# On Robust Trimming of Bayesian Network Classifiers

**YooJung Choi** and **Guy Van den Broeck**
Computer Science Department, University of California, Los Angeles
{yjchoi, guyvdb}@cs.ucla.edu

## Abstract

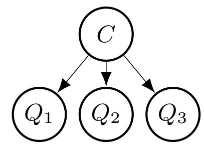
This paper considers the problem of removing costly features from a Bayesian network classifier. We want the classifier to be robust to these changes, and maintain its classification behavior. To this end, we propose a closeness metric between Bayesian classifiers, called the *expected classification agreement (ECA)*. Our corresponding trimming algorithm finds an optimal subset of features and a new classification threshold that maximize the expected agreement, subject to a budgetary constraint. It utilizes new theoretical insights to perform branch-and-bound search in the space of feature sets, while computing bounds on the ECA. Our experiments investigate both the runtime cost of trimming and its effect on the robustness and accuracy of the final classifier.

## 1 Introduction

Bayesian classification plays a prominent role throughout machine learning [Wu *et al.*, 2008; Laidlaw *et al.*, 1998; Metsis *et al.*, 2006]. In this setting, one has a model that specifies a probability distribution $\Pr$ over a set of variables, including class variable $C$ and attributes or features $\mathbf{F} = \{F_1, \ldots, F_n\}$. Given a particular instance, described as an assignment to features $\mathbf{f} = \{f_1, \ldots, f_n\}$, this model is used to compute the posterior probability $\Pr(C|f_1, \ldots, f_n)$ which is then compared against a threshold $T$ to classify the instance.

In practice, observing features often has a cost, and one typically needs to keep it within a given budget. For example, features in a medical diagnosis may be invasive, time-consuming, or expensive medical tests [Kononenko, 2001]. Similar issues arise in active sensing [Gao and Koller, 2011], adaptive testing [Millán and Pérez-De-La-Cruz, 2002; Munie and Shoham, 2008], and robotics [Kollar and Roy, 2008]. This problem has been studied from different angles, often under the umbrella of *feature selection*. For example, one may select features at learning time based on their relevance, redundancy, or classification accuracy [Kira and Rendell, 1992; Yu and Liu, 2004]. Alternatively, features may be selected at prediction time based on their expected misclassification cost or information gain [Bilgic and Getoor, 2011; Krause and Guestrin, 2009; Zhang and Ji, 2010]. Such probabilistic objectives are computed on the distribution of the Bayesian classifier.

$T = 0.07$




| $C$ | $\Pr(Q_1 = + \mid C)$ |
|---|---|
| $+$ | 0.9 |
| $-$ | 0.3 |

| $\Pr(C = +)$ |
|---|
| 0.1 |

| $C$ | $\Pr(Q_2 = + \mid C)$ |
|---|---|
| $+$ | 0.9 |
| $-$ | 0.6 |

| $C$ | $\Pr(Q_3 = + \mid C)$ |
|---|---|
| $+$ | 0.4 |
| $-$ | 0.2 |

Figure 1: Naive Bayes classifier for a quiz scenario where answers on $\mathbf{Q} = \{Q_1, Q_2, Q_3\}$ (features) depend on knowledge $C$ (class)

This paper approaches the problem from a different perspective, which we call classifier *trimming*. In addition to selecting features that fit the budget, trimming adjusts the threshold $T$ to induce a new classifier. Moreover, instead of simply optimizing the predictive accuracy, we want trimming to be robust. That is, we want to preserve the original classifier's *general behavior*. Our motivation is two-fold. First, Bayesian classifiers often incorporate significant expert knowledge in the form of priors, structural assumptions, and choice of distribution class [Lucas, 2001]. This is particularly true in medical applications where data is scarce [Bellazzi and Zupan, 2008]. Second, two classifiers with the same predictive quality can exhibit vastly different behavior and failure modes. For example, Zhao *et al.* [2017] describe two classifiers with a similar accuracy, but markedly different amounts of gender bias in their predictions. In either scenario, it is essential to retain the desired behavior of the original classifier during trimming.

Figure 1 depicts a classifier utilizing three features $\mathbf{Q} = \{Q_1, Q_2, Q_3\}$ with a threshold of $T = 0.07$. Consider two possible trimmings of this classifier: one obtained by removing $Q_2$ and adjusting the threshold to $0.10$, the other with $Q_1$ removed and the threshold changed to $0.30$. The trimmed classifiers are clearly less expensive than the original one, but how do we quantitatively compare and choose between these trimmings? To answer this question, we introduce the notion of *expected classification agreement (ECA)*. It is an expectation of the two classifiers agreeing on instances, measuring how much behavior from the original classifier is preserved.

Probabilistic graphical models, such as Bayesian networks, are often used to represent the Bayesian classifier's distribution. We propose an algorithm to find the best trimming of a Bayesian network classifier subject to a budgetary constraint. The algorithm selects features and chooses a new classification threshold in order to maximize the ECA. We also propose a specialized algorithm for the case of naive Bayes classifiers [Friedman *et al.*, 1997; Cheng and Greiner, 1999] that exploits the naive Bayes independence assumptions for more efficient trimming. These novel trimming algorithms are based on the following progression of ideas. First, we show how an existing compilation algorithm to compute *expected same-decision probability (E-SDP)* can be modified to compute the ECA between a classifier and its trimming [Choi *et al.*, 2017]. This objective was previously used for feature selection where the classification threshold remains fixed [Chen *et al.*, 2015]. Second, we propose an upper bound on the ECA that can be computed more efficiently, enabled by our formulation that adjusts the threshold. Lastly, we use this upper bound to effectively trim classifiers with branch-and-bound search.

Finally, with evaluation on real-world data, we show that our approach finds robust trimmings and demonstrate the relationship between robustness and accuracy. We also illustrate the importance of optimizing the threshold for both classification similarity and efficiency of search. Moreover, we show that our trimming approach consistently returns a classifier that is significantly more similar to the original classifier than selecting features based on information gain.

## 2 Expected Classification Agreement

We use the standard notation where variables are denoted by upper case letters ($X$) and their instantiations by lower case letters ($x$). Sets of variables are denoted in bold upper case ($\mathbf{X}$) and their joint instantiations in bold lower case ($\mathbf{x}$). Concatenations of sets ($\mathbf{XY}$) represent their union.

A binary Bayesian classifier is a tuple $\alpha = (C, \mathbf{F}, T)$, where $C$ is a binary class variable, $\mathbf{F}$ are (possibly multi-valued) features, and $T$ is a threshold. On a joint probability distribution $\Pr(.)$ over variables $C$ and $\mathbf{F}$, the classification function is

$$C_T(\mathbf{f}) = \begin{cases} c, & \text{if } \Pr(c \mid \mathbf{f}) \geq T \\ \bar{c}, & \text{otherwise.} \end{cases}$$

For example, with a threshold of $0.5$, an instance will be classified into the more probable class after observing its features.

Next, we motivate and define our proposed closeness measure between classifiers, quantifying their expected agreement.

### 2.1 Example and Motivation

Consider again the scenario shown in Figure 1, where an instructor uses a quiz to test students' knowledge. The quiz contains three independent questions: $Q_1$ is strongly indicative of being knowledgeable, $Q_2$ is an easy question, and $Q_3$ is a hard question (only 40% of the knowledgeable students answer it correctly). The subject of this quiz is quite difficult, and only 10% of the students are expected to master it, as reflected by the prior on class variable $C$. Hence, the instructor sets a lenient threshold of $T = 0.07$ to avoid failing students who may have grasped the subject.

According to this classifier, a student will pass the quiz precisely when their answer matches one of the following three (out of eight) outcomes: $\{Q_1 = +, Q_2 = +, Q_3 = +\}$, $\{Q_1 = +, Q_2 = +, Q_3 = -\}$, and $\{Q_1 = +, Q_2 = -, Q_3 = +\}$. Moreover, the probability of seeing one of these outcomes is $32\%$: the fraction of students that are expected to pass the quiz. Suppose now that we drop questions $Q_1$ and $Q_2$, relying solely on question $Q_3$ to evaluate students (using the same threshold). Since $\Pr(C = + \mid Q_3)$ is always greater than $T = 0.07$, all students will pass the quiz, completely ignoring the test results. Alternatively, we can make more intuitive use of the test question and pass only the students who answered $Q_3$ correctly. This is equivalent to comparing $\Pr(C = + \mid Q_3)$ against a new threshold of $T = 0.15$. Using this new threshold, we will now obtain the same student assessment on five test outcomes,[1] whose probabilities add up to $75\%$. This is the expected classification agreement (ECA). In particular, we say that the two classifiers $\alpha = (C, \{Q_1, Q_2, Q_3\}, 0.07)$ and $\beta = (C, \{Q_3\}, 0.15)$ have an ECA of $75\%$.

### 2.2 Formalization

We now formalize the notion of ECA and classifier trimming.

**Definition 1** *Let $\alpha = (C, \mathbf{F}, T)$ be a Bayesian classifier using distribution* $\Pr(.)$. *The classifier $\beta = (C, \mathbf{F}', T')$ is a* trimming *of $\alpha$ if it uses the same class variable $C$ and distribution* $\Pr(.)$ *as $\alpha$, and a subset of its features (i.e., $\mathbf{F}' \subset \mathbf{F}$).*

**Definition 2** *Let $\alpha = (C, \mathbf{F}, T)$ be a Bayesian classifier and let $\beta = (C, \mathbf{F}', T')$ be one of its trimmings. The* expected classification agreement (ECA) *between these classifiers is:*

$$\mathrm{ECA}(\alpha, \beta) = \sum_{\mathbf{f}} [C_T(\mathbf{f}) = C_{T'}(\mathbf{f}')] \cdot \Pr(\mathbf{f}).$$

*Here, $\mathbf{f}'$ is the subset of instantiation $\mathbf{f}$ pertaining to variables in $\mathbf{F}'$, and $[.]$ is an indicator function (evaluates to $1$ when its argument is true and to $0$ otherwise).*

Section 1 asks to compare trimmings of classifier $\alpha$ in Figure 1. The first trimming has $\mathrm{ECA}(\alpha, (C, \{Q_1, Q_3\}, 0.10)) = 91\%$ while the second has $\mathrm{ECA}(\alpha, (C, \{Q_2, Q_3\}, 0.30)) = 68\%$.

We are now ready to define the *classifier trimming problem* more formally. The input to this problem is a binary Bayesian classifier $\alpha = (C, \mathbf{F}, T)$, a positive cost for each feature in $\mathbf{F}$, and a budget $B$. The output is a subset of features $\mathbf{F}^\star \subseteq \mathbf{F}$ whose sum of costs is at most $B$ and a threshold $T^\star$, leading to a trimmed classifier $\beta^\star = (C, \mathbf{F}^\star, T^\star)$ that maximizes the ECA with $\alpha$:

$$\beta^\star = \arg\max_{\beta} \mathrm{ECA}(\alpha, \beta).$$

In other words, we wish to find a solution to the following optimization problem:

$$\mathrm{ECA}^\star = \max_{\mathbf{F}' \subseteq \mathbf{F}} \max_{T'} \mathrm{ECA}(\alpha, (C, \mathbf{F}', T'))$$
$$\text{s.t.} \sum_{F' \in \mathbf{F}'} \mathrm{cost}(F') \leq B$$

This problem can alternatively be described as feature subset selection using the following criterion.

[1]The two classifiers will disagree on $\{Q_1 = +, Q_2 = +, Q_3 = -\}$, $\{Q_1 = -, Q_2 = +, Q_3 = +\}$ and $\{Q_1 = -, Q_2 = -, Q_3 = +\}$.

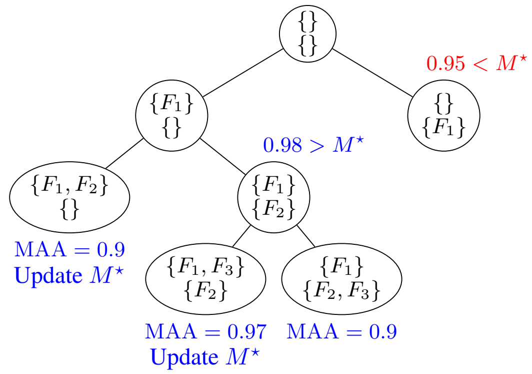



Figure 2: Branch-and-Bound search tree to select a subset with budget $B = 1.5$ among features $\{F_1, F_2, F_3, F_4\}$ with costs $\{0.5, 1.0, 1.0, 2.0\}$ respectively. Every node contains a set of included features and a set of excluded features.

**Algorithm 1** ECA-TRIM$(\mathbf{I}, \mathbf{E}, b)$

**Input:**
$\alpha$ : Bayesian classifier $(C, \mathbf{F}, T)$; $B$ : budget
**Data:**
$\mathbf{I} \leftarrow \emptyset$, $\mathbf{E} \leftarrow \emptyset$: set of included/excluded features
$b \leftarrow B$: remaining budget
$\mathbf{F}^\star, M^\star, T^\star$ : optimal subset, MAA value, and threshold
**Output:** Optimal trimmed classifier $\beta^\star = (C, \mathbf{F}^\star, T^\star)$

1: **if** $b \geq 0$ **then**
2: $(m, T_m) \leftarrow \text{MAA}(\mathbf{I})$
3: **if** $m > M^\star$ **then** $M^\star \leftarrow m$; $\mathbf{F}^\star \leftarrow \mathbf{I}$; $T^\star \leftarrow T_m$
4: **if** $\min_{F \in \mathbf{F} \setminus (\mathbf{I} \cup \mathbf{E})} \text{cost}(F) \leq b$ **then**
5: $m \leftarrow \text{UB}(\mathbf{F} \setminus \mathbf{E})$
6: **if** $m \leq M^\star$ **then return**
7: $F \leftarrow$ a feature from $\mathbf{F} \setminus (\mathbf{I} \cup \mathbf{E})$
8: ECA-TRIM$\big(\mathbf{I} \cup \{F\}, \mathbf{E}, b - \text{cost}(F)\big)$
9: ECA-TRIM$\big(\mathbf{I}, \mathbf{E} \cup \{F\}, b\big)$

**Definition 3** *Let $\alpha = (C, \mathbf{F}, T)$ be a Bayesian classifier. The* maximum achievable agreement (MAA) *for feature subset $\mathbf{F}' \subseteq \mathbf{F}$ is defined as:*

$$\text{MAA}_\alpha(\mathbf{F}') = \max_{T'} \text{ECA}(\alpha, (C, \mathbf{F}', T')).$$

The $\text{MAA}_\alpha(\mathbf{F}')$ corresponds to the maximum ECA that is achievable by a trimmed classifier with features $\mathbf{F}'$. Hence, the classifier trimming problem reduces to searching for the subset of features that fits within the budget and maximizes the MAA. We will drop the subscript $\alpha$ when clear from context.

## 3 Searching for an Optimal Trimming

In this section, we describe our approach to search for an optimal trimming of Bayesian classifiers, or equivalently, selecting a feature subset with optimal MAA.[2] Our approach is based on a branch-and-bound search algorithm similar to Narendra and Fukunaga [1977] and Kolesar [1967]. As shown in Algorithm 1, we run a depth-first search through a binary tree where each node is branched into two nodes: one that includes and one that excludes a feature. Each node then represents the set of features that are included by the path from the root to that node. The algorithm computes the MAA at each node if the represented feature subset fits within the budget, keeping track of the best subset and its MAA at each point in search, as in Lines 1–3. In particular, this means that we compute the MAA even if the subset does not exhaust the budget, because MAA does not necessarily increase as the subset size grows.

The essence of the algorithm is pruning subtrees without affecting the optimality of the solution. That is, the algorithm finds the optimal solution without generating the full binary tree. Suppose given any node, we know the largest value of MAA that its descendants can achieve (UB). Then we can safely prune the subtree rooted at that node if the bound does not exceed the current best score. This correspond to backtracking the search, as shown in Line 6 in Algorithm 1. Formally, let $\mathbf{E}$ be the set of features that were excluded by the path to a certain node. Each descendant node will then represent a subset of $\mathbf{F} \setminus \mathbf{E}$. Hence, if we can compute an upper bound on MAA for all subsets of $\mathbf{F} \setminus \mathbf{E}$, then we can successfully prune intermediate nodes in the search tree.

Figure 2 illustrates an example search execution. The tree is traversed depth-first, from left to right. If a node represents a feature subset within budget, its MAA is computed, and the score $M^\star$ is updated accordingly. Otherwise, the upper bound is computed and compared against the current best value. For example, we backtrack the search after excluding $F_1$, because the upper bound on MAA for subsets of $\{F_2, F_3, F_4\}$ is 0.95 which is smaller than the current $M^\star = 0.97$. Note that this particular subset has a cost of 4.0 and does not fit within the budget, but we still compute the bound on it for pruning. That is, the algorithm computes the upper bound on MAA of the set $\mathbf{F} \setminus \mathbf{E}$ and it subsets, even if their cost may exceed the budget. We will later show experimentally that the pruning of subtrees makes these extra computations worthwhile.

## 4 Maximum Potential Agreement

We now introduce an upper bound for the MAA and show how it can be used in the search for an optimal trimming.

**Definition 4** *Consider a Bayesian classifier $\alpha = (C, \mathbf{F}, T)$ . Let $\mathbf{F}' \subseteq \mathbf{F}$ be a subset of its features, and let $\mathbf{R} = \mathbf{F} \setminus \mathbf{F}'$. The* maximum potential agreement *(MPA) is*

$$\text{MPA}_\alpha(\mathbf{F}') = \sum_{\mathbf{f}'} \max_c \left\{ \sum_{\mathbf{r}} [C_T(\mathbf{f}'\mathbf{r}) = c] \cdot \Pr(\mathbf{f}'\mathbf{r}) \right\}.$$

Intuitively, the MPA is the expected agreement between a Bayesian classifier $\alpha$ and a hypothetical classifier $\gamma$ that classifies an instance $\mathbf{f}'$ into the class that is more likely after observing the remaining features in $\mathbf{R}$. Note that such classifier $\gamma$ is not a Bayesian classifier as it does not test the posterior $\Pr(c \mid \mathbf{f}')$ against a threshold. However, the MPA is still a useful computational tool due to its relationship to the MAA.

**Proposition 1** *The MPA is an upper bound on the MAA:* $\text{MAA}_\alpha(\mathbf{F}') \leq \text{MPA}_\alpha(\mathbf{F}')$.

[2] Code available at https://github.com/UCLA-StarAI/TrimBN.

In addition, the MPA is monotonically increasing, a property that we utilize later in the proposed algorithms.

**Proposition 2** *For any* $\mathbf{F}_1 \subseteq \mathbf{F}_2$, $\mathrm{MPA}_\alpha(\mathbf{F}_1) \leq \mathrm{MPA}_\alpha(\mathbf{F}_2)$.

These two propositions together imply that the MPA of $\mathbf{F}'$ also upper-bounds the MAA of all subsets of $\mathbf{F}'$.

**Corollary 1** *For any* $\mathbf{F}_1 \subseteq \mathbf{F}_2$, $\mathrm{MAA}_\alpha(\mathbf{F}_1) \leq \mathrm{MPA}_\alpha(\mathbf{F}_2)$.

Therefore, we can use the MPA as an upper bound on the MAA of a node's descendants in the branch-and-bound search algorithm for optimal trimming.

Lastly, we provide an observation that leads to a computational gain, especially in the case of naive Bayes models.

**Proposition 3** *If features* $\mathbf{F}'$ *and* $\mathbf{F} \setminus \mathbf{F}'$ *are independent given the class* $C$*, then* $\mathrm{MAA}_\alpha(\mathbf{F}') = \mathrm{MPA}_\alpha(\mathbf{F}')$.

The above property is useful because it is generally easier to compute $\mathrm{MPA}(\mathbf{F}')$ than $\mathrm{MAA}(\mathbf{F}')$, as we can maximize each instantiation $\mathbf{f}'$ separately. Moreover, in naive Bayes models, the quantity MAA that we wish to optimize is now monotonic. Thus, we need to compute this quantity only for those subsets that exhaust the budget, instead of every subset that fits within budget. Detailed proofs of above propositions can be found in the appendix.

# 5 Probabilistic Reasoning Algorithms

In this section, we first introduce the notion of same-decision probability and formalize its connection to the ECA. We then describe our proposed algorithms to compute the MPA and the MAA that exploit this connection.

## 5.1 Same-Decision Probability and ECA

Suppose we make a decision based on whether $\Pr(c \mid \mathbf{e}) \geq T$, where $\mathbf{e}$ is the evidence collected thus far. We may ask whether to collect more evidence or to commit to the current decision. The same-decision probability (SDP) was introduced as a solution to this *stopping problem* [Choi *et al.*, 2012]. It measures how likely we are to keep our current decision even after observing more variables $\mathbf{X}$, and is defined as follows.

**Definition 5** *Consider a Bayesian classifier* $(C, \mathbf{F}, T)$ *where* $\mathbf{F}$ *includes disjoint sets of features* $\mathbf{E}$ *and* $\mathbf{X}$*. The* same-decision probability (SDP) *for* $\mathbf{X}$ *given* $\mathbf{e}$ *is defined as*

$$\mathrm{SDP}_{C,T}(\mathbf{X} \mid \mathbf{e}) = \sum_{\mathbf{x}} [C_T(\mathbf{xe}) = C_T(\mathbf{e})] \cdot \Pr(\mathbf{x} \mid \mathbf{e}).$$

A high SDP encourages one to stop collecting information and commit to the current decision, while a low SDP suggests otherwise. In the latter case, SDP also provides a solution to deciding which observations to collect next. In particular, observing a subset of variables $\mathbf{Y}$ in $\mathbf{X}$ that maximizes the *expected SDP (E-SDP)* will lead to the most robust decision *in expectation.* Hence, such subset maximally eliminates the need for further observations. This is formalized as follows.

**Definition 6** *Consider a Bayesian classifier* $(C, \mathbf{F}, T)$ *where* $\mathbf{F}$ *includes disjoint sets of features* $\mathbf{E}$*,* $\mathbf{Y}$ *and* $\mathbf{Z}$*. The* expected same-decision probability *for* $\mathbf{Z}$ *given* $\mathbf{Y}$ *and* $\mathbf{e}$ *is defined as*

$$\begin{aligned}\mathrm{SDP}_{C,T}(\mathbf{Z} \mid \mathbf{Y}, \mathbf{e}) &= \sum_{\mathbf{y}} \mathrm{SDP}_{C,T}(\mathbf{Z} \mid \mathbf{ye}) \cdot \Pr(\mathbf{y} \mid \mathbf{e}) \\ &= \sum_{\mathbf{yz}} [C_T(\mathbf{yze}) = C_T(\mathbf{ye})] \cdot \Pr(\mathbf{yz} \mid \mathbf{e}). \quad (1)\end{aligned}$$

While the ECA and E-SDP are conceptually different notions motivated by different considerations, they are quite similar computationally as they are both expectations. The ECA is equivalent to a variant of E-SDP (denoted $\mathrm{SDP}_{C,T,T'}$) that uses two thresholds instead of one, by replacing the indicator term in Equation 1 with $[C_T(\mathbf{yze}) = C_{T'}(\mathbf{ye})]$.

**Proposition 4** *Let* $\alpha = (C, \mathbf{F}, T)$ *be a Bayesian classifier and let* $\beta = (C, \mathbf{F}', T')$ *be one of its trimmings. We then have*

$$\mathrm{ECA}(\alpha, \beta) = \mathrm{SDP}_{C,T,T'}((\mathbf{F} \setminus \mathbf{F}') \mid \mathbf{F}'),$$

*where the expected SDP is computed w.r.t. classifier* $\alpha$*.*

Therefore, one can utilize an E-SDP algorithm to compute ECA during classifier trimming. For example, the E-SDP algorithm by Choi *et al.* [2017] can be modified to evaluate the above variant of E-SDP without extra computational overhead.

## 5.2 Computing the MPA

We now describe how we compute the MPA at each search step. First, the MPA can be expressed as follows:

$$\begin{aligned}\mathrm{MPA}(\mathbf{F}') = \sum_{\mathbf{f}'} \max \Big( &\mathrm{SDP}_{C,T,0}(\mathbf{R}|\mathbf{f}'), \\ &1 - \mathrm{SDP}_{C,T,0}(\mathbf{R}|\mathbf{f}') \Big) \cdot \Pr(\mathbf{f}'). \quad (2)\end{aligned}$$

Here, $\mathrm{SDP}_{C,T,0}(\mathbf{R}|\mathbf{f}')$ is the expected probability that we will decide positive class if we observe features $\mathbf{R}$ given $\mathbf{f}'$. Then $1 - \mathrm{SDP}_{C,T,0}(\mathbf{R}|\mathbf{f}')$ is its complement: the expected probability of negative classification. Exploiting this connection to SDP, our algorithm makes heavy use of the E-SDP algorithm by Choi *et al.* [2017], of which we provide a high-level description here and refer to the original paper for details. The algorithm is based on compiling a Bayesian network into a tractable circuit representation, called a Sentential Decision Diagram (SDD) [Darwiche, 2011]. Even though compiling the circuit is computationally heavy in general, computing the E-SDP and hence the MPA is efficient once we have successfully compiled the circuit. Moreover, we can sometimes efficiently compile certain networks (e.g. high treewidth) in which traditional inference techniques become infeasible [Choi *et al.*, 2013]. As a part of the process to compute $\mathrm{SDP}_{C,T}(\mathbf{R} \mid \mathbf{F}')$, the E-SDP algorithm calculates and saves the values $\Pr(\mathbf{f}')$ and $\mathrm{SDP}_{C,T,0}(\mathbf{R} \mid \mathbf{f}')$ for each $\mathbf{f}'$. Given these values, computing the MPA for $\mathbf{F}'$ is a straightforward evaluation of Equation 2.

With the ability to compute the MPA, we can now search for optimal trimmings of naive Bayes classifiers. The condition in Proposition 3 holds for all feature subsets of a naive Bayes model, and thus the MAA of a subset is always equal to its MPA. An optimal trimming is then found as shown in Algorithm 1 where both the upper bound and value of MAA are computed using the MPA algorithm described before.

**Algorithm 2** COMPUTE-MAA

**Input:**
$\alpha$ : Bayesian network classifier $(C, \mathbf{F}, T)$; $\quad \mathbf{F}' \subset \mathbf{F}$
**Data:**
$\text{CPR}(i) \leftarrow \Pr(c \mid \mathbf{f}'_i)$ for all $i$
$\text{MAR}(i) \leftarrow \Pr(\mathbf{f}'_i)$ for all $i$
$\text{POS}(i) \leftarrow \text{SDP}_{C,T,0}((\mathbf{F} \setminus \mathbf{F}') \mid \mathbf{f}'_i)$ for all $i$
**Output:** The score $\text{MAA}(\mathbf{F}')$ and the optimal threshold $T'$

1: Sort instances $\mathbf{f}'_i$ in nondecreasing order of $\text{CPR}(i)$
2: $m \leftarrow \sum_i \text{POS}(i) \cdot \text{MAR}(i)$; $\quad m^\star \leftarrow m$
3: $t^\star \leftarrow [0, \text{CPR}(1)]$
4: **for** $i$ in $1, 2, \ldots$ **do**
5: $\quad m \leftarrow m - \text{MAR}(i) \cdot (2\text{POS(i)} - 1)$
6: $\quad$ **if** $m > m^\star$ **then**
7: $\quad\quad m^\star \leftarrow m$; $t^\star \leftarrow (\text{CPR}(i), \text{CPR}(i+1)]$
8: **return** $\text{MAA}(\mathbf{F}') = m^\star$ and any $T' \in t^\star$

### 5.3 Computing the MAA

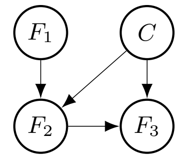
Searching for an optimal trimming of arbitrary Bayesian network classifiers requires the computation of MAA, which involves tuning the trimmed classifier's threshold to maximize the ECA. First, we utilize the observation that a change in threshold affects the value of ECA only if the class probability given some instance lies on a different side of the threshold after the change. For example, recall the trimmed classifier using only $Q_3$ from the quiz example in Section 2. We showed that a threshold of $0.15$ will result in passing only the students who answered $Q_3$ correctly. In fact, any threshold between $\Pr(C=+|Q_3=-) = 0.08$ and $\Pr(C=+|Q_3=+) = 0.18$ results in the same behavior and hence the same ECA. Therefore, the number of threshold values we need to consider is finite and in fact linear in the number of possible instances $\mathbf{f}'$.

To compute the MAA using this observation, we first express the ECA as the following:

$$\text{ECA}(\alpha, \beta) = \sum_{\mathbf{f}': \Pr(c|\mathbf{f}') \geq T'} \text{SDP}_{C,T,0}(\mathbf{R} \mid \mathbf{f}') \cdot \Pr(\mathbf{f}') + \sum_{\mathbf{f}': \Pr(c|\mathbf{f}') < T'} \Big(1 - \text{SDP}_{C,T,0}(\mathbf{R} \mid \mathbf{f}')\Big) \cdot \Pr(\mathbf{f}'). \quad (3)$$

Then we can compute the ECA for any given threshold $T'$ if we have the values of $\Pr(c \mid \mathbf{f}')$, $\Pr(\mathbf{f}')$, and $\text{SDP}_{C,T,0}(\mathbf{R} \mid \mathbf{f}')$ for each instance $\mathbf{f}'$, which we can indeed easily obtain from an execution of the E-SDP algorithm for $\text{SDP}_{C,T}(\mathbf{R} \mid \mathbf{F}')$.

Algorithm 2 shows the pseudocode to compute the MAA and the optimal threshold given these values from the E-SDP algorithm as inputs. Starting from $T' = 0$, the threshold is repeatedly incremented to just above the next lowest class probability given some feature instance $\mathbf{f}'$. With each threshold change, the ECA value is also updated by subtracting the expected probability of positive classification given that instance and adding the complement of it, weighted by the marginal probability of that instance, as in Line 5. In other words, that instance $\mathbf{f}'$ is moved from the first sum to the second in Equation 3. At the end, the highest value of ECA and its corresponding threshold is reported.




| $\Pr(F_1 = +)$ |
| --- |
| 0.9 |

| $\Pr(C = +)$ |
| --- |
| 0.6 |

| $C$ | $F_1$ | $\Pr(F_2 = + \mid C F_1)$ |
| --- | --- | --- |
| $+$ | $+$ | 0.6 |
| $+$ | $-$ | 1.0 |
| $-$ | $+$ | 0.4 |
| $-$ | $-$ | 0.5 |

| $C$ | $F_2$ | $\Pr(F_3 = + \mid C F_2)$ |
| --- | --- | --- |
| $+$ | $+$ | 0.4 |
| $+$ | $-$ | 1.0 |
| $-$ | $+$ | 1.0 |
| $-$ | $-$ | 0.4 |

Figure 3: A Bayes net over features $\{F_1, F_2, F_3\}$ and class $C$.

| $\{F_1, F_2\}$ | $\Pr(c \mid F_1 F_2)$ | $+$ class pr. | $-$ class pr. |
| --- | --- | --- | --- |
| $\{-,+\}$ | 0.75 | 0.04 | 0.04 |
| $\{+,+\}$ | 0.69 | 0.20 | 0.27 |
| $\{+,-\}$ | 0.50 | 0.30 | 0.13 |
| $\{-,-\}$ | 0.00 | 0.00 | 0.02 |

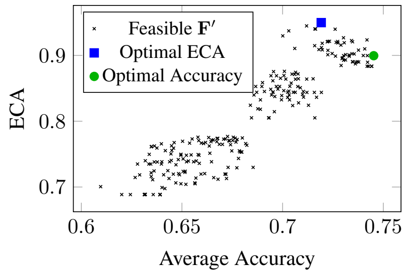
Table 4: Table to calculate the $\text{MAA}(\{F_1, F_2\})$

Table 4 offers a visualization of the algorithm. Here, we wish to compute $\text{MAA}(\{F_1, F_2\})$ with respect to the Bayesian network classifier $\alpha = (C, \{F_1, F_2, F_3\}, 0.55)$ in Figure 3. Each table row corresponds to a feature instance, sorted by the class probability. We consider five different cutoff points, and the ECA value at each cutoff point is the sum of expected probability of positive class for instances above the line and the expected probability of negative class below the line. In this case, $\text{MAA}(\{F_1, F_2\}) = 0.56$ with the optimal threshold $T^\star \in (0, 0.50]$, indicated by a dotted line in the table.

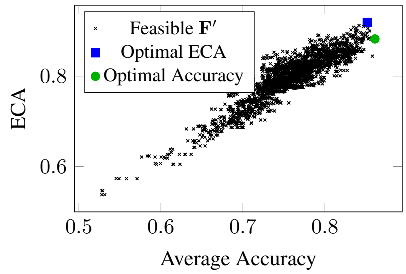
### 5.4 Complexity

Our proposed MPA and MAA algorithms, as described so far, appear to consider all possible feature instances, which is exponential in the size of the feature subset. Nevertheless, this can be improved by exploiting context-specific independence, a property having to do with the parameters of the classifier's probability distribution [Boutilier *et al.*, 1996]. In particular, suppose given a context $\mathbf{g}$ (where $\mathbf{G} \subseteq \mathbf{F}'$), the remaining features $\mathbf{F}' \setminus \mathbf{G}$ become independent of the class variable. Then in our MPA and MAA calculations, we only need to consider the context $\mathbf{g}$ instead of individually considering all instances of $\mathbf{F}'$ that share this partial instantiation. The E-SDP algorithm by Choi *et al.* [2017] that we utilize is based on SDD compilation, which exploits some context-specific independence to simplify the circuit.[3] However, this is currently limited to certain cases, such as when some features have uniform parameters given a context, and does not exploit all simplification opportunities described above. Hence, compilation that fully exploits context-specific independence is an interesting future direction to further improve the efficiency of our algorithms.

[3] We refer to Chavira and Darwiche [2008] and Choi *et al.* [2013] for more details on this simplification.




(a) ECA and average accuracy for `pima`




(b) ECA and average accuracy for `heart`

| | | Agreement | Accuracy |
|---|---|---|---|
| `pima` | Opt. ECA | 0.9863 | 0.7123 |
| | Opt. Acc. | 0.9452 | 0.7260 |
| `heart` | Opt. ECA | 0.9245 | 0.8491 |
| | Opt. Acc. | 0.9057 | 0.7925 |

(c) Test agreement and accuracy

Figure 5: (a),(b) ECA and average accuracy achieved by feasible feature subsets. (c) evaluation of subsets with highest ECA and accuracy.

# 6 Experimental Evaluation

We now empirically evaluate our proposed algorithms on several naive Bayes and general Bayesian network benchmarks.

## 6.1 Accuracy vs. Agreement

We evaluate our method on real-world datasets from the UCI repository [Bache and Lichman, 2013], BFC[4], and CRESST[5]. We randomly split each dataset into 80/20 train and test sets and learn a naive Bayes classifier using the training set. With the budget set as half the number of features and threshold as 0.5, we compute the ECA of each feasible feature subset. In addition, we compute the average classification accuracy of each feature subset using 10-fold cross validation on the training set.

Figure 5 shows the ECA and average accuracy achieved by each feasible subset, and the subsets with highest ECA and highest accuracy are highlighted. We can see that optimizing the ECA tends towards higher accuracy. More interestingly, we can observe a Pareto frontier where one cannot increase the ECA without sacrificing average accuracy, and vice versa. This suggests that one may need to make a tradeoff between classifier agreement (i.e., robustness) and accuracy when selecting features. Moreover, we evaluate the subsets with highest ECA and accuracy on the test set and report their empirical classification agreement and accuracy in Figure 5c. The subset chosen for optimal ECA on the training set also achieves high classification agreement on the test set. Surprisingly, on network `heart`, it also achieves higher test accuracy than the subset with the highest average cross-validation accuracy on the training set. A possible explanation is that choosing subsets based on their cross-validation classification accuracy does not generalize well to the test set. It may introduce additional overfitting that ECA does not suffer from: if the original model generalizes well, we also expect our trimmed classifier to generalize well. We also evaluated accuracy and agreement of feature selection by information gain, but it neither outperformed optimizing the ECA nor the average cross-validation accuracy. In addition, our method achieves higher accuracy on most splits of the `heart` data, which suggests that this may be a property of the dataset. In particular, the average cross-validation accuracy of the original full classifier for `pima` was approximately 0.720, which was lower than the average accuracy of about 21% of the candidate subsets. As our method optimizes for agreement with this original classifier, which has relatively low accuracy, the resulting trimming may have lower accuracy than if we were to actively optimize for average accuracy. On the other hand, the original classifier for `heart` had average accuracy 0.866, which was lower than only 2% of the candidate subsets. Hence, in this case, optimizing the ECA to closely mimic the original classifier's behavior also results in relatively high classification accuracy.

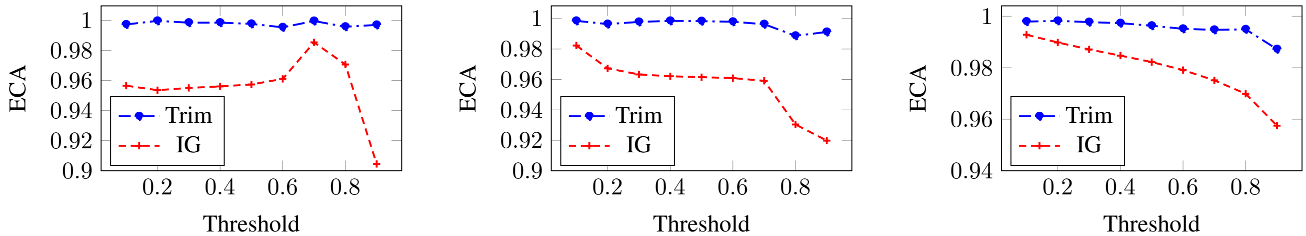
[4]http://www.berkeleyfreeclinic.org/
[5]http://www.cse.ucla.edu/

| | $\lvert\mathbf{F}\rvert$ | FS-SDD Time | FS-SDD # Eval | ECA-TRIM Time | ECA-TRIM # Eval | $\binom{n}{m}$ |
|---|---|---|---|---|---|---|
| `bupa` | 6 | 0.044 | 21 | 0.026 | 14 | 15 |
| `pima` | 8 | 0.056 | 36 | 0.039 | 45 | 28 |
| `ident` | 9 | 0.128 | 129 | 0.097 | 89 | 84 |
| `anatomy` | 12 | 2.252 | 793 | 1.085 | 283 | 495 |
| `heart` | 13 | 7.346 | 1092 | 2.234 | 209 | 715 |
| `voting` | 16 | 819.163 | 6884 | 407.571 | 3345 | 4368 |
| `hepatitis` | 19 | Timeout | 43795 | 4390.71 | 2208 | 27132 |

Table 6: Runtime in seconds and number of criteria evaluations

## 6.2 Runtime

We also compare the efficiency of our algorithm against FS-SDD, which is the feature selection algorithm based on E-SDP [Choi *et al.*, 2017]. Each naive Bayes classifier was trimmed with the budget set to ⅓ the number of features, each feature given unit cost, and classification thresholds in $\{0.1, 0.2, \ldots, 0.9\}$. Table 6 shows the average running times in seconds and the number of times each algorithm computes the criterion value. The performance of our method is comparable to that of FS-SDD on smaller classifiers, and more efficient for larger ones. In particular, our algorithm could handle the largest network with around 4000 ECA computations, whereas an exhaustive search method by FS-SDD requires a significantly greater number of computations and could not be run to completion. Moreover, in 16% of experiment instances described above, the trimmed classifier with optimized threshold reported higher classification similarity (measured by the ECA) than the classifier using features selected by FS-SDD, which keeps the threshold fixed. Note that our method optimizes the threshold of the trimmed classifier and thus is theoretically guaranteed to achieve classification similarity (measured by the ECA) no lower than feature selection us-




(a) `alarm` (b) `win95pts` (c) `mammography`

Figure 7: Comparing ECA of features selected by classifier trimming and information gain

ing E-SDP. Hence, threshold optimization allows us to find more robust trimmings more efficiently. We also show the total number of subsets within budget, $\binom{n}{m}$, to illustrate that branch-and-bound algorithms tend to require fewer number of evaluations than exhaustive search as the number of features increase. This justifies the extra MPA calculations done for pruning on subsets larger than the budget.

### 6.3 Trimming General Networks

Next, we evaluate the quality of trimmed classifiers on general Bayesian networks from the UAI 2008 evaluation and a tree-augmented naive Bayes model for mammography reports [Gimenez *et al.*, 2014]. We run our method with $T$ in $\{0.1, 0.2, \ldots, 0.9\}$ and the budget set to 1/3 the number of features. For the UAI networks, we randomly chose a root node to be the class variable and used the set of all leaf nodes as the feature set $\mathbf{F}$. Each setting was repeated for three randomly selected class variables. For the mammography network, we used the (root) decision node as the class variable and chose 17 out of the 20 variables to be the feature set. Training data was not available for these networks, so we compare against feature selection by information gain instead of classification accuracy. Figure 7 highlights the results. The trimmed classifier by our algorithm consistently achieves higher expected classification agreement, demonstrating that robustness is not easily achieved by other feature selection methods. We also want to stress that the features selected using information gain differ for different class variables, but stay the same across different initial threshold values. On the other hand, our algorithm is sensitive to the original threshold, and thus results in trimmed classifiers with similar behavior as the original classifier with a particular threshold.

## 7 Conclusion

This paper developed a novel operator on Bayesian classifiers: to trim the set of features to fit within a budget, while simultaneously adjusting the classification threshold. Our objective was to optimize the expected classification agreement between the original classifier and its trimmed counterpart. By analyzing the properties of classifier agreement and its maximum potential agreement, we developed a branch-and-bound search algorithm to find optimal trimmings. Experiments on naive and general Bayesian networks demonstrated the effectiveness of our approach in finding robust trimmings of classifiers, especially compared to optimizing more traditional objectives such as expected SDP and information gain.

## Acknowledgments

The authors wish to thank Adnan Darwiche for his contributions to an earlier version of this work, and Arthur Choi for helpful advice and discussions. This work is partially supported by NSF grants #IIS-1657613, #IIS-1633857 and DARPA XAI grant #N66001-17-2-4032.

## References

[Bache and Lichman, 2013] K. Bache and M. Lichman. UCI machine learning repository, 2013.

[Bellazzi and Zupan, 2008] Riccardo Bellazzi and Blaz Zupan. Predictive data mining in clinical medicine: current issues and guidelines. *International journal of medical informatics*, 77(2):81–97, 2008.

[Bilgic and Getoor, 2011] Mustafa Bilgic and Lise Getoor. Value of information lattice: Exploiting probabilistic independence for effective feature subset acquisition. *Journal of Artificial Intelligence Research (JAIR)*, 41:69–95, 2011.

[Boutilier *et al.*, 1996] Craig Boutilier, Nir Friedman, Moises Goldszmidt, and Daphne Koller. Context-specific independence in bayesian networks. In *Proceedings of the Twelfth international conference on Uncertainty in artificial intelligence*, pages 115–123. Morgan Kaufmann Publishers Inc., 1996.

[Chavira and Darwiche, 2008] Mark Chavira and Adnan Darwiche. On probabilistic inference by weighted model counting. *Artificial Intelligence*, 172(6-7):772–799, 2008.

[Chen *et al.*, 2015] Suming Chen, Arthur Choi, and Adnan Darwiche. Value of information based on Decision Robustness. In *Proceedings of the 29th Conference on Artificial Intelligence (AAAI)*, 2015.

[Cheng and Greiner, 1999] Jie Cheng and Russell Greiner. Comparing bayesian network classifiers. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 101–108. Morgan Kaufmann Publishers Inc., 1999.

[Choi *et al.*, 2012] Arthur Choi, Yexiang Xue, and Adnan Darwiche. Same-Decision Probability: A confidence measure for threshold-based decisions. *International Journal of Approximate Reasoning (IJAR)*, 2:1415–1428, 2012.

[Choi *et al.*, 2013] Arthur Choi, Doga Kisa, and Adnan Darwiche. Compiling probabilistic graphical models using sentential decision diagrams. In *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, pages 121–132. Springer, 2013.

[Choi *et al.*, 2017] YooJung Choi, Adnan Darwiche, and Guy Van den Broeck. Optimal feature selection for decision robustness in bayesian networks. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, August 2017.

[Darwiche, 2011] Adnan Darwiche. SDD: A new canonical representation of propositional knowledge bases. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pages 819–826, 2011.

[Friedman *et al.*, 1997] Nir Friedman, Dan Geiger, and Moises Goldszmidt. Bayesian network classifiers. *Machine learning*, 29(2-3):131–163, 1997.

[Gao and Koller, 2011] Tianshi Gao and Daphne Koller. Active classification based on value of classifier. In *Advances in Neural Information Processing Systems*, pages 1062–1070, 2011.

[Gimenez *et al.*, 2014] Francisco J Gimenez, Yirong Wu, Elizabeth S Burnside, and Daniel L Rubin. A novel method to assess incompleteness of mammography reports. In *AMIA Annual Symposium Proceedings*, volume 2014, page 1758. American Medical Informatics Association, 2014.

[Kira and Rendell, 1992] Kenji Kira and Larry A Rendell. The feature selection problem: Traditional methods and a new algorithm. In *AAAI*, volume 2, pages 129–134, 1992.

[Kolesar, 1967] Peter J Kolesar. A branch and bound algorithm for the knapsack problem. *Management science*, 13(9):723–735, 1967.

[Kollar and Roy, 2008] Thomas Kollar and Nicholas Roy. Efficient optimization of information-theoretic exploration in slam. In *AAAI*, volume 8, pages 1369–1375, 2008.

[Kononenko, 2001] Igor Kononenko. Machine learning for medical diagnosis: history, state of the art and perspective. *Artificial Intelligence in medicine*, 23(1):89–109, 2001.

[Krause and Guestrin, 2009] Andreas Krause and Carlos Guestrin. Optimal value of information in graphical models. *Journal of Artificial Intelligence Research (JAIR)*, 35:557–591, 2009.

[Laidlaw *et al.*, 1998] David H Laidlaw, Kurt W Fleischer, and Alan H Barr. Partial-volume bayesian classification of material mixtures in mr volume data using voxel histograms. *IEEE transactions on medical imaging*, 17(1):74–86, 1998.

[Lucas, 2001] Peter Lucas. Expert knowledge and its role in learning bayesian networks in medicine: an appraisal. In *Conference on Artificial Intelligence in Medicine in Europe*, pages 156–166. Springer, 2001.

[Metsis *et al.*, 2006] Vangelis Metsis, Ion Androutsopoulos, and Georgios Paliouras. Spam filtering with naive bayes-which naive bayes? In *CEAS*, volume 17, pages 28–69, 2006.

[Millán and Pérez-De-La-Cruz, 2002] Eva Millán and José Luis Pérez-De-La-Cruz. A Bayesian diagnostic algorithm for student modeling and its evaluation. *User Modeling and User-Adapted Interaction*, 12(2-3):281–330, 2002.

[Munie and Shoham, 2008] Michael Munie and Yoav Shoham. Optimal testing of structured knowledge. In *Proceedings of the 23rd National Conference on Artificial intelligence*, pages 1069–1074, 2008.

[Narendra and Fukunaga, 1977] Patrenahalli M. Narendra and Keinosuke Fukunaga. A branch and bound algorithm for feature subset selection. *IEEE Transactions on Computers*, 26(9):917–922, 1977.

[Wu *et al.*, 2008] Xindong Wu, Vipin Kumar, J Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J McLachlan, Angus Ng, Bing Liu, S Yu Philip, et al. Top 10 algorithms in data mining. *Knowledge and information systems*, 14(1):1–37, 2008.

[Yu and Liu, 2004] Lei Yu and Huan Liu. Efficient feature selection via analysis of relevance and redundancy. *Journal of machine learning research*, 5(Oct):1205–1224, 2004.

[Zhang and Ji, 2010] Yongmian Zhang and Qiang Ji. Efficient sensor selection for active information fusion. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 40(3):719–728, June 2010.

[Zhao *et al.*, 2017] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. In *EMNLP*, 2017.