Naively Boosted

Minghui Jiang

Department of Computer Science, Montana State University, Bozeman, MT 59717-3880, USA.
jiang@cs.montana.edu

1 Background

Elkan [1] won the first place out of 45 entries in the KDD Cup data mining competition [4] at KDD'97, the Third International Conference on Knowledge Discovery and Data Mining, August 1997. Elkan's winning method is very simple:

1. Obtain a maximum likelihood classifier based on the naive Bayes model.
2. Enhance it with AdaBoost (Freund and Schapire [2]) to obtain an ensembled classifier.

Naive Bayes model is a well-known model of Bayesian network, with a long history dating back to 1950s. Because of its simplicity and computational efficiency, naive Bayes model is immensely popular; however, it is still a rather *naive* model, as its name suggests. Elkan's success in KDD'97 can only be explained by his judicious adoption of the ingenious AdaBoost algorithm. (Note the temporal closeness of these two events.) Freund and Schapire's AdaBoost paper [2] recently won the 2003 Gödel Prize; it is the first AI paper in the prize's history. Gödel Prize usually awards outstanding papers in the area of theoretical computer science, especially complexity theory; past winners include such famous theoretical results as the PCP theorem, prime factorization on quantum computers, and interactive proof systems.

2 Bayesian Network

Bayesian network models the world as a directed graph. Each node represents a random variable. An edge from node A to node B signifies an influence of A on B—a dependency between them. Each node has a conditional probability table (CPT), specifying the conditional probability of each possible value of the node given each possible combination of values of its parent nodes. An example (CPTs omitted):

- BirdFlu, MadCow, Chicken, Beef, KFC[1], SirScottsOasis[2].
- (BirdFlu, Chicken), (MadCow, Beef), (Beef, Chicken), (Chicken, KFC), (Beef, SirScottsOasis).

Posterior probability inference example: $\Pr(madcow \mid chicken \wedge \neg beef) = ?$.

The graph structure of a Bayesian network captures the dependencies among the random variables; equipped with the CPTs, a Bayesian network provides a complete description, the full joint distribution, of the variables. A Bayesian network can be much simpler than a single-table entry-by-entry full joint distribution, since edges only exist between dependent variables. However, exact inference in general Bayesian network is NP-hard (by an easy reduction from SAT)!

[1] The one in Vietnam, not here in Bozeman...
[2] Sir Scott's Oasis, a steakhouse in Manhattan, MT, has awesome prime rib.

3 Naive Bayes Model

Naive Bayes model simplifies Bayesian network: one class variable, many attribute variables; one edge from the class variable to each attribute variable. A naive Bayesian network is a tree of depth one: the class variable is the root; the attribute variables are the leaves. An example:

- **BirdFlu**, Chicken, KFC.
- (BirdFlu, Chicken), (BirdFlu, KFC).

If all variables are boolean, then a naive Bayesian network can be specified by $2m+1$ parameters, where m is the number of attribute variables. To train the maximum likelihood classifier, we have a remarkably simple learning algorithm: the parameters are computed directly by counting relevant instances in the training data. For example:

$$\Pr(birdflu \mid chicken) = \frac{N_{birdflu \wedge chicken}}{N_{chicken}}.$$

To make a prediction (with the trained classifier) given the attributes $chicken \wedge \neg kfc$, we simply evaluate the probability $\Pr(birdflu \mid chicken \wedge \neg kfc)$ by exact inference and check whether it is larger than $\frac{1}{2}$:

$$\begin{aligned}
\Pr(birdflu \mid chicken \wedge \neg kfc) &= \frac{\Pr(birdflu \wedge chicken \wedge \neg kfc)}{\Pr(chicken \wedge \neg kfc)} \\
&= \alpha \Pr(chicken \wedge \neg kfc \mid birdflu) \Pr(birdflu) \\
\text{Conditional independence!} \quad &= \alpha \Pr(chicken \mid birdflu) \Pr(\neg kfc \mid birdflu) \Pr(birdflu).
\end{aligned}$$

The maximum likelihood classifier thus obtained is efficient in doing inference, but it might not be good enough in making predictions. The optimal Bayesian learning method makes predictions using *all* hypotheses (possible classifiers), weighted by their posterior probabilities, instead of using a single maximum likelihood hypothesis [5]. This optimal method is certainly unrealistic here because the parameters in the naive Bayes model, the conditional probabilities, are continuous numbers—there are infinite candidate classifiers! We have to use a less ambitious ensemble learning method.

4 AdaBoost

AdaBoost is an ensemble learning method that approximates the optimal Bayesian learning. Despite the puzzling details of its weight adjustment procedures [2], AdaBoost has a simple and elegant idea, which can be summarized in one sentence: starting with a single classifier (the maximum likelihood classifier, for example), the algorithm generates new classifiers to add to its ensemble by successively re-weighting the training examples to emphasize the more difficult ones, and, at the same time, improves the performance of the ensemble by "promoting" the better classifiers.

References

1. Charles Elkan. Boosting and naive Bayesian learning. Technical Report No. CS97-557, Department of Computer Science and Engineering, University of California, San Diego, 1997.
2. Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Science*, 55(1):119–139, 1997.
3. 2003 Gödel Prize. `http://sigact.acm.org/prizes/godel/2003.html`, 2003.
4. KDD Cup data mining competition. `http://www.acm.org/sigkdd/kdd2004/previous.html`, 2004.
5. Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach (2nd Edition).* Prentice Hall, 2003.