Primer on Generative vs. Discriminative Learning

Thorstun Joachims
Cornell University
Department of Computer Science

Project

- Project Topic
  - Self-defined: comparison of methods, application of exiting
  method to new domain, explore modifications of existing
  method, etc.
  - Can be centered around a paper, but should envision some
  originality
  - I will help refine the topic
- Project Proposal
  - Outline general idea for project
  - Discuss resources and timeline
  - Due: September 19 (in class)

Generative vs. Discriminative Learning

Generative vs. Discriminative Models

- Learning Task: \( P(X,Y) = P(X) P(Y|X) \)
  - Input Space: \( X \) (e.g. feature vectors, word sequence, etc.)
  - Output Space: \( Y \) (e.g. class label)
  - Training Data: \( S_{train} = \{ (x_1, y_1), \ldots, (x_n, y_n) \} \sim_{iid} P(X,Y) \)
  - Goal: Find \( h: X \rightarrow Y \) with low prediction error \( Err_P(h) \)

Generalization Error and Sample Error

Definition: The prediction error/generalization error/true error/expected loss/risk \( Err_P(h) \) of a hypothesis \( h \) for a learning task \( P(X,Y) \) is

\[
Err_P(h) = \sum_{x \in X, y \in Y} \Delta(h(x), y) P(X = x, Y = y).
\]

Definition: \( \Delta(u, b) \) is a loss function that measures the cost of making a wrong prediction. A commonly used loss function is the 0/1-loss

\[
\Delta(u, b) = \begin{cases} 
0 & \text{if } u = b \\
1 & \text{else}
\end{cases}
\]

Definition: The error on sample \( S \) \( Err_S(h) \) of a hypothesis \( h \) is \( Err_S(h) = \frac{1}{n} \sum_{i=1}^{n} \Delta(h(x_i), y_i) \).

Generative Model: Model \( P(X,Y) \)

- Bayes' Decision Rule: Optimal Decision is

\[
\hat{h}(x) = \arg\max_{h \in H} \sum_{y \in Y} P(Y=y|h(X) = x) \\
= \arg\max_{h \in H} P(Y=y|X=x) \\
= \arg\max_{h \in H} P(X=x|Y=y) P(Y=y) \\
= \arg\max_{h \in H} P(Y=y|X=x) P(X=x) \\
= \arg\max_{h \in H} P(Y=y|X=x)
\]

- Equivalent Reformulations: For \((0,1)-Loss \Delta(y'|y) = 1, \text{if } y' \neq y, 0 \text{ else} \)

\[
\Delta(y') = \begin{cases} 
0 & \text{if } y' = y \\
1 & \text{else}
\end{cases}
\]

- Learning: maximum likelihood (or MAP, or Bayesian)
  - Assume model class \( P(X,Y|\omega) \) with parameters \( \omega \in \Omega \)
  - Find \( \omega \in \Omega \) that best matches \( P(X,Y|\omega) \) on training data (e.g. MLE)

- Examples: naive Bayes, HMM

Discriminative Model

- Model \( P(X|Y) \) with distributions \( P(Y|X) \)
  - Find \( \omega \) that best matches \( P(X|Y) \) on training data (e.g. MLE)
  - Examples: Log. Reg., CRF

- Training Examples \( \{(x_1, y_1), \ldots, (x_n, y_n)\} \sim_{iid} P(X,Y) \)
### Learning Task:
- Model discriminant functions
- Assume:
  - Output Space \( Y: \{1,-1\} \)
  - Class conditional model (one for each class)
  - Prior class probabilities
  - Class conditional model (one for each class)

### Generative vs. Discriminative Models
- **Discriminative Model**
  - Model \( P(Y|X) \) with \( P(Y|X,a) \)
  - Find \( a \) e.g. via MLE
  - Examples: Log. Reg., CRF
  - Model discriminant functions
  - Find \( b_a \in H \) with low train loss (e.g. Emp. Risk Min.)
  - Examples: SVM, Dec. Tree

### Naïve Bayes’ Classifier (Multivariate)
- Input Space \( X \): Feature Vector
- Output Space \( Y: \{1,-1\} \)
- Model:
  - Prior class probabilities
  - Class conditional model (one for each class)

### Estimating the Parameters of Naïve Bayes
- Count frequencies in training data
  - \( n: \) number of training examples
  - \( n_{i,y}: \) number of pos/neg examples
  - \( n(x^0=x^0_i,y): \) number of times feature \( x^0 \) takes value \( x^0_i \) for examples in class \( y \)
  - \( |x^0|: \) number of values of attribute of \( x^0 \)

### Discriminative Model: Model \( P(Y|X) \)
- **Bayes’ Decision Rule:**
  - General:
    \[
    \hat{y} = \text{argmax}_y \sum_{x \in X} n(x,y) P(Y=y|X=x) 
    \]
  - Assume 0/1 Loss \( \Delta(y,y') = 1, \) if \( y \neq y', \) 0 else
    \[
    \hat{y} = \text{argmax}_y \sum_{x \in X} n(x,y) P(Y=y|X=x) 
    \]
- **Learning:** maximum likelihood (or MAP, or Bayesian)
  - Assume model class \( P(Y|X,a) \) with parameters \( a \in \Omega \)
  - Find
    \[
    \hat{a} = \text{argmax}_a \sum_{x \in X} \log P(Y=y|X=x,a) 
    \]

### Logistic Regression/“Maximum Entropy”
- Assume:
  \[
  P(Y=y|x_1,...,x_m) = \frac{e^{f(x_1,...,x_m)}}{\sum_{y} e^{f(x_1,...,x_m)}} 
  \]
- Learn one weight vector \( w \) for each class \( y \in Y \) (linear discriminant)
- **Maximum Likelihood training:**
  \[
  \hat{w} = \text{argmax}_w \sum_{x \in X} \log P(Y=y|x,\hat{w}) 
  \]
  \[
  = \text{argmax}_w \sum_{x \in X} \log \left( \sum_{y} e^{f(x,y)} \right) 
  \]

### Generative vs. Discriminative Models
- **Generative Model**
  - Model \( P(X,Y) \) with distributions \( P(X,Y|a) \)
  - Find \( a \) that best matches \( P(X,Y) \) on training data (e.g. MLE)
  - Examples: na"ıve Bayes, HMM
Discriminative Model: Model Discriminant Function \( h \) Directly

- Discriminant Function: \( h: X \times Y \rightarrow \mathbb{R} \)

\[
h(x) = \frac{\sum_{y \in Y} p(x, y) \cdot h(x, y)}{\sum_{y \in Y} p(x, y)}\]

- Consistency of Empirical Risk:
  - Training Error (i.e. Empirical Risk): \( \sum_{(x, y) \in \text{training}} \mathbb{I}(y \neq h(x)) \)
  - For sufficiently “small” \( H_0 \) and “large” \( S \): Rule \( h \in H_0 \) with best \( \mathbb{E}_{X,Y} \mathbb{I}(y \neq h(x)) \) close to \( \min_{h \in H_0} \mathbb{E}_{X,Y} \mathbb{I}(y \neq h(x)) \)

- Learning: Empirical Risk Minimization (ERM)
  - Assume class \( H_0 \) of discriminant functions \( h: X \rightarrow Y \)
  - Find \( \hat{h} = \arg\max_{h \in H_0} \mathbb{E}_{X,Y} \mathbb{I}(y \neq h(x)) \)

Support Vector Machine

- Training Examples: \( \{(x_i, y_i) \}_{i=1}^n \in \mathbb{R} \times \{1, -1\} \)
- Hypothesis Space: \( H_0 = \{h: \mathbb{R} \rightarrow \mathbb{R}^+ \mid h \geq 0 \} \)
- Training Loss: \( \mathbb{E}_{X,Y} \mathbb{I}(y \neq h(x)) \leq \delta \)

Optimization Problem:

\[
\frac{1}{2} \sum_{i,j} w_{ij} x_i^T x_j + \sum_i \alpha_i \leq 1 - \xi_i \quad \text{for all } i
\]

So what about Complex Outputs?

- Approach: view as multi-class classification task
  - Every complex output \( y \in Y \) is one class
  - The bear chased the cat

- Problem: Exponentially many classes!
  - Generative Model: \( P(X,Y) = P(Y)P(X|Y) \)
  - Discriminative Model: \( P(Y|X) \)
  - Discriminant Functions: \( h: X \times Y \rightarrow \mathbb{R} \)

- Challenges
  - How to compactly represent model?
  - How to do efficient inference with model (i.e. \( \arg\max_{h \in H} \mathbb{E}_{X,Y} \mathbb{I}(y \neq h(x)) \))?
  - How to effectively estimate model from data?
  - (e.g. compute \( \mathbb{E}_{X,Y} \mathbb{I}(y \neq h(x)) \))

Predicting Sequences: Hidden Markov Model

- Bayes rule: \( \hat{a}(x) = \arg\max_{a} P(y = \hat{y} | x = \hat{x}) \cdot P(y = \hat{y} | a) \)

- Assumptions for compact representation

\[
P(y = \hat{y} | x = \hat{x}) = \prod_i P(y_i = \hat{y}_i | x_i = \hat{x}_i)
\]

- Prediction rule:

\[
\hat{a}(x) = \arg\max_{a} \prod_i P(y_i = \hat{y}_i | x_i = \hat{x}_i) \cdot P(x = \hat{x})
\]

\( \rightarrow \) Viterbi (dynamic prog) algorithm computes \( \arg\max \) efficiently

Predicting Sequences: HMM Training

- Maximum Likelihood (or alternative estimator)

\[
\hat{a}(x) = \arg\max_{a} \prod_i P(y_i = \hat{y}_i | x_i = \hat{x}_i) \cdot P(x = \hat{x})
\]

- For Hidden Markov Model

  - Closed-form solutions

\[
P(Y = y | x = \hat{x}) = \frac{1}{2 \text{#of Times StateA Follows StateB}} \quad \text{#of Times StateB occurs}
\]

\[
P(Y = y | x = \hat{x}) = \frac{1}{2 \text{#of Times OutputM Observed StateB}} \quad \text{#of Times StateB occurs}
\]

  - Need for smoothing the estimates (e.g. max a posteriori)
Questions in this Class

- Important applications for which conventional methods don’t fit!
  - Noun-phrase co-reference: two step approaches of pair-wise classification and clustering as postprocessing, e.g. [Ng & Cardie, 2002]
  - Directly optimize complex loss functions (e.g. F1, AvgPrec)
- Improve upon existing methods!
  - Part-of-Speech Tagging: generative models like HMM
  - Is it possible to train discriminatively? Log Regression, SVMs, Boosting, etc.
  - SVM outperforms Naive Bayes for text classification [Joachims, 1998] [Dumais et al., 1998]

<table>
<thead>
<tr>
<th>Model</th>
<th>True Positive Rate</th>
<th>False Positive Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reuters</td>
<td>72.1</td>
<td>87.5</td>
</tr>
<tr>
<td>WebKB</td>
<td>82.0</td>
<td>90.3</td>
</tr>
<tr>
<td>Ohsumed</td>
<td>62.4</td>
<td>71.6</td>
</tr>
</tbody>
</table>

Reading

- Generative vs. Discriminative Training
- Hidden-Markov Models
- Logistic Regression
  - T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning, Springer, 2001. (Section 4.4)
- Support Vector Machines