

CS573: Homework 3 Solution

Part of the solution is compiled from Jaewoo L.'s and Pedro P.'s submissions

December 11, 2010

Tree Augmented Naive Bayes

In this assignment we investigate a graphical model that extends Naive Bayes (NB) for classification. NB uses a generative model which assumes conditional independence between the attributes ($\mathbf{X} = \{X_1, \dots, X_n\}$) given the class (C). This can be represented using a directed graphical model where the nodes are $V = \{X_i | 1 \leq i \leq n\} \cup \{C\}$ and the edges are $E = \{(C, X_i) | 1 \leq i \leq n\}$. Tree Augmented Naive Bayes (TAN) augments this graphical model with a set of edges $E' \subset \mathbf{X} \times \mathbf{X}$. The restriction on E' is that every X_i has exactly one parent from \mathbf{X} (in addition to C), except for one X_i that has no parents other than C . Figure 1 gives a general description of Naive Bayes versus TAN.

To estimate a TAN model, the learning algorithm needs to search over the structure of the model (i.e., which edges to add among the attributes) and then estimate the parameters of the conditional probability distributions (CPDs). Learning the structure of TAN model requires a procedure that consists of five main steps:

1. Compute $I_P(X_i; X_j | C)$ between each pair of attributes, $i \neq j$, where

$$I_P(X; Y | z) = \sum_{x,y,z} P(x, y, z) \log \frac{P(x, y | z)}{P(x | z)P(y | z)}$$

2. Build a complete undirected graph in which the vertices are the attributes X_1, \dots, X_n . Annotate the weight of an edge connecting X_i to X_j by $I_P(X_i; X_j | C)$.
3. Build a maximum weighted spanning tree.
4. Transform the resulting undirected tree to a directed one by choosing a root variable and setting the direction of all edges to be outward from it.
5. Construct a TAN model by adding a vertex labeled by C and adding an arc from C to each X_i .

Programming assignment

In this question you will implement and evaluate NB and TAN. You can use a language of your choice to implement the algorithms (e.g., Java, C, Python, R). Please hand in a hard copy of your code with the assignment. Evaluate the two algorithms on the *Congressional Records* dataset from the UCI machine learning repository. The data set includes votes for each of the U.S. House of Representatives Congressmen on the 16 key votes identified by the Congressional Quarterly Almanac. The data set includes 435 records; each person is classified as democrat or republican and the attribute values correspond to the y/n/missing positions on each of the 16 votes.

1. Implement a Naive Bayes algorithm. (10 pts)
 - (a) Given a dataset D with attributes \mathbf{X} and class C as described above, write down the expression for the factored joint distribution $P(X_1, \dots, X_n, C)$ for NB.

Sample solution:

Assuming independence between the attributes (\mathbf{X}) given the class label (C), the expression should be:

$$P(C|\mathbf{X}) = \frac{P(\mathbf{X}|C)P(C)}{P(\mathbf{X})} \propto \prod_{i=1}^m P(X_i|C)P(C)$$

- (b) Specify the set of parameters that need to be estimated and outline the maximum likelihood estimate for an example CPD.

Sample solution:

The set of parameters that need to be estimated are:

$$P(C), P(X_1|C), P(X_2|C), \dots, P(X_n|C)$$

The maximum likelihood estimation for the following example CPD: $P(x_1 = \text{yes}, c = \text{democrat})$

$$= \frac{\text{count of } x_1 = \text{yes and } c = \text{democrat}}{\text{count of } c = \text{democrat}}$$

2. Implement a TAN Bayes algorithm. (10 pts)
 - (a) Let G be a TAN model with $\pi(X_i) = j$ if X_j is a parent of X_i in G or $\pi(X_i) = \emptyset$ if X_i has no parents in \mathbf{X} (note that G is determined only by π , given X_i and C). Write down an expression for the factored joint distribution $P(X_1, \dots, X_n, C)$ for G . You may use π in your expression.

Sample solution:

The factored joint distribution $P(X_1, \dots, X_m, C)$ for G can be given by:

$$P(C) \prod_{i=1}^m F(X_i, C)$$

$$F(X_i, C) = \begin{cases} P(X_i|X_j, C) & \text{if } \pi(X_i) \neq \emptyset \\ P(X_i|C) & \text{if } \pi(X_i) = \emptyset \end{cases}$$

where X_j is the parent of X_i given by π .

- (b) Specify the set of parameters that need to be estimated and outline the maximum likelihood estimate for an example CPD.

Sample solution:

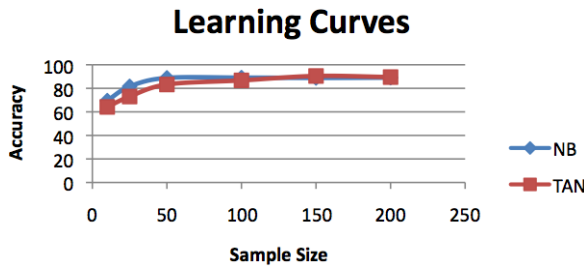
The set of parameters are: $P(C), F(X_1, C), F(X_2, C), \dots, F(X_n, C)$.

The maximum likelihood estimation for the following example CPD: $P(x_1 = \text{yes} | c = \text{democrat}, x_{\pi(x_1)} = \text{no})$

$$= \frac{\text{count of } x_1 = \text{yes and } c = \text{democrat and } x_{\pi(x_1)} = \text{no}}{\text{count of } c = \text{democrat and } x_{\pi(x_1)} = \text{no}}$$

3. Evaluate the algorithms using cross validation and learning curves. (10 pts)
 Create 10 training/test set pairs using 10-fold cross validation. From each training set, learn models with [10, 25, 50, 100, 150, 200] randomly selected examples from the training set. Apply those models to the test set, measuring classification accuracy. Graph the learning curves for each model, reporting average accuracy over the ten training/test splits. How do the two algorithms compare? Discuss the results.

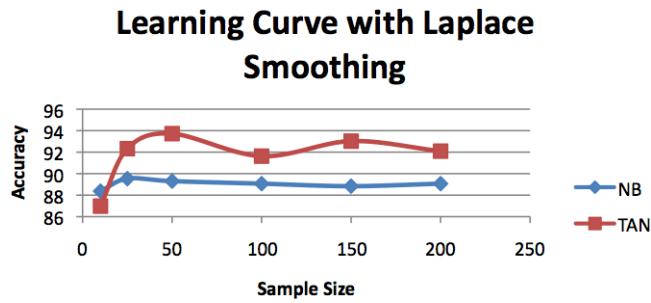
Sample solution:



We can see from the graphs that for both models, after certain point of accuracy (90%) it does not increase significantly with more data. We can see that naive Bayes starts with better classification performance than TAN, but as the size of the data increases the performance of TAN reaches and overpasses the performance of naive Bayes. We would expect that TAN could achieve better performance than naive Bayes with a more complex model but requires more data in order to avoid overfitting.

4. Smoothing (5 pts)
 Implement Laplace smoothing in the parameter estimation. For an attribute X_i with k values, Laplace correction adds 1 to the numerator and k to the denominator of the MLE for $P(X_i|C)$. Describe what the Laplace smoothing technique should be for the TAN model CPDs. Discuss what impact you expect smoothing to have on the results of each model. Which model do you expect it to impact more? Rerun the experiment (with the same training/test splits) with smoothing implemented and plot the learning curves to see how the change affects performance.

Sample solution:



The performance of the TAN model would increase more after applying the Laplace smoothing. This is because TAN is a more complex model and has to handle more conditional probabilities than naive Bayes. With a small data set and a complex model, Laplace smoothing has a bigger impact on the zero counts and overfitting. We can see from the graph that it was the case. Since the zero counts are smoothed the classification performance increases by almost 10% in average. On the other hand the Laplace smoothing did not have a significant impact on the naive Bayes model.