

$$\mathbb{E}[y_j] = +1 \times P(y_j = +1|\hat{\theta}, \boldsymbol{x}_j) + (-1) \times P(y_j = -1|\hat{\theta}, \boldsymbol{x}_j)$$

  - M step can then perform standard MLE for re-estimating the parameters θ

SSL using EM based Generative Classification

- Suppose data $\mathcal{D}$ is labeled $\mathcal{L} = \{\boldsymbol{x}_i, y_i\}_{i=1}^{L}$ and unlabeled $\mathcal{U} = \{\boldsymbol{x}_j\}_{j=L+1}^{L+U}$
- Assume the following model

$$p(\mathcal{D}|\theta) = \underbrace{\prod_{i=1}^{L} p(\boldsymbol{x}_i, y_i|\theta)}_{\text{labeled}} \underbrace{\prod_{j=L+1}^{L+U} p(\boldsymbol{x}_j|\theta)}_{\text{unlabeled}} = \prod_{i=1}^{L} p(\boldsymbol{x}_i, y_i|\theta) \prod_{j=L+1}^{L+U} \sum_{y_j} p(\boldsymbol{x}_j, y_j|\theta)$$

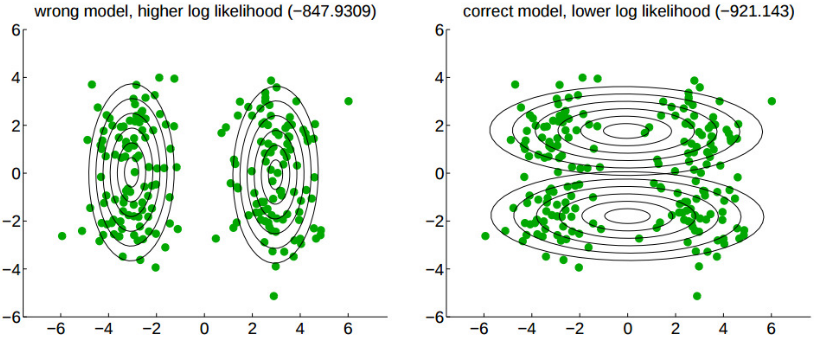
- The unknowns are $\{y_j\}_{j=L+1}^{L+U}$ (latent variables) and θ (parameters)
- We can use EM to estimate the unknowns
  - Given $\theta = \hat{\theta}$, E step will compute expected labels for unlabeled examples

$$\mathbb{E}[y_j] = +1 \times P(y_j = +1|\hat{\theta}, \boldsymbol{x}_j) + (-1) \times P(y_j = -1|\hat{\theta}, \boldsymbol{x}_j)$$

  - M step can then perform standard MLE for re-estimating the parameters θ
- A fairly general framework for semi-supervised learning. Can be used for different types of data (by choosing the appropriate $p(\boldsymbol{x}|y)$ distribution)

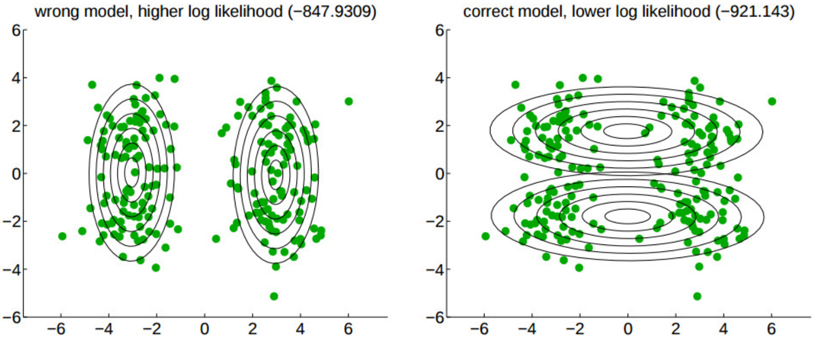
Things can go wrong..

If assumptions are not appropriate for the data (e.g., incorrectly specified class conditional distributions)



Things can go wrong..

If assumptions are not appropriate for the data (e.g., incorrectly specified class conditional distributions)



Thus need to be careful/flexible about the choice of class conditional distributions

Summary

- Looked at SSL methods based on
  - Self-training
  - Changing the regularizer (e.g., graph-regularization)
  - Changing the loss function (e.g., transductive SVM)
  - Using generative classification models
- Caution: SSL may not always help, especially if the assumptions about the data distribution are wrongly specified

Summary

- Looked at SSL methods based on
  - Self-training
  - Changing the regularizer (e.g., graph-regularization)
  - Changing the loss function (e.g., transductive SVM)
  - Using generative classification models
- Caution: SSL may not always help, especially if the assumptions about the data distribution are wrongly specified
- Very important idea in general. Lots of prior work. Lots of recent/renewed interest (especially in improving Deep Learning models that usually require hige amounts of labeled data to train).

Summary

- Looked at SSL methods based on
  - Self-training
  - Changing the regularizer (e.g., graph-regularization)
  - Changing the loss function (e.g., transductive SVM)
  - Using generative classification models
- Caution: SSL may not always help, especially if the assumptions about the data distribution are wrongly specified
- Very important idea in general. Lots of prior work. Lots of recent/renewed interest (especially in improving Deep Learning models that usually require hige amounts of labeled data to train).
- Also has similar goals as areas like Active Learning (selectively deciding which training examples to acquire labels for) and Crowdsourced Labeling