VOILA: Efficient Feature-value Acquisition for Classification

Mustafa Bilgic and **Lise Getoor**
University of Maryland
College Park, MD
{mbilgic,getoor}@cs.umd.edu

Abstract

We address the problem of efficient feature-value acquisition for classification in domains in which there are varying costs associated with both feature acquisition and misclassification. The objective is to minimize the sum of the information acquisition cost and misclassification cost. Any decision theoretic strategy tackling this problem needs to compute value of information for sets of features. Having calculated this information, different acquisition strategies are possible (acquiring one feature at time, acquiring features in sets, etc.). However, because the value of information calculation for arbitrary subsets of features is computationally intractable, most traditional approaches have been greedy, computing values of features one at a time. We make the problem of value of information calculation tractable in practice by introducing a novel data structure called the Value of Information Lattice (VOILA). VOILA exploits dependencies between missing features and makes sharing of information value computations between different feature subsets possible. To the best of our knowledge, performance differences between greedy acquisition, acquiring features in sets, and a mixed strategy have not been investigated empirically in the past, due to inherit intractability of the problem. With the help of VOILA, we are able to evaluate these strategies on five real world datasets under various cost assumptions. We show that VOILA reduces computation time dramatically. We also show that the mixed strategy outperforms both greedy acquisition and acquisition in sets.

Introduction

Optimal cost-sensitive feature-value acquisition requires a sequential decision making process, in which a decision maker can repeatedly acquire information, and, at some point, makes a decision, based on the acquired information. At each step in the process, the decision maker must find an optimal subset of features for acquisition (a set for which the expected value of information minus the acquisition cost is maximized). As information is acquired, the expected utility of unobserved features sets changes, and needs to be recomputed.

Because of the inherent exponential complexity of exploring all possible feature acquisition strategies, and the expense of computing the value of information calculation for arbitrary sets, the traditional approaches to the problem have made a variety of assumptions. The most common assumption is to use a myopic, or greedy, approach, calculating the value of information for each feature independently and acquiring a single feature at a time (see for example (Gaag & Wessels 1993)). Other approaches approximate the nonmyopic strategy; early examples include (Heckerman, Horvitz, & Middleton 1993); more recent work includes hardness results (Krause & Guestrin 2005b), and more sophisticated approximation algorithms (Krause & Guestrin 2005a).

Within the context of cost-sensitive feature-acquisition for classification and diagnosis, previous research has focused on feature-value acquisition during learning, for specific models such as Naive Bayes (Melville *et al.* 2005), for learning a Markov Decision Process which captures the diagnostic policies (Bayer-Zubek 2004), and analyzing the theoretical aspects of the problem under a PAC learning framework (Greiner, Grove, & Roth 2002). Other approaches examine feature value acquisition during testing for Naive Bayes (Chai *et al.* 2004) and decision trees (Sheng & Ling 2006).

Here, we propose an approach to cost-sensitive feature-value acquisition for classification based on graphical models which enables us to consider nonmyopic feature acquisition strategies. We propose a data structure, the Value of Information Lattice (VOILA), which is a directed graph where each node represents a unique subset of the features and each edge represents a subset relationship between its nodes. As we have discussed above, the number of potential feature subsets is exponential. To reduce the number of subsets in practice, we exploit the dependencies between different features. Specifically, we assume that we have a Bayesian network as the model of the domain. This network lets us ignore sets that have irrelevant features in them. For instance, assume that observing the feature A renders another feature B irrelevant for the purpose of diagnosis. Then, we do not need to compute value of information for the set that contains both A and B because it would be equivalent to the information value of the set that contains only A. By exploiting such relationships, we can reduce the number of subsets dramatically.

VOILA naturally lets us address the problem of feature subset acquisition strategies. We propose a mixed strategy of computation and acquisition. First we find the most beneficial set of features to acquire, and then we acquire one fea-

ture from that set and then repeat. The acquisition made at each step changes the space of relevant subsets and VOILA makes tracking these changes efficient.

The rest of the paper is organized as follows. We begin by defining the problem formally. Next we introduce VOILA and give a construction algorithm. Then we describe feature acquisition strategies and show how the VOILA is maintained. We present experimental results, and then conclude.

Problem Formulation

We assume our classification task is to predict the value for a random variable Y, given the values for some subset of acquired features $\mathbf{X}^a$. We assume that we have a set of n features $\mathbf{X} = \{X_1, \ldots, X_n\}$, and our main task is to find a strategy for acquiring $\mathbf{X}^a \subseteq \mathbf{X}$, given a cost model for misclassification and feature-value acquisition. We assume that a probabilistic model over these random variables has already been learned. For the purposes of this paper, we assume that we are given a Bayesian network over the variables $\mathbf{X}$ and Y; however, any probabilistic model which allows us to efficiently answer context-specific conditional independence queries can be used.

We also assume that we are given a cost model that specifies the cost of feature acquisition and misclassification. Formally, we assume that we have a feature acquisition cost function that given a subset of features, $\mathbf{S}$, and the set of currently acquired features $\mathbf{e}$, returns a non-negative real number, $C_{\mathbf{e}}(\mathbf{S})$, and a misclassification cost model which returns the misclassification cost c_{ij} incurred when Y is assigned y_i when the correct assignment is y_j. Note that we are assuming a very general feature acquisition cost model and not assuming symmetric misclassification costs.

Given the probabilistic model and the cost model, the *expected misclassification cost* given evidence $\mathbf{e}$ is:

$$EMC(Y \mid \mathbf{e}) = \min_{y_i} \sum_{y_j \neq y_i} P(Y = y_j \mid \mathbf{e}) \times c_{ij} \qquad (1)$$

The objective in feature acquisition is to acquire the subset of features which minimizes the sum of the feature-value acquisition cost and the misclassification cost. Formally, we would like to acquire the values for a subset $\mathbf{S}$ of $\mathbf{X} \setminus E$ so that $EMC(Y \mid \mathbf{e}, \mathbf{S}) + C_{\mathbf{e}}(\mathbf{S})$ is minimized. However, because we do not know the values of the variables in $\mathbf{S}$ apriori, we need to average over all possible values. This is in effect computing the expected value of information for the set $\mathbf{S}$:

$$EVI_{\mathbf{e}}(\mathbf{S}) = EMC(Y \mid \mathbf{e}) - (\sum_{\mathbf{s}} P(\mathbf{s} \mid \mathbf{e}) EMC(Y \mid \mathbf{e}, \mathbf{s})) \qquad (2)$$

And, now we can defined a *locally optimal feature* set $\mathbf{S}$:

Definition 1 *A feature acquisition set $\mathbf{S}$ is* **locally optimal** *with respect to evidence $\mathbf{e}$ if it maximizes the difference $EVI_{\mathbf{e}}(\mathbf{S}) - C_{\mathbf{e}}(\mathbf{S})$.*

However, as we have discussed in the previous section, there are an exponential number of subsets $\mathbf{S}$ to consider. Moreover, because of the summation in Eq. (2), the value of information calculation for a set is impractical if the set contains large number of features. Finally, even if we can do all of these computations, as we acquire feature values, the locally optimal set of features changes. To address these problems, we introduce a data structure called the value of information lattice and we describe how it can be used by a mixed feature acquisition strategy that effectively exploits the structure in the probabilistic model and the cost model.

Value of Information Lattice (VOILA)

We propose a data structure that we call the Value of Information Lattice (VOILA). VOILA is a data structure that contains only the potentially relevant feature subsets for acquisition. It allows effective computation sharing and provides an efficient mechanism for incrementally updating the search space and feature cost calculations as new evidence is acquired.

Definition 2 *A set $\mathbf{S} \subseteq \mathbf{X}$ is* **irreducible** *with respect to evidence $\mathbf{e}$ iff $\forall X_i \in \mathbf{S}$, X_i is <u>not</u> conditionally independent of Y given $\mathbf{e}$ and $\mathbf{S} \setminus \{X_i\}$.*

Given a Bayesian network over $\mathbf{X}$ and Y, it is straightforward to check this d-separation property (Pearl 1988).

Proposition 1 *Let $\mathbf{S}'$ be a maximal irreducible subset of $\mathbf{S}$ with respect to $\mathbf{e}$. Then, $EVI_{\mathbf{e}}(\mathbf{S}) = EVI_{\mathbf{e}}(\mathbf{S}')$.*

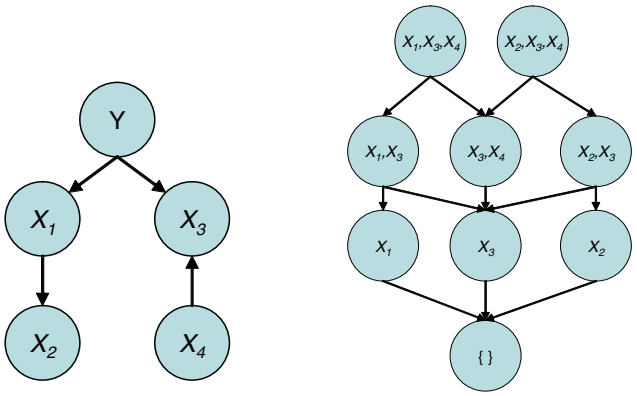
Proof: Let $\mathbf{S}'' = \mathbf{S} \setminus \mathbf{S}'$. If $\mathbf{S}'$ is a maximal irreducible set, $\mathbf{S}' \cup \mathbf{E}$ d-separates Y and $\mathbf{S}''$. Otherwise, we could make $\mathbf{S}'$ larger by including the non-d-separated element(s) from $\mathbf{S}''$ in $\mathbf{S}'$. Thus, we have $P(Y \mid \mathbf{e}, \mathbf{s}) = P(Y \mid \mathbf{e}, \mathbf{S}', \mathbf{S}'') = P(Y \mid \mathbf{e}, \mathbf{S}')$. Substitution in Eq. (2) yields the desired property.

Note that under the assumption that $C_{\mathbf{e}}(\mathbf{S}') \leq C_{\mathbf{e}}(\mathbf{S})$ for any $\mathbf{S}' \subseteq \mathbf{S}$, it suffices to consider only the irreducible sets to find the optimal solution to the feature set acquisition problem. VOILA is a data structure that contains only the irreducible feature subsets of $\mathbf{X}$ (with respect to a particular set of evidence $\mathbf{e}$). We next define VOILA formally.

Definition 3 *A VOILA $\mathcal{V}$ is a directed graph in which there is a node corresponding to each possible irreducible set of features, and there is a directed edge from a feature set $\mathbf{S}$ to each node which corresponds to a direct (maximal) subset of $\mathbf{S}$. Other subset relationships in the lattice are then defined through the directed paths in $\mathcal{V}$.*

One important observation is that if there is a directed path from node $\mathbf{S}$ to $\mathbf{S}'$ in $\mathcal{V}$, then $\mathbf{S} \supset \mathbf{S}'$ and hence $EVI_{\mathbf{e}}(\mathbf{S}) \geq EVI_{\mathbf{e}}(\mathbf{S}')$.

Example 1 *Figure 1(a) shows a simple Bayesian network and its corresponding VOILA, with respect to the empty evidence set, is shown in Figure 1(b). Notice that the VOILA contains only the irreducible subsets given the Bayesian network with no evidence; for instance, the VOILA does not contain sets that include both X_1 and X_2 because X_1 d-separates X_2 from Y. We also observe that the number of irreducible subsets is 9 in contrast to $2^4 = 16$ possible subsets. Moreover, note that the largest subset size is now 3 in contrast to 4. Having smaller feature sets sizes has a dramatic effect on the value of information calculations. In fact, these savings can make optimal feature-value acquisition strategies feasible in practice.*



(a) (b)

Figure 1: A simple Bayesian network illustrating dependencies between attributes and the class variable. (b) The VOILA corresponding to the network.

Algorithm DependencyConstraint(X_i, Y)

Input: X_i, Y
Output: dependency constraint for X_i, denoted $DC(X_i)$

```
1:  DC(X_i) ← false
2:  for each undirected path p_j between X_i and Y do
3:      DC_j(X_i) ← true
4:      for each X_k on the path do
5:          if X_k does not a cause a v-structure then
6:              DC_j(X_i) ← DC_j(X_i) ∧ ¬X_k
7:          else
8:              DC_j(X_i) ← DC_j(X_i) ∧ (X_k ∨ Descendants(X_k))
9:          end if
10:     end for
11:     DC(X_i) ← DC(X_i) ∨ DC_j(X_i)
12: end for
```

Figure 2: Dependency constraint computation for X_i.

VOILA Construction

Efficient construction of VOILA is not a straightforward task. The brute force approach would be to enumerate all possible subsets of $\mathbf{X}$ and for each subset check whether it is irreducible. However, this brute force approach is clearly suboptimal.

Dependency Constraints To tackle this problem, we first define the notion of "dependency constraints." A dependency constraint for a feature X_i is a constraint on evidence sets required for a dependency between X_i and Y to exist. For instance, in our running example, a dependency constraint for X_2 is $\neg X_1$; in other words, in order for X_2 to be relevant, X_1 should not be included in the evidence. Informally, a dependency constraint for a feature X_i requires that all X_j on the path from Y to X_i to be unobserved if X_j is not part of a v-structure; if X_j is part of a v-structure, then either X_j or one of its descendants must be observed (we refer to these latter constraints as *positivity constraints*). The algorithm for dependency constraint computation for features is given in Figure 2. The running time of the algorithm is linear in the number of the edges in the Bayesian network.

A dependency constraint for a set $\mathbf{S}$ specifies the constraints on the evidence set for $\mathbf{S}$ to be irreducible. Irreducibility requires that $\forall X_i \in \mathbf{S}$, dependency flows between X_i and Y, given the rest of the elements in $\mathbf{S}$ and any additional evidence $\mathbf{E}$. Thus, a dependency constraint for $\mathbf{S}$ is the conjunction of the dependency constraints of its members. The irreducibility of $\mathbf{S}$ can be checked by setting the elements of $\mathbf{S}$ and $\mathbf{E}$ to "true" and setting the remaining elements of $\mathbf{X}$ to "false" and evaluating the set's dependency constraint. In our running example, the dependency constraint for the set $\{X_2, X_4\}$ is $\neg X_1 \wedge X_3$. Assuming $\mathbf{E} = \emptyset$, when X_2 and X_4 are set to true and X_1 and X_3 are set to false, this constraint evaluates to false and thus this set is not irreducible. This makes sense because given no evidence, X_4 is independent of Y, so while $\{X_2\}$ is a useful feature set to consider for acquisition, $\{X_2, X_4\}$ is not.

Construction Algorithm We now describe constructing the VOILA using the computed dependency constraints. VOILA construction proceeds in a bottom up fashion, beginning with the lowest level which initially contains only the empty set and constructs new irreducible feature sets by adding features one at a time into the VOILA structure. Figure 3 gives the details of the algorithm. The algorithm keeps track of the irreducible feature sets $\mathbf{IS}$, and the set of potentially irreducible feature sets $\mathbf{PS}$. A feature set is potentially irreducible if it is possible to extend it with additional features so that the set becomes irreducible. Note that this is possible due to the non-monotonic nature of d-separation. The check can be done efficiently by setting members of $\mathbf{S}$ and $\mathbf{E}$ to true, setting literals with positivity constraints that remain to be processed to true, setting the remaining literals to false, and evaluating the set dependency constraint. The difference between checking for potential irreducibility and irreducibility is that we set positively constraint literals that might be added later to true for the potential irreducibility whereas we set them to false to check for irreducibility. When we are done processing feature X_{i_j}, we remove from $\mathbf{PS}$ any potentially irreducible sets that cannot become irreducible because X_{i_j} will not be re-considered. This step improves the running time of the future steps.

Variable Ordering A good ordering processes features with literals with positivity constraints in other features' dependency constraints earlier. That is, for each undirected path from Y to X_i that includes X_j in a v-structure, a good ordering puts X_j earlier in the ordering than everything between X_j and X_i. For instance, in our sample Bayesian network in Figure 1, we should put X_3 earlier than X_4 in the ordering. We refer to an ordering as *perfect* if it satisfies all the positivity constraints. If a perfect ordering is used, VOILA construction algorithm never generates a potentially irreducible set. Unfortunately, it is not always possible to find a perfect ordering. A perfect ordering is not possible when two features have each other as a positivity constraint literal in their dependency constraints. This case occurs only when there is a loop from Y to Y with two or

Algorithm ConstructVOILA($\mathbf{X}$, Y)

Input: Set of features $\mathbf{X}$ and class variable Y.
Output: The VOILA data structure $\mathcal{V}$, given $\mathbf{E}$.

1: Pick an ordering of elements of $\mathbf{X} = X_{i_1}, X_{i_2}, \ldots, X_{i_n}$
2: $\mathbf{IS} \leftarrow \{\emptyset\}$; $\mathbf{PS} \leftarrow \emptyset$
3: **for** j = 1 to n **do**
4: **for** each $\mathbf{S} \in \mathbf{IS} \cup \mathbf{PS}$ **do**
5: $\mathbf{S}' \leftarrow \mathbf{S} \cup X_{i_j}$,
6: $DC(\mathbf{S}') \leftarrow DC(\mathbf{S}) \wedge DC(X_{i_j})$
7: **if** $\mathbf{S}'$ is irreducible **then**
8: $\mathbf{IS} \leftarrow \mathbf{IS} \cup \{\mathbf{S}'\}$
9: Add a node corresponding to $\mathbf{S}'$ to $\mathcal{V}$
10: **else if** $\mathbf{S}'$ is potentially irreducible **then**
11: $\mathbf{PS} \leftarrow \mathbf{PS} \cup \{\mathbf{S}'\}$
12: **end if**
13: **end for**
14: Remove from $\mathbf{PS}$ all sets that are no longer potentially irreducible (because X_{i_j} can no longer be added)
15: **end for**
16: max = size of largest $\mathbf{S}$ in $\mathbf{IS}$; $L_l = \{S \mid S \in \mathbf{IS} \text{ and } |S| = l\}$
17: **for** $l = 0$ to $max - 1$ **do**
18: **for** each $\mathbf{S} \in \mathbf{L}_l$ **do**
19: **for** each $\mathbf{S}' \in \mathbf{L}_{l+1}$ **do**
20: **if** $\mathbf{S} \subset \mathbf{S}'$ **then**
21: Add an edge from $\mathbf{S}'$ to $\mathbf{S}$ to $\mathcal{V}$
22: **end if**
23: **end for**
24: **end for**
25: **end for**

Figure 3: The VOILA construction algorithm.

more v-structures. A perfect ordering was possible in four of the five real world datasets that we used.

Analysis of VOILA Construction Algorithm The construction algorithm puts a node in the VOILA only if the corresponding set is irreducible (lines 7-9). Moreover, by keeping track of potentially irreducible sets (lines 10-12), we generate every possible irreducible set that can be generated. Thus, VOILA contains only and all of the possible irreducible subsets of $\mathbf{X}$.

The worst-case running time of the algorithm is still exponential in the number of initially unobserved features. The running time in practice, though, depends on the structure of the Bayesian network that the VOILA is based upon. We empirically show in the experimental results section that for five real world datasets, the number of irreducible subsets is substantially smaller than the number of possible subsets.

Using VOILA for Feature-value Acquisition

The process of feature-value acquisition is first to find the locally optimal set of features for acquisition, acquire some or all features in that set, and repeat the process depending on what has been observed. To carry out this process, we need to compute expected value of information for the irreducible sets and also process evidence as it is acquired. VOILA makes these calculations efficient in practice by exploiting the structure of the problem space.

Value of Information Calculation VOILA exploits the subset relationship between different feature sets in order to avoid value of information computation for some nodes. First of all, remember that if there is a directed path from node $\mathbf{S}_1$ to $\mathbf{S}_2$ in VOILA, then $EVI_{\mathbf{e}}(\mathbf{S}_1) \geq EVI_{\mathbf{e}}(\mathbf{S}_2)$. Now assume that there is a directed path from $\mathbf{S}_i$ to $\mathbf{S}_j$ and $EVI_{\mathbf{e}}(\mathbf{S}_i) = EVI_{\mathbf{e}}(\mathbf{S}_j)$. Then, all of the nodes on this path will also have the same value of information, thus we do not need to do the computation for those subsets. An algorithm that makes use of this observation is given in Figure 4.

Algorithm ComputeEVI($\mathcal{V}$, $\mathbf{E}$)

Input: VOILA $\mathcal{V}$ and current evidence $\mathbf{E}$
Output: VOILA updated with correct $EVI()$ values.

1: **for** all root node(s) $\mathbf{S}$ not marked unnecessary **do**
2: $value \leftarrow EVI_{\mathbf{e}}(\mathbf{S})$
3: $ub(descendants(\mathbf{S})) \leftarrow value$
4: **end for**
5: **for** all leaf node(s) $\mathbf{S}$ not marked unnecessary **do**
6: $value \leftarrow EVI_{\mathbf{e}}(\mathbf{S})$
7: $lb(ancestors(\mathbf{S})) \leftarrow value$
8: **end for**
9: **for** all node $\mathbf{S}$ not marked unnecessary where $lb(\mathbf{S}) \neq ub(\mathbf{S})$ **do**
10: $value \leftarrow EVI_{\mathbf{e}}(\mathbf{S})$
11: $lb(ancestors(\mathbf{S})) \leftarrow value$
12: $ub(descendants(\mathbf{S})) \leftarrow value$
13: **end for**

Figure 4: Efficient *EVI* computation using VOILA.

In order to share computations between different nodes of the lattice, we keep lower and upper bounds on the expected value of information for a node. We found this strategy surprisingly effective experimentally. The lower bound is determined by the values at the descendants of the node whereas the upper bound is determined by the values of its ancestors. First, we initialize these bounds by computing the value of the information at the boundary of the lattice, i.e. the root node(s) and the leaf node(s) (lines 1-8). Then, we loop over the nodes whose upper bounds and lower bounds are not equal (line 9-13), computing their values and updating the bounds at their ancestors and descendants. The order in which to choose the nodes in line 9 so that the number of sets for which a value is calculated is minimum is still a research question. A possible heuristic is choosing a middle node on a path between two nodes for which the values have already been calculated.

Evidence Integration Once the best candidate feature set is found, the next step is to acquire some or all of the attributes from that set. The VOILA and the costs of some sets of features might change after the acquisition. An algorithm to handle these changes is given in Figure 5.

The acquired evidence might render some of the previously irreducible sets to be reducible now. We find such sets by first finding the features whose dependency constraints evaluate to false with the cumulative evidence (line 2) and then marking all sets that contain these features as unnecessary (line 4). Moreover, some nodes in the lattice will be equivalent with the introduction of new evidence. For example, once we observe X_1 from $\{X_1, X_2, X_3\}$, the node that contains this set will be equivalent to the node that contains

$\{X_2, X_3\}$. We mark the latter node as unnecessary (lines 7-9). Finally, we need to update the costs of sets of features given the new evidence (lines 10-12).

Algorithm IntegrateEvidence(VOILA, X_i, $\mathbf{E}$)

Input: $\mathcal{V}$, X_i, newly acquired evidence, and $\mathbf{E}$, the current evidence
Output: Update $\mathcal{V}$ structure and values
1: **for** $\forall X_j \in \mathbf{X} \setminus \mathbf{E}$ **do**
2: $DC'(X_j) \leftarrow$ evaluate $DC(X_j)$ with $X_i =$ true and $X_k =$ true $\forall X_k \in \mathbf{E}$
3: **if** $DC'(X_j) =$ false **then**
4: Mark all $\mathbf{S} \in \mathcal{V}$ s.t. $X_j \in \mathbf{S}$ as unnecessary
5: **end if**
6: **end for**
7: **for** $\forall \mathbf{S}' \in \mathcal{V}$ such that $\exists \mathbf{S} \in \mathcal{V}$ s.t. $\mathbf{S} = \mathbf{S}' \cup \{X_i\}$ **do**
8: Mark $\mathbf{S}'$ as unnecessary
9: **end for**
10: **for** $\forall \mathbf{S} \in \mathcal{V}$ such that $\mathbf{S}$ is not marked unnecessary **do**
11: Update $C_{\mathbf{e},X_i}(\mathbf{S})$
12: **end for**
13: $\mathbf{E} \leftarrow \mathbf{E} \cup \{X_i\}$

Figure 5: Incremental evidence integration algorithm.

Using VOILA for Different Strategies We now describe different feature-acquisition strategies. The strategies vary depending on their value of information computation (only for a single feature or for sets of features) and their way of acquiring features (a single feature or a set of features). The greedy or myopic strategy computes the expected value of information for one attribute at a time. It first finds the locally optimal feature to acquire. If such a feature exists, then it is acquired and we repeat the process with the remaining features; otherwise stop and make a decision. The greedy strategy does not consider sets, so it does not need the VOILA. However, this strategy is not guaranteed to make the optimal decisions. Because the greedy strategy does both computation and acquisition at the feature level, we refer to it as the Feature-Feature (**FF**) strategy.

Another possible strategy is to find the set of features that is locally optimal and acquire *all* of the features at once and make a decision. This strategy can make use of VOILA for value of information calculations. We initialize VOILA with the appropriate cost values for the sets. Then, we find the locally optimum set by making use of the *ComputeEvi* algorithm described in Figure 4, acquire all of the features from that set (if any) and make a decision. This strategy makes the correct decision on average but it is not guaranteed to make the optimal decision for each instance. Because both computation and acquisition is done in the set level, we refer to it as the Set-Set (**SS**) strategy.

Another possible strategy is a mixed strategy which first finds the locally optimally set, acquires only the locally optimum feature from that set, and repeats the process with the remaining features. The process stops and makes a decision when the locally optimal set is empty. Implementing this strategy with VOILA is straightforward. We first initialize VOILA. Then, we find the locally optimum set using *ComputeEvi* algorithm (Figure 4), find the locally optimum feature from that set by searching the leaves of VOILA and using the previous *EVI* computation, acquire that feature, integrate it as evidence using *IntegrateEvidence* algorithm (Figure 5) and repeat the process. Because this strategy does the computation at the set level but acquisition at the feature level, we refer to it as the Set-Feature (**SF**) strategy.

Experimental Results

We ran experiments testing the effectiveness of VOILA comparing the three different strategies described above. We ran experiments on the same medical datasets used by Turney (Turney 1995). We learned a Bayesian network and constructed the VOILA for each dataset. Table 1 compares various dimensions of the domains: the number of features, number of potential subsets and the number of irreducible subsets given the network. As can be seen, exploiting the independencies in the Bayesian network decreases the number of relevant feature sets dramatically and makes value of information computation for sets feasible in practice for these datasets.

Dataset	# Features	# of Possible Subsets	# of Nodes in VOILA
Bupa	5	32	26
Heart	13	8,192	990
Hepatitis	19	524,288	18,132
Pima	8	256	139
Thyroid	20	1,048,576	28,806

Table 1: Number of features, potential feature subsets, and number of VOILA nodes for the five datasets.

Next, we tested how different feature-acquisition strategies perform on these datasets. Following Turney, we used the same feature costs and varied misclassification costs from \$1 to \$10,000. In order to test how effective the three strategies described above are, we compared them to a strategy that buys all of the information in the Markov blanket for the class regardless of the feature costs and misclassification costs. We refer to this strategy as the Markov Blanket (**MB**) strategy.

We calculated the expected misclassification costs and feature costs for different strategies by first creating the equivalent decision trees and calculating the probabilities and the costs using the Bayesian network and the cost model. We present results showing how well greedy (**FF**), best subset (**SS**), and the mixed (**SF**) strategies perform in comparison to **MB** strategy (Figure 6). We show only results up to \$5,000 because the trends generally stabilize after this point.

One immediate observation is that when the misclassification cost is small relative to feature costs, all three strategies incur a lower cost than the **MB** strategy simply by acquiring a small number inexpensive features. Secondly, the **FF** strategy initially performs better but after some point gets stuck in local minimum and no matter what the misclassification cost is, it refuses to acquire additional features. Therefore, its savings become negative for high misclassification costs and in this case, the **MB** strategy is a better strategy than **FF**. However, neither the **SS** nor the **SF** strategy suffers from lo-

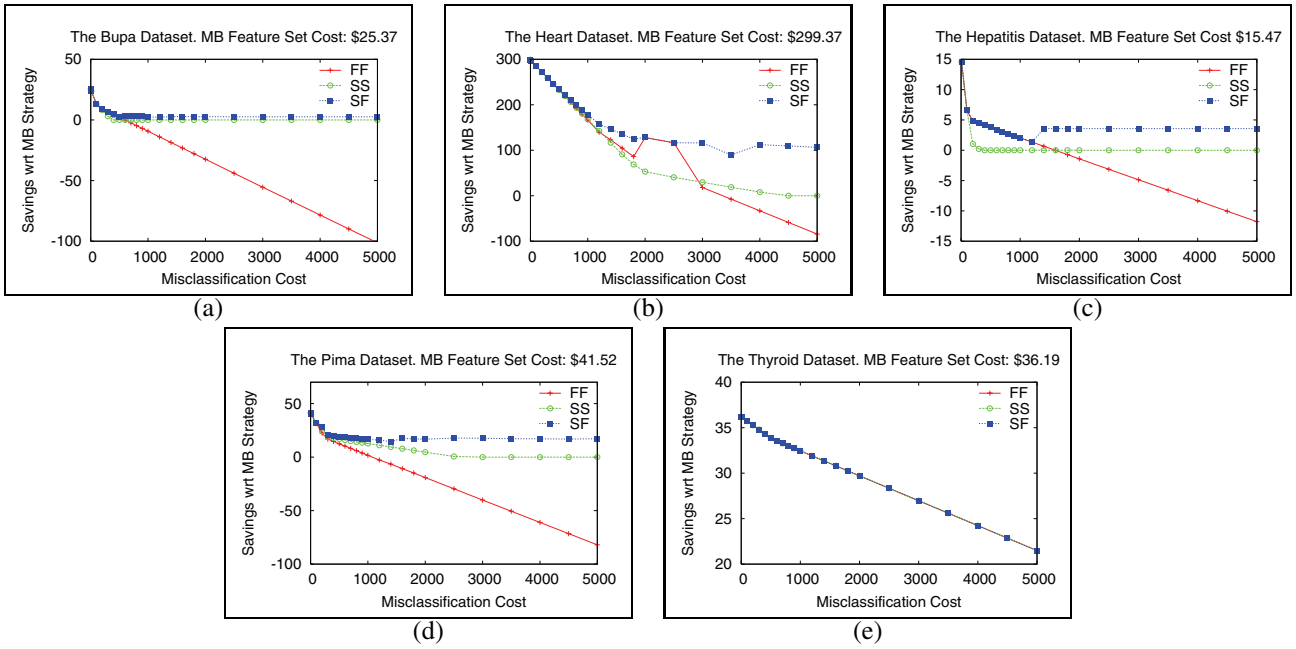


Figure 6: Total savings achieved by the greedy (**FF**), best subset (**SS**), and the mixed (**SF**) strategies when their costs are compared to the costs incurred by the Markov blanket (**MB**) strategy. **FF** usually gets stuck in local minimum and does in fact perform worse than **MB** when the misclassification costs are high compared to feature costs. Both **SS** and **SF** always do better than **MB**, and **SF** usually outperforms **SS**.

cal minimum; that is why they always have non-negative savings. Finally, the **SF** strategy outperforms the **SS** strategy in four of the five datasets and performs equally well in the fifth; for example in the heart dataset **SF** saves on average $100 per instance over **SS** for higher misclassification costs. The reason is that **SF** can change its mind about the usefulness of a set as it acquires more evidence.

Conclusions

We have introduced a novel data structure called the Value of Information Lattice. VOILA exploits dependencies between missing features and makes sharing of information value computations between different feature subsets possible. VOILA allows us to compare different feature subset value of information calculations and feature subset acquisition strategies. We evaluate these strategies on five real world datasets under various cost assumptions and show that we are able to reduces computation time dramatically, and achieve dramatic cost improvements using a mixed computation and feature acquisition strategy.

Acknowledgments: This work was supported by NSF Grant #0423845.

References

Bayer-Zubek, V. 2004. Learning diagnostic policies from examples by systematic search. In *Proc. of UAI*.

Chai, X.; Deng, L.; Yang, Q.; and Ling, C. X. 2004. Test-cost sensitive naive bayes classification. In *Proc. of Int. Conf. on Data Mining*.

Gaag, L. van der, and Wessels, M. 1993. Selective evidence gathering for diagnostic belief networks. *AISB Quarterly* (86):23–34.

Greiner, R.; Grove, A. J.; and Roth, D. 2002. Learning cost-sensitive active classifiers. *Artificial Intelligence* 139(2):137–174.

Heckerman, D.; Horvitz, E.; and Middleton, B. 1993. An approximate nonmyopic computation for value of information. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15(3):292–298.

Krause, A., and Guestrin, C. 2005a. Near-optimal nonmyopic value of information in graphical models. In *Proc. of UAI*.

Krause, A., and Guestrin, C. 2005b. Optimal nonmyopic value of information in graphical models - efficient algorithms and theoretical limits. In *Int. Joint Conf. on AI*.

Melville, P.; Provost, F.; Saar-Tsechansky, M.; and Mooney, R. 2005. Economical active feature-value acquisition through expected utility estimation. In *Proc. of the KDD Workshop on Utility-based Data Mining*.

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems*. San Francisco: Morgan Kaufmann.

Sheng, V. S., and Ling, C. X. 2006. Feature value acquisition in testing: a sequential batch test algorithm. In *Proc. of Int. Conf. on Machine Learning*.

Turney, P. D. 1995. Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm. *Journal of Artificial Intelligence Research* 2:369–409.