Transductive SVM and Domain Adaptation

CAP6676: Knowledge Representation
Guo-Jun Qi
Semi-Supervised Learning and Manifold Assumption

• Manifold assumption
  • All the examples can be re-parameterized on a low-dimensional manifold
  • The similarity between examples is not calculated based on their distances in ambient space.
  • The similarity should be calculated based on their distances over the manifold

• Semi-supervised learning
  • Label propagation – labels are propagated over the graph representation of manifold
  • Semi-supervised learning – graph cut + consistency with labeled examples
Drawback of Manifold Assumption

• In many cases, examples are not supposed to reside on a low-dimensional manifold
• Rather, they reside in an ambient space, which cannot be embedded in a lower-dimensional manifold
• Can we still be able to explore unlabeled data? Yes.
Transductive SVM

• Maximum-margin decision boundary given a small number of labeled examples
Transductive SVM

• Maximum-margin decision boundary given a small number of labeled examples

• How will the decision boundary change given both labeled and unlabeled examples?
Transductive SVM

• Maximum-margin decision boundary given a small number of labeled examples

• How will the decision boundary change given both labeled and unlabeled examples?

• Move the decision boundary so that the new boundary passes through a low density area, yielding larger margin.
Transductive SVM

• Maximum-margin decision boundary given a small number of labeled examples

• How will the decision boundary change given both labeled and unlabeled examples?

• Move the decision boundary so that the new boundary passes through a low density area, yielding larger margin.

• Having the classification results on the unlabeled data.
Transductive SVM: Formulation

• Labeled data

\[ L = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \]

• Unlabeled data

\[ D = \{(x_{n+1}), (x_{n+2}), \ldots, (x_{n+m})\} \]

• Maximum margin principle for mixture of labeled and unlabeled data
  • Objective: Seeking for the maximum margin classifier over all possible assignment of labels to unlabeled data
  • Given hypothetical labels to unlabeled data, computing the corresponding margin
  • Finding the label assignment whose margin is maximized
Transductive SVM

- Different label assignment for unlabeled data, resulting in different margins
Review: Traditional SVM

\[ \vec{w} \cdot \vec{x} + b = 0 \]
\[ \vec{w} \cdot \vec{x} + b = -1 \]
\[ \vec{w} \cdot \vec{x} + b = +1 \]

\[ y = \begin{cases} 
1 & \text{if } \vec{w} \cdot \vec{x} + b \geq 1 \\
-1 & \text{if } \vec{w} \cdot \vec{x} + b \leq -1 
\end{cases} \]

Margin = \( \frac{2}{\| \vec{w} \|^2} \)
SVM Formulation

• Maximizing

\[ \text{Margin} = \frac{2}{\| \mathbf{w} \|^2} \]

• Equivalent to minimizing

\[ \| \mathbf{w} \|^2 = \mathbf{w} \cdot \mathbf{w} \]

• Constraints:

\[ \mathbf{w} \cdot \mathbf{x}_i + b \geq 1 \quad \text{if} \quad y_i = 1 \]
\[ \mathbf{w} \cdot \mathbf{x}_i + b \leq -1 \quad \text{if} \quad y_i = -1 \]
\[ y_i (\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \]
Original SVM

\[ \{\mathbf{w}^*, b^*\} = \underset{\mathbf{w}, b}{\text{argmin}} \mathbf{w} \cdot \mathbf{w} \]

\[ y_1(\mathbf{w} \cdot \mathbf{x}_1 + b) \geq 1 \]

\[ y_2(\mathbf{w} \cdot \mathbf{x}_2 + b) \geq 1 \]

\[ \vdots \]

\[ y_n(\mathbf{w} \cdot \mathbf{x}_n + b) \geq 1 \]

Transductive SVM

\[ \{\mathbf{w}^*, b^*\} = \underset{\mathbf{w}, b}{\text{argmin}} \mathbf{w} \cdot \mathbf{w} \]

\[ y_{n+1}(\mathbf{w} \cdot \mathbf{x}_{n+1} + b) \geq 1 \]

\[ y_{n+2}(\mathbf{w} \cdot \mathbf{x}_{n+2} + b) \geq 1 \]

\[ \vdots \]

\[ y_{n+m}(\mathbf{w} \cdot \mathbf{x}_{n+m} + b) \geq 1 \]

Constraints for unlabeled data

labeled examples

unlabeled examples

labeled examples
Alternative Optimization

Transductive SVM

\[
\{ \tilde{w}^*, b^* \} = \arg \min_{\tilde{w}, b} \arg \min_{y_{n+1}, \ldots, y_{n+m}} \tilde{w} \cdot \tilde{w}
\]

\[
y_1 (\tilde{w} \cdot \tilde{x}_1 + b) \geq 1
\]

\[
y_2 (\tilde{w} \cdot \tilde{x}_2 + b) \geq 1 \quad \text{labeled examples}
\]

\[
\ldots
\]

\[
y_n (\tilde{w} \cdot \tilde{x}_n + b) \geq 1
\]

\[
y_{n+1} (\tilde{w} \cdot \tilde{x}_{n+1} + b) \geq 1 \quad \text{unlabeled examples}
\]

\[
\ldots
\]

\[
y_{n+m} (\tilde{w} \cdot \tilde{x}_{n+m} + b) \geq 1
\]

- Step 1: fix \( y_{n+1}, \ldots, y_{n+m} \), learn weights \( \mathbf{w} \)

- Step 2: fix weights \( \mathbf{w} \), try to predict \( y_{n+1}, \ldots, y_{n+m} \)
Self-Training

- Self-training
  - Initialize the label assignment to unlabeled data
  - Train SVM with labeled + unlabeled examples
  - Update the label assignment to unlabeled data
  - Retrain SVM ...

- Apply to the other classifier as well
  - Logistic regression
  - Neural Networks
Domain Adaption

• What if unlabeled examples have different distribution from the labeled examples?

[Diagram showing a flow from training (labeled) to classifier to test (unlabeled) with 85.5% accuracy marked on the test set.

New York Times

New York Times]
Domain Adaption

• What if unlabeled examples have different distribution from the labeled examples?
Domain Discrepancy

- **Ideal Setting**: Train on NYT, classifier, test on NYT, accuracy 85.5%
- **Realistic Setting**: Train on Reuters, classifier, test on NYT, accuracy 64.1%
Domain Adaptation

- Reuters
- Newsgroup

Training (labeled)

Classifier

Test (completely unlabeled)

New York Times
Goal

• Finding and combining the knowledge consistent with target domain

• Re-weighting source examples
  • The source examples overlapping with the target domain should have large weights
  • Otherwise, they are of no importance or even noisy
Instance-based domain adaptation

• **Input**: labeled examples in source domain and unlabeled examples in target domain

• **Output**: classification model for target domain

• **Assumption**: source domain and target domain have different distributions $P(X_s)$ and $P(X_T)$, thus the examples from the source domain may have different importance in training the classifier for a target domain
  
  • The source examples in the high density area of target domain shall be more important
  
  • The source examples in the low density area of target domain can be considered as outliers, thus being less important or even noisy in training the classifier for the target domain.
Remedy Distribution Discrepancy

\[ \theta^* = \arg \min_{\theta \in \Theta} \sum_{i=1}^{n_S} \frac{P(x_{T_i})}{P(x_{S_i})} f(x_{S_i}, y_{S_i}, \theta) \]

- The ratio of target domain density to the source domain density measures the importance in training the classifier.
- Using this ratio to weigh the loss over the training examples:
  - Inlier: an example is important if it has relatively high density in target domain.
  - Outlier: an example is less important if it has relatively low density in target domain.
Estimating density ratio

**How to estimate** \( \frac{P(x_{T_i})}{P(x_{S_i})} \)?

One straightforward solution is to estimate \( P(X_s) \) and \( P(X_T) \), respectively. However, estimating density function is a hard problem.

Alternatively, we estimate \( \beta_i = \frac{P(x_{S_i})}{P(x_{T_i})} \) directly.
Minimizing Mean discrepancy

• Minimizing the mean of distributions between two domains
  • Distribution mean of target domain, approximated by sample mean
    \[ \mu_T = E_{x \sim P_T(x)}(x) \approx \frac{1}{n} \sum_{i=1}^{n} x_{T_i} \]
  • Alternatively, we can compute distribution mean of target domain from the source domain examples
    \[ \mu_T = E_{x \sim P_T(x)}(x) = \int_x xP_T(x)dx = \int_x x\beta(x)P_S(x)dx \]
    \[ = E_{x \sim P_S(x)}(\beta(x)x) \approx \frac{1}{m} \sum_{j=1}^{m} \beta(x_{S_j})x_{S_j} \]
  • Minimizing these two results
    \[ \left\| \frac{1}{n} \sum_{i=1}^{n} x_{T_i} - \frac{1}{m} \sum_{j=1}^{m} \beta(x_{S_j})x_{S_j} \right\|^2 \]
Minimizing Mean discrepancy

• Step 1: Finding the weight $\beta_i = \frac{p(x_{S_i})}{p(x_{T_i})}$

$$\min_{\beta} \left\| \frac{1}{n} \sum_{i=1}^{n} x_{T_i} - \frac{1}{m} \sum_{j=1}^{m} \beta(x_{S_j}) x_{S_j} \right\|^2$$

s.t., $0 \leq \beta(x_{S_j}) \leq B$

$$1 - \frac{1}{m} \sum_{j=1}^{m} \beta(x_{S_j}) \leq \varepsilon$$

• Step 2: Minimizing the weighted loss to solve the target classifier

$$\theta^* = \arg \min_{\theta \in \Theta} \sum_{i=1}^{n_S} \frac{p(x_{T_i})}{p(x_{S_i})} l(x_{S_i}, y_{S_i}, \theta)$$

• Constraints
  • Limiting the scope of ratio
  • Ensure $\beta(x_{S_j}) p(x_{S_j}) \approx \frac{1}{m} \beta(x_{S_j})$ is a validate probability
Summary

• Exploring unlabeled examples
  • Semi-Supervised Learning
    • Manifold Assumption: data are distributed over a low-dimensional manifold embedded in the ambient space

• Transductive SVM
  • Maximum margin with unlabeled examples
  • Self-training in the loop

• Domain Adaptation: when training and test distributions differ
  • Minimizing the mean between the training and test distributions