Augmented Semi-naive Bayes Classifier

Bojan Mihaljevic, Pedro Larrañaga, and Concha Bielza

Computational Intelligence Group, Departament de Inteligencia Artificial
Facultad de Informática, Universidad Politécnica de Madrid
{bmihaljevic,pedro.larranaga,mcbielza}@fi.upm.es

Abstract. The naive Bayes is a competitive classifier that makes strong conditional independence assumptions. Its accuracy can be improved by relaxing these assumptions. One classifier which does that is the semi-naive Bayes. The state-of-the-art algorithm for learning a semi-naive Bayes from data is the backward sequential elimination and joining (BSEJ) algorithm. We extend BSEJ with a second step which removes some of its unwarranted independence assumptions. Our classifier outperforms BSEJ and five other Bayesian network classifiers on a set of benchmark databases, although the difference in performance is not statistically significant.

Keywords: semi-naive Bayes, tree augmented naive Bayes, Bayesian network classifiers.

1 Introduction

A classifier is a function which uses a set of features of an object to assign it to a class. The naive Bayes classifier [1,2] is an effective probabilistic classifier. It assumes that the features are independent given the class. This assumption is violated in many domains and more accurate classification can often be obtained by avoiding unwarranted independence assumptions [3]. A common approach to this is to augment naive Bayes by accounting for interactions between features, obtaining an augmented naive Bayes model [3]

Semi-naive Bayes [4] is one such augmented naive Bayes classifier. It assumes that correlations exist only inside disjoint subsets of features. No independence assumptions are made within a feature subset, i.e., each feature directly depends on every other. The best-known algorithm for learning a semi-naive Bayes is the backward sequential elimination and joining (BSEJ) algorithm [4]. This algorithm tends to capture few correlations among the features [3].

We set out to extend the BSEJ algorithm with a second step which removes some of its independence assumptions that are not warranted by the data. We use tests of conditional independence to identify the unwarranted independences. We augment the semi-naive Bayes model with a restricted set of interactions. This procedure is inspired by the selective tree augmented naive Bayes algorithm [5].

We report an empirical comparison of our proposal with the BSEJ algorithm and with five other reference Bayesian network classifiers.

C. Bielza et al. (Eds.): CAEPIA 2013, LNAI 8109, pp. 159–167, 2013.

This paper is organized as follows. Sections 2 introduces Bayesian network classifiers. Section 3 explains the backward sequential elimination and joining (BSEJ) algorithm. Section 4 describes the selective tree augmented naive Bayes algorithm. Section 5 explains the proposed extension of BSEJ. Section 6 reports the empirical evaluation of our proposal. Section 7 sums the paper up.

2 Bayesian Network Classifiers

We use upper-case letters to denote variables (X) and lower-case letters (x) to denote variable values. We use boldface letters to denote multidimensional vectors. A problem domain is described with n predictive variables or features $\mathbf{X} = (X_1, \ldots, X_n)$ and a class variable C. In our setting, all variables are discrete with $x_i \in \{1, \ldots, r_i\}$ and $c \in \{1, \ldots, r_c\}$. A *Bayes classifier* assigns a vector of feature values $\mathbf{x}$ to the most probable class, i.e.

$$c^* = \arg\max_c p(c|\mathbf{x}).$$

A Bayesian network classifier [3] uses a Bayesian network [6] to encode $p(c, \mathbf{x})$. A Bayesian network consists of two components: a directed acyclic graph G and a set of parameters $\mathbf{\Theta}$. Each node V in the graph corresponds to a random variable and the arcs represent direct dependencies between the variables. G encodes the conditional independence assumptions about the variables: a variable V is independent of its nondescendants given $\mathbf{Pa}(V)$, its parents in G. The parameters $\mathbf{\Theta}$ quantify the network by specifying the local probability distribution for each V, $p(v|\mathbf{pa}(v))$, where $\mathbf{pa}(v)$ is a value of the set of variables $\mathbf{Pa}(V)$. A Bayesian network classifier assigns $\mathbf{x}$ to the class that maximizes $p(c, \mathbf{x})$ since $\arg\max_c p(c, \mathbf{x}) = \arg\max_c p(c|\mathbf{x})$.

The best-known Bayesian network classifier is the naive Bayes. It assumes that the features are conditionally independent given the class (see Fig. 1a for its network structure), factorizing $p(c, \mathbf{x})$ as

$$p(c, \mathbf{x}) = p(c) \prod_{i=1}^{n} p(x_i|c).$$

This assumption is violated in many domains and more accurate classification can often be obtained by avoiding unwarranted independence assumptions [3]. A common approach to this is to augment naive Bayes' structure with arcs between features, obtaining an augmented naive Bayes model [3].

3 Semi-naive Bayes

The semi-naive Bayes (SB) is an augmented naive Bayes classifier. It assumes that correlations exist only inside disjoint subsets of features. No independence assumptions are made within a feature subset, i.e., each feature depends directly on every other. This means that the structure of a naive Bayes is augmented with

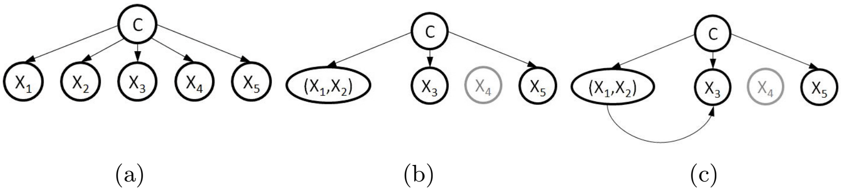

Fig. 1. Examples of Bayesian network classifier structures. Naive Bayes (a), semi-naive Bayes (b), and augmented semi-naive Bayes (c)

an arc between every pair of features in the same feature subset. For simplicity of representation, we depict the dependencies within a feature subset with a compound node corresponding to the Cartesian product of the features within the subset (see Fig. 1b). Unlike naive Bayes, the semi-naive Bayes model does not necessarily include all the features of a domain. According to the semi-naive Bayes,

$$p(c, \mathbf{x}) = p(c) \prod_{j \in Q} p(\mathbf{x}_{S_j} | c), \tag{1}$$

where $S_j \subseteq \{1, \ldots, n\}$ is the j-th feature subset, $Q = \{1, \ldots, K\}$ is the set of indices of feature subsets, and the following conditions hold: $\cup_{j \in Q} S_j \subseteq \{1, 2, ..., n\}$ and $S_j \cap S_l = \emptyset$, $j \neq l$.

The number of possible partitions of the feature set into disjoint subsets grows faster than exponential in n. That justifies the use of heuristics for learning a semi-naive Bayes from data. The backward sequential elimination and joining (BSEJ) [4] algorithm is the state-of-the-art algorithm for this purpose. It uses a greedy search which, starting from the structure of a naive Bayes (where each feature is a singleton feature subset), chooses between two operations in each step:

- Removing a feature X_i from the model
- Creating a new feature subset $\mathbf{X}_{S_k}$ by merging two subsets, $\mathbf{X}_{S_j}$ and $\mathbf{X}_{S_j}$

A cross-validation estimate of predictive accuracy is used to evaluate the candidate operations. If no operation improves the accuracy of the current structure, the search stops.

4 Selective Tree Augmented Naive Bayes

The tree augmented naive Bayes (TAN) augments the naive Bayes with a tree over the features. That is, it conditions every feature except one (the root of the tree) on exactly one other feature. The augmenting tree which maximizes the likelihood of the TAN can be efficiently found using Chow-Liu's algorithm.

The selective tree augmented naive Bayes (STAN) may remove less than $n-1$ conditional independence assumptions of the naive Bayes. Before learning the augmenting tree, STAN discards dependencies that are not statistically significant. It may occur that a subset of features has no warranted conditional dependencies on other features. In that case there can be no arcs between this feature subset and the other features, and the augmenting structure will be a forest (a set of trees) rather than a tree.

5 Augmented Semi-naive Bayes

We would like to know if correlating some of the disjoint (and conditionally independent) feature subsets of a semi-naive Bayes can improve its predictive accuracy. Just before outputting the final semi-nave Bayes model, the BSEJ algorithm considers correlating each pair of feature subsets and finds that no correlation improves its estimate of accuracy. We consider correlating a pair of feature subsets if their conditional dependency is statistically significant. We augment the semi-naive Bayes with a tree or a forest over the feature subsets (see Fig. 1c), removing at most $K-1$ unwarranted independence assumptions, where K is the number of feature subsets. We select the augmenting edges that maximize the likelihood of the model. Although correlating any feature subset pair of the final semi-naive Bayes did not improve the accuracy estimate of BSEJ, it is possible that removing several, unwarranted independence assumptions at once can improve prediction. In any case, augmenting the semi-naive Bayes in this way is fast compared to BSEJ's time complexity.

The augmented semi-naive Bayes (ASB) factorizes $p(c, \mathbf{x})$ as

$$p(c, \mathbf{x}) = p(c) \prod_{i \in R} p(\mathbf{x}_{S_i} | c) \prod_{i \in Q \setminus R} p(\mathbf{x}_{S_i} | \mathbf{x}_{j(i)}, c),$$

where Q and S_i are defined as in Equation (1), $R \subseteq Q$ is the set of indices of feature subsets that are conditioned only on the class variable (root(s) of the trees(s)), and $\{X_{j(i)}\} = \mathbf{Pa}(X_{S_i}) \setminus C$.

To test if two sets of features, $\mathbf{X}_{S_i}$ and $\mathbf{X}_{S_j}$, are conditionally independent given the class we use the χ^2 test of conditional independence (see, e.g., [7]). If the null hypothesis of conditional independence holds, then $2NI(\mathbf{X}_{S_i}; \mathbf{X}_{S_j} | C)$ asymptotically follows the χ^2 distribution with $(r_{S_i} - 1)(r_{S_j} - 1)r_c$ degrees of freedom, where N is the number of cases in our data sample, and $r_{S_i} = \prod_{k \in S_i} r_i$. The χ^2 approximation is not reliable when there are little cases in the contingency table over $\mathbf{X}_{S_j}$, $\mathbf{X}_{S_j}$, and C [8]. Following [9], we consider the χ^2 approximation to be reliable if the average cell count in the contingency table is at least 5. Also following [9], we assume conditional independence when this condition is not fulfilled. That is, we do not remove the independence assumption for a pair of feature subsets if the test of their conditional independence is unreliable.

The procedure for finding the augmenting structure is based on Chow-Liu's algorithm. First, we build a complete undirected graph $G = (K, A)$. Each vertex $j \in K$ corresponds to $\mathbf{X}_{S_j}$, a subset of features correlated in the semi-naive Bayes, and there is an edge between every two nodes i and j such that corresponding feature subsets, $\mathbf{X}_{S_i}$ and $\mathbf{X}_{S_j}$, are not conditionally independent according to the χ^2 test. As G is not necessarily a complete graph it is possibly not connected. In this case, the augmenting structure that maximizes the likelihood is not necessarily a tree but a maximum weighted forest (MWF) [10]. The MWF is given by the union of the maximum weighted spanning trees (MWST) for each connected component of G. This union of MWSTs can be found by applying Kruskal's algorithm on G (there is no need to run it separately for each connected component of G) [10].

Our procedure for augmenting the semi-naive Bayes is similar to the STAN algorithm for augmenting the naive Bayes. The differences are that ASB can remove independence assumptions between non-singleton sets of features and that it uses the standard procedure for testing for conditional independence (the one described in [7]). Namely, it seems that the authors of STAN were not aware of the test for conditional independence and therefore they developed and used a heuristic based on the χ^2 test of independence.

The full augmented semi-naive Bayes algorithm is specified more formally in Algorithm 1.

Algorithm 1. Augmented semi-naive Bayes

```
$B \leftarrow$ a semi-naive Bayes model
$\mathbf{S} \leftarrow$ a partition of features such that $\cup_{j=1}^{K} S_j = \mathbf{S}$ and $\mathbf{X}_{S_j}$ is a set of features correlated in $B$
$r_{S_j} \leftarrow \prod_{l \in S_j} r_l$, $j \in \{1, \ldots, K\}$
$G \leftarrow (K, E)$, a complete undirected graph with nodes $K$ and edges $E$
for all $i, j = 1, \ldots, K, i < j$ do
    if $\frac{N}{r_{S_j} r_{S_i} r_c} \geq 5$ and $2NI(X_{S_i}; X_{S_j}|C)$ passes the $X^2_{(r_{S_i}-1)(r_{S_j}-1)r_c}$ test at significance level $\alpha$ then
        weight of edge $i$—$j$ in $E \leftarrow I(\mathbf{X}_{S_i}; \mathbf{X}_{S_j}|C)$
    else
        remove edge $i$—$j$ from $E$
    end if
end for
$\mathbf{T} \leftarrow$ maximum weighted forest obtained by applying Kruskal's algorithm on $G$
$\mathbf{T}' \leftarrow$ for each $T \in \mathbf{T}$ choose a root node at random and direct edges away from it
for all $i, j$ such that arc $i \rightarrow j \in \mathbf{T}'$ do
    augment $B$ with arcs from each $X_l$ in $\mathbf{X}_{S_i}$ to every $X_k$ in $\mathbf{X}_{S_j}$
end for
```

6 Experimental Evaluation

6.1 Setup

We compare the augmented semi-naive Bayes (ASB) algorithm to six reference algorithms for learning Bayesian network classifiers. Two of those algorithms learn a selective naive Bayes (SNB) [11] model. The forward sequential selection (FSS) algorithm [11] performs a greedy search guided by predictive accuracy while the filter forward sequential selection (FFSS) omits from the model the features that are deemed independent of the class by the χ^2 independence test. Besides SNB, we consider the naive Bayes (NB), the tree augmented naive Bayes (TAN), the selective tree augmented naive Bayes (STAN), and the backward sequential elimination and joining (BSEJ) algorithm.

We compare the classifiers over 14 natural domains from UCI repository [12] (see Table 1). Prior to classifier comparison, we removed incomplete rows and discretized numeric features with the MDL method [13].

For the BSEJ and the FSS, we used 5-fold stratified cross-validation to estimate predictive accuracy. For statistical tests of (conditional) independence we used a significance level of 0.05 and applied the criterion of χ^2 approximation reliability. In FFSS, if a test of independence of X_i and C is not reliable, then independence is assumed and X_i is omitted from the model. For STAN, we used the same test of conditional independence as for ASB. Laplace's correction of maximum likelihood was used to estimate parameters. We estimated predictive accuracy of the classifiers with 5 repetitions of 5-fold stratified cross-validation.

The Bayesian network classifiers are implemented in the `bayesClass` [14] package for the `R` statistical environment [15]. We used the `caret` [16] package for `R` to estimate predictive accuracy with cross-validation.

Table 1. Data sets. #Instances column displays the number of complete instances

No.	Data set	#Features	#Instances	#Classes
1	`Balance Scale`	4	625	3
2	`Breast Cancer (Wisconsin)`	9	683	2
3	`Car`	6	1728	4
4	`Chess (kr vs. kp)`	36	3196	2
5	`Dermatology`	34	358	6
6	`Ecoli`	7	336	8
7	`House Voting 84`	16	232	2
8	`Ionosphere`	34	351	2
9	`Lymphography`	18	148	4
10	`Molecular Biology (Promoters)`	57	106	2
11	`Molecular Biology (Splice)`	61	3190	3
12	`Primary Tumor`	17	132	22
13	`Tic-tac-toe`	9	958	2
14	`Wine`	13	178	3

6.2 Results

Following [17], we performed Friedman's test [18,19] and Iman and Devenport's correction [20] to compare the classifiers over all the data sets. Our proposal outperforms the other methods (see Table 2 for Friedman's ranks) although the difference is not statistically significant[1].

[1] The p-value from both Friedman's and Iman and Davenport's test was 0.2.

Table 2. Average Friedman's ranks. Lower ranking means better performance. ASB = augmented semi-naive Bayes, STAN = selective tree augmented Bayes, FSS = forward sequential selection, BSEJ = backward sequential elimination and joining, FFSS = filter forward sequential selection, NB = naive Bayes, TAN = tree augmented naive Bayes.

Algorithm	Friedman's ranks
ASB	3.11
STAN	3.96
FSS	5.21
BSEJ	3.57
FFSS	4.35
NB	3.53
TAN	4.25
p-value$_{\text{Friedman}}$	0.20
p-value$_{\text{Iman-Davenport}}$	0.20

Table 3. Estimated accuracies (in %) of the compared classifiers. The best performing classifiers on a data set are marked in bold. Some data set names are shorter than in Table 1 but the order is the same. ASB = augmented semi-naive Bayes, STAN = selective tree augmented Bayes, FSS = forward sequential selection, BSEJ = backward sequential elimination and joining, FFSS = filter forward sequential selection, NB = naive Bayes, TAN = tree augmented naive Bayes.

No.	Data set	ASB	STAN	FSS	BSEJ	FFSS	NB	TAN
1	Balance Scale	72.9±2.5	73.2±2.9	**73.6±2.2**	72.8±2.3	73.3±2.3	73.3±2.3	73.2+-2.9
2	Breast Cancer	97.1±1.1	97.1±1.1	96.9±1.4	**97.5±1.0**	**97.5±1.0**	97.5±1.0	97.1±1.1
3	Car	93.3±1.6	93.5±1.5	70.0±0.1	90.0±1.8	85.1±1.7	85.3±1.4	**94.1±1.6**
4	Chess	**94.1±1.1**	92.6±0.8	**94.1±1.0**	92.2±1.1	87.8±1.4	87.8±1.4	92.4±0.9
5	Dermatology	**98.2±1.5**	98.0±1.6	95.1±3.4	**98.2±1.5**	98.0±1.6	98.0±1.6	97.1±1.7
6	Ecoli	**85.7±3.4**	**85.7±3.4**	83.4±2.8	**85.7±3.4**	**85.7±3.4**	**85.7±3.4**	84.5±3.2
7	House Voting 84	94.3±2.8	92.9±2.8	**97.0±2.4**	91.2±4.5	91.3±4.5	91.2±4.4	93.6±2.7
8	Ionosphere	92.0±3.7	91.9±3.7	90.7±3.6	90.7±3.8	90.7±4.1	90.7±4.1	**92.2±3.1**
9	Lymphography	**85.4±6.1**	82.7±5.6	78.4±7.3	85.0±6.5	82.7±7.1	84.6±6.2	83.4±6.0
10	Promoters	89.8±6.4	90.5±5.0	84±11.2	89.8±6.4	90.5±5.0	**91.7±6.2**	48.7±1.2
11	Splice	94.9±0.7	95.0±0.8	93.5±0.8	**95.5±0.8**	95.4±0.9	95.5±0.8	52.5±0.3
12	Primary Tumor	46.5±9.4	21.3±2.0	42.5±7.7	46.5±9.4	21.3±2.0	**48.3±9.3**	41.6±8.0
13	Tic-tac-toe	75.3±3.2	74.8±2.9	69.6±3.4	71.7±3.7	70.4±3.9	70.4±3.8	**75.8±2.9**
14	Wine	98.7±1.6	98.7±1.6	95.4±2.9	**98.9±1.4**	**98.9±1.4**	**98.9±1.4**	96.9±2.6

The ASB significantly[2] improves on BSEJ on four data sets (car, chess, ionosphere, and tic-tac-toe. See Table 3 for accuracies.). The BSEJ outputs a model similar to the NB on those data sets (e.g. on ionosphere it removes a single feature and accounts for one interaction) while the ASB heavily augments the BSEJ (e.g. on ionosphere it builds a full tree among feature groups). This shows that useful interactions missed by BSEJ can be recovered by ASB.

There is no significant difference between ASB and BSEJ on the remaining data sets. The ASB degrades BSEJ on only three data sets and the degradation is

[2] According to Wilcoxon's signed rank test at 5% significance level.

minor (by at most 0.6% accuracy). On four data sets the ASB model is identical to the BSEJ. One of those data sets - primary tumor - has many classes (22) and not many cases (132). This yields the conditional independence test unreliable for every pair of features and therefore no arcs are be added. On the other three data sets, lowering the significance threshold would have would have produced augmented BSEJ models (i.e. arcs would have been added).

7 Concluding Remarks

We have presented the augmented-semi naive Bayes (ASB) algorithm, a method for removing some of the unwarranted independence assumptions of a semi-naive Bayes model. The ASB is computationally inexpensive compared to the BSEJ, the algorithm used for learning a semi-naive Bayes. Our experiments show that ASB improves BSEJ in some domains without degrading it others. The ASB outperformed BSEJ and five other Bayesian network classifiers on 14 benchmark data sets, although the improvement in performance is not statistically significant. Further experiments, over more data sets, might give more conclusive results. Since ASB seems to improve BSEJ, it might be interesting to extend the approach to augmenting other Bayesian network classifier learned by maximizing predictive accuracy, such as the forward sequential selection algorithm for learning a selective naive Bayes.

Acknowledgements. This work has been partially supported by the Spanish Economy and Competitiveness Ministry through Cajal Blue Brain (C080020-09) and TIN2010-20900-C04-04 projects. The authors thank Rubén Armañanzas and anonymous reviewers for useful comments.

References

1. Minsky, M.: Steps toward artificial intelligence. Transactions on Institute of Radio Engineers 49, 8–30 (1961)
2. Duda, R., Hart, P.: Pattern Classification and Scene Analysis. John Wiley and Sons (1973)
3. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. Machine Learning 29, 131–163 (1997)
4. Pazzani, M.: Constructive induction of Cartesian product attributes. In: Proceedings of the Information, Statistics and Induction in Science Conference (ISIS 1996), pp. 66–77 (1996)
5. Blanco, R., Inza, I., Merino, M., Quiroga, J., Larrañaga, P.: Feature selection in Bayesian classifiers for the prognosis of survival of cirrhotic patients treated with TIPS. Journal of Biomedical Informatics 38(5), 376–388 (2005)
6. Pearl, J.: Probabilistic Reasoning in Intelligent Systems. Morgan Kaufmann (1988)
7. Koller, D., Friedman, N.: Probabilistic graphical models: principles and techniques. MIT Press (2009)
8. Agresti, A.: Categorical Data Analysis. Wiley (1990)

9. Yaramakala, S., Margaritis, D.: Speculative Markov blanket discovery for optimal feature selection. In: ICDM 2005: Proceedings of the Fifth IEEE International Conference on Data Mining, pp. 809–812. IEEE Computer Society, Washington, DC (2005)
10. Murphy, K.P.: Machine learning: a probabilistic perspective. The MIT Press (2012)
11. Langley, P., Sage, S.: Induction of selective Bayesian classifiers. In: Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence (UAI 1994), pp. 399–406. Morgan Kaufmann (1994)
12. Bache, K., Lichman, M.: UCI machine learning repository (2013), `http://archive.ics.uci.edu/ml`
13. Fayyad, U., Irani, K.: Multi-interval discretization of continuous-valued attributes for classification learning. In: Proceedings of the 9th International Joint Conference on Artificial Intelligence (IJCAI 1993), pp. 1022–1029. Morgan Kaufmann (1993)
14. Mihaljevic, B., Larrañaga, P., Bielza, C.: BayesClass: A package for learning Bayesian network classifiers (2013), R package version 1.0
15. R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2012)
16. Kuhn, M., Wing, J., Weston, S., Williams, A., Keefer, C., Engelhardt, A., Cooper, T.: Caret: Classification and Regression Training (2013) R package version 5.15-052
17. García, S., Fernández, A., Luengo, J., Herrera, F.: Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. Information Sciences 180(10), 2044–2064 (2010)
18. Friedman, M.: The use of ranks to avoid the assumption of normality implicit in the analysis of variance. Journal of the American Statistical Association 32(200), 675–701 (1937)
19. Friedman, M.: A comparison of alternative tests of significance for the problem of m rankings. The Annals of Mathematical Statistics 11(1), 86–92 (1940)
20. Iman, R., Davenport, J.M.: Approximations of the critical region of the friedman statistic. Communications in Statistics 9(6), 571–595 (1980)