Learning from a Probabilistic Perspective

Data Mining and Concept Learning
CSI 5387

- Bayesian network classifiers
- Decision trees
- Random Forest
- Neural networks
Bayes Classifier

\[ P(C|X_1, X_2, \ldots, X_i) = \frac{P(C) \prod_{i=1}^{n} P(X_i|Parent(X_i), C)}{P(X_1, X_2, \ldots, X_i)} \]

- Posterior probability \( P(C|X_1, X_2, \ldots, X_i) \)
- Prior probability: \( P(C) \)
- Conditional probability: \( P(X_i|Parent(X_i), C) \)
- Normalization factor: \( P(X_1, X_2, \ldots, X_i) \)

\[ P(X_1, X_2, \ldots, X_i) = \sum_{j=1}^{C} P(c_j) \prod_{i=1}^{n} P(X_i|Parent(X_i), c_j) \]

\[ P(C|X_1, X_2, \ldots, X_i) = aP(C) \prod_{i=1}^{n} P(X_i|Parent(X_i), C) \]

---

Probabilistic Independence

\[ P(X_1, X_2, X_3) = P(X_1)P(X_2)P(X_3) \]

- For a Boolean concept
- \( P(X_1, X_2, X_3) \) requires 8 parameters(\(2^n\))
- \( P(X_1)P(X_2)P(X_3) \) requires 6 parameters(\(2*n\))
- A way to reduce complexity
  (OO design, Voting)
- What is independence? (committee)
Bayes Classifier

<table>
<thead>
<tr>
<th>X1</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>0</td>
<td>+</td>
</tr>
<tr>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>0</td>
<td>-</td>
</tr>
</tbody>
</table>

IF X1=0, Then C=-

\[
P(+)P(X1=0|+) = 0.2 \times \frac{1}{1} = 0.2
\]

\[
P(-)P(X1=0|-) = 0.8 \times \frac{2}{4} = 0.4
\]

\[
\alpha = \frac{1}{0.2 + 0.4}
\]

\[
P(+)\mid X1 = 0 = \frac{0.2}{0.6} = 0.33
\]

Naïve Bayes Classifier

Simplest Bayesian networks

<table>
<thead>
<tr>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>+</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
</tbody>
</table>

P(X1|C) P(X2|C) P(X3|C)
Naïve Bayes Classifier

When the independence assumption is violated
- Inaccurate probability estimation
- For classification, large tolerance for dependencies

\[
P(+)P(X1 = 0|+) = 0.2 \times \left(\frac{1}{4}\right)^3 = 0.2 \\
P(-)P(X1 = 0|-) = 0.8 \times \left(\frac{2}{4}\right)^3 = 0.1 \\
\alpha = \frac{1}{0.2 + 0.1} \\
P(+|X1 = 0, X2 = 0, X3 = 0) = \frac{0.2}{0.3} = 0.66
\]

Bayesian network Classifiers

IF X1=0, Then C=-

Advantage: independence makes model simpler
Disadvantage: if variables contain dependencies, searching structure is difficult
Naïve Bayes is the most popular Bayesian network classifier
Probabilistic Decision Trees

<table>
<thead>
<tr>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>+</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
</tbody>
</table>

Decision trees only use X1 to split

IF X1=0, Then C=-

Probabilistic Decision Trees

- Each leaf represents a probabilistic distribution $P(C|X_1,X_2\ldots,X_3)$
- The leaf class distributions are only related to
  1. The number of training instances
  2. The number of instances in the leaf

\[
P(+) = \frac{1}{5} = 0.2
\]

\[
P(+) = \frac{3}{5} = 0.6
\]

\[
\alpha = \frac{1}{0.2 + 0.4}
\]

\[
P(+|X_1 = 0) = \frac{\alpha N_+}{N} \times \frac{N_{+0}}{N_+} = \frac{\alpha N_{+0}}{N}
\]
Bias and Variance

- Each leaf represents a probabilistic distribution \( P(C| X_1, X_2 \ldots X_3) \)
- A small leaf has inaccurate probability estimation. High variance
- A large leaf represents a less number of variables. High bias
Example: use \( P(C) \) to approximate \( P(C| X_1, X_2 \ldots X_3) \)
- Ideally, a large leaf and many path variables

Decision Trees vs. Bayesian networks

<table>
<thead>
<tr>
<th></th>
<th>Bayesian networks</th>
<th>Decision trees</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample efficiency</td>
<td>better</td>
<td>worse</td>
</tr>
<tr>
<td>Structure learning</td>
<td>Hard (general graph)</td>
<td>Easy. (divide-conquer)</td>
</tr>
</tbody>
</table>

- Similarity: probabilistic model.
  - learn optimal classifiers given sufficient data
- Practice: Bayes classifiers perform well in small datasets, when decision trees are optimal given sufficient data
- Context specific (in)dependence
## Duplication Problem in Decision trees

A Boolean concept: $C = (A_1 \text{ and } A_2) \text{ or } (A_3 \text{ and } A_4)$

(A3 and A4) has been learned twice, and thus requires more instances to learn:

- $\{A_1=T, A_2=F, A_3=T, A_4=T\}$
- $\{A_1=T, A_2=F, A_3=T, A_4=F\}$
- $\{A_1=T, A_2=F, A_3=F, A_4=F\}$

## Independent Decision trees

A Boolean concept: $C = (A_1 \text{ or } A_2) \text{ and } (A_3 \text{ or } A_4)$

$P(C=1)=7/16$

$P(C=0)=9/16$

$P(A_1=1, A_2=1 | C=1) = 1$

$P(A_1=1, A_2=1 | C=0) = 0$

$P(A_1=1, A_2=1 | C=1) = 1$

$P(A_1=1, A_2=1 | C=0) = 0$

$P(A_1, A_2, A_3, A_4 | C) \approx P(A_1, A_2 | C)P(A_3, A_4 | C)$

Note that we have a large leaf for each tree, and still utilize the same number of variables to make predictions.
Independent Decision trees

• A set of independent trees are more compact than a decision tree, and thus require less training data to learn

• Finding independence between variables is difficult

• An approximation learning algorithm is desired in practice

Learning Independent Decision Trees

• Construct a set of decision trees by injecting randomness to tree learning

• Randomness makes each tree tend to be independent with each other. Lower variance

• Each decision tree represents the dependency between variables. Lower bias
Random Trees

- Bagging: grow a tree on randomly select samples from training data with replacement
  1. each tree has high prediction power but dependencies between trees are strong
  2. when the sample size is large, all trees converge to one

- Random trees: Randomly select the splitting attribute in each node.
  (1) each tree is more independent, but the prediction power is low.
  (2) data contains many useless variables

Random Forests

Building one tree in random forests

1. For each node, build a tree
   - Randomly select $k$ variable $S$
   - Pick the best variable from $S$
   - Split the training data into subsets based on the values of the best variable
3. For each derived subset, repeat the preceding steps.
Parameters in Random Forest

- The larger $k$, the more dependencies between trees
- The larger $k$ may not always improve the accuracy of individual trees in small dataset, but may have difference in larger datasets. Why? Example: a set of equal important variables, and the training data only allows one split.
- The performance of random forest is not very sensitive with $k$ if it is small ($<\log$ (number of variables))
- The number of tree should be large enough. (>30)

Advantages

- Random forests are competitively with other popular algorithms, such as boosting, SVM in accuracy
- Simple to use, not sensitive with parameters
- Do not overfit data, no regularization
- Accurate probability estimation
- Resistance to noise
- No pruning, less imbalance problem
Perceptrons vs. Naïve Bayes

Naïve Bayes

\[ P(C|X_1, X_2, \ldots, X_i) = \frac{P(C) \prod_{i=1}^{n} P(X_i|C)}{P(X_1, X_2, \ldots, X_i)} \]

\[ \ln \frac{P(+|X_1, X_2, \ldots, X_i)}{P(-|X_1, X_2, \ldots, X_i)} = \ln \frac{P(+)}{P(-)} + \sum_{i=1}^{n} \ln \frac{P(X_i|+)}{P(X_i|-)} \]

Perceptrons

\[ o(X_1, X_2, \ldots, X_i) = w_0 + \sum_{i=1}^{n} w_i \cdot x_i \]

• Similarity: a generalized linear model, and the representational power can be increased by structure learning (multi-layer, unrestricted Bayesian)

• Parameter learning:
  naive Bayes: generative learning (frequency estimate)
  perceptrons: discriminative learning (perceptron rule, gradient descent)

Generative vs. Discriminative learning

<table>
<thead>
<tr>
<th></th>
<th>generative learning</th>
<th>discriminative learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objective function</td>
<td>( P(C,X_1,\ldots,X_i) )</td>
<td>( P(C</td>
</tr>
<tr>
<td>Training time</td>
<td>efficient</td>
<td>slow</td>
</tr>
<tr>
<td>Sample efficiency</td>
<td>Better(&lt;100)</td>
<td>worse</td>
</tr>
<tr>
<td>overfitting</td>
<td>no</td>
<td>yes</td>
</tr>
</tbody>
</table>
Generative vs. Discriminative Learning

- variables are independent: generative and discriminative learning may learn the same parameters

- duplicated variables: discriminative learning learns better parameters than generative learning

- XOR functions among variables: both of them need to resort to structure learning

Some Observations in Practice

variables are dependent

duplicated variables

XOR

Naive Bayes

Decision trees

Perceptrons