An Adaptive Evidence Structure for Bayesian Recognition of 3D Objects

Ahmed M. Naguib
Intelligent Systems Research Institute
Sungkyunkwan University
Suwon, Rep. of Korea
+8231-299-6471
ahmed.m.naguib@gmail.com

Sukhan Lee
Intelligent Systems Research Institute
Department of Interaction Science
Sungkyunkwan University
Suwon, Rep. of Korea
+8231-299-6470
lsh@ece.skku.ac.kr

ABSTRACT
Classification of an object under various environmental conditions is a challenge for developing a reliable service robot. In this work, we show problems of using simple Naïve Bayesian classifier and propose a Tree-Augmented Naïve (TAN) Bayesian Network – based classifier. We separate feature space into binary TRUE/FALSE regions which allows us to drive Bayesian inference prior conditional probabilities from statistical database. We go further using TRUE/FALSE regions to estimate expected posterior probabilities of each object under online specific conditions. These expectations are then used to select optimal feature sets under this environment and autonomously reconstruct Bayesian Network. Experimental results, validation and comparison show the performance of the proposed system.

Categories and Subject Descriptors
I.5.2 [PATTERN RECOGNITION]: Design Methodology – Classifier design and evaluation, Feature evaluation and selection

General Terms
Algorithms, Design, Reliability.

Keywords
3D Object Recognition System, Bayesian Network Restructuring, Optimal Feature set Selection, Environmental Adaptation.

1. INTRODUCTION
A humanoid robot is expected to survive in a typical human environment and perform various types of tasks that are intuitively easy for us as humans. While some applications, such as industrial robotics, permit to design a robot and explicitly program it to perform a specific task under specific predefined/controllable conditions; performance of such a robot drops dramatically with any minor unpredictable variation in the environment. A humanoid robot is expected to cope with far more major continuous variations in its workspace. For such application, regardless of their robustness, open-loop vision systems will simply fail due to their inflexibility. A cognitive vision system is required in which multiple features are merged and learning algorithms close the loop and allow primitive intelligence to emerge.

2. PROBLEM STATEMENT AND RELATED WORK
Classification is a process with two inputs: prior objects’ models and current scene measurement. Any variations or noise in any of these two inputs will deteriorate classification accuracy. Issues related to prior object’s models, such as under-sampled distributions, conditions in which samples are collected, untrained background objects, approximation of likelihoods and model representation can cause misclassifications. Similarly, Measurements far from expected models can cause misclassifications too. Since a robot is expected to operate in an uncontrollable environment, changes of this environment can drive feature measurement far from the trained model. Generally, causes of classification errors can be categorized into the following table:

<table>
<thead>
<tr>
<th>Issues</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Features</td>
<td>- Sensor Capability</td>
</tr>
<tr>
<td>Uncertainties</td>
<td>- Feature Extraction Algorithm</td>
</tr>
<tr>
<td></td>
<td>- Occlusion</td>
</tr>
<tr>
<td></td>
<td>- Distance, Intensity, Orientation</td>
</tr>
<tr>
<td></td>
<td>- Wrong Segmentation</td>
</tr>
<tr>
<td>Classification using Irrelevant Feature(s)</td>
<td></td>
</tr>
<tr>
<td>Approximation Error</td>
<td>- Undersampled Training</td>
</tr>
<tr>
<td></td>
<td>- Low Populated Regions of Likelihood</td>
</tr>
<tr>
<td></td>
<td>- Inaccurate statistical representations of Model Likelihoods</td>
</tr>
</tbody>
</table>

Charu C. Aggarwal [1] Research work assumes a measurement noise that follows a random pattern and tries to recover the measurement of training samples assuming they are inherently sparse in feature space. Other work such as [2], [3], [4] address the classification problem under training sample uncertainty. Online random noise in measurement, however, has been addressed very little in literatures in the scope of classification. We believe that sensor error can be modeled through a sensor supervised calibration and can be used to recover the measurement in both offline training and online acquisition. Problem of occlusion has been repeatedly addressed in literature as well. It is usually addressed from two points of view: 1) Active searching approach that moves sensor to a new occlusion-free perspective, such as the work presented by Xing Chen and James Davis [5] and Xi Chen and Sukhan Lee [6]. 2) Passive approach that measure occlusion and prevent recognition from taking the wrong decision. For the scope of this work, we will only focus on the passive approach. Michael Boshra and Bir Bhanu [7] presented an approach that predict object recognition...
performance under occlusion. They, however, assumed that feature subset of a model is uniformly distributed along object surface and, thus, are equally likely to be occluded given that they have the same size. This may not be a good assumption since segmentation may dictate a non-uniform distribution of surfaces that can be measured with acceptable certainty. Thus, it is not appropriate to assume occlusion is linearly proportional with amount of distortion in measured features. The relationship depends on the location of occlusion, location of features, and their spatial sizes. Assuming that occlusion likelihood is uniform along the surface is not true too since it ignores occlusion common structures. For example, an occlusion is mostly casted by another object placed on the same ground. Thus, occlusion is highly likely to occur on lower parts of an object surface. David Meger et al [8] presented an approach that localize occlusion along object surface. Features associated with affected regions are denied contribution to naïve Bayesian classifier if occlusion rate is above a hand-chosen threshold. This is a preventive approach that discard contaminated features. Edward Hsiao and Martial Hebert [9] extended LINE2D by introducing visibility penalty corresponding to occlusion conditional likelihood that is computed from a prior occlusion model and detected occlusion regions from scene. Even though they approximated the global relationship between visibility states on an object, the resulting occlusion conditional likelihood is very reasonable. They have compared two schemes of penalties and showed that it improved LINE2D by 3-5%. Their occluding object model is, however, approximated with a surrounding 3D box that causes miss modeling and over penalization in actual scene with freeform objects. We have proposed a novel approach for a cognitive recognition under unstructured severe occlusion in [10].

Variations due to differences in distance/scale, orientation and illumination are addressed in literatures either individually [11], [12], [13], or together [14]. While algorithms relying on machine learning and similarity measurements may produce reasonable results, we believe that an explicit model of effect of such variations on each feature can guarantee performance and avoid unsupervised machine learning drifts and artifacts. Thus, we adapted the approach of W. Jeong [15].

Identifying irrelevant features and discarding them is addressed in researches either implicitly through dimensionality reduction or explicitly through feature selection. Hyunsoo Kim et al [16] compared truncated singular value decomposition to clusters-aware centroid-based algorithm and generalized linear discriminant analysis for the classification performance of an SVM classifier. What is interesting is that even though the purpose of the study is dimensionality reduction for enhancing classification performance of high dimensional feature space, the results show that there are cases where accuracy can be degraded by dimensionality reduction. Authors concluded that these cases require a nonlinear dimensionality reduction approach. Patricia E.N. Lutu [17] used symmetrical uncertainty coefficient (function of entropy) to measure correlation between features and class correlation and continuously select relevant unique feature set from a sliding window method of stream mining. The result shows improvement in predictive performance of naïve Bayesian classifier. A more sophisticated approach, such as [18], may weight features according to their relevance instead of discarding irrelevant features completely.

Database may also cause classification issues when number of training samples are not enough to build a good classifier. This typically occurs when dealing with a very high dimensional feature space. An assumption of independencies may relax the problem and result in appropriate classification. Yuguang Huang and Lei Li [19] used naïve Bayesian classifier with Poisson likelihood distribution. Since naïve Bayesian assumes independency of each feature, it performs expectedly well with under sampled data. Naïve Bayesian classifier overall performance, however, can only be as good as the worst feature.

Low populated likelihood regions issue results from the fact that a feature value usually will be contained within small sparse regions of feature space. No matter how many samples we collect, densities of likelihood far from these regions will always be under-sampled. Decisions taken far from these regions will have a high error since the likelihood ratio shown in equation (1) will be at the edge of singularity as shown by the false 100% decision in the left of figure 1. Feature smoothing, such as Laplace, is required to overcome this issue. Laplace smoothing introduces new virtual samples to the system to accommodate for under sampled regions and overcome singularities. This is simply accomplished by assuming that every possible outcome of an event will happen at least k number of times. This will lead to the elevation of likelihood distributions by a uniform distribution with an amplitude (e) proportional to selected k. since likelihoods will never be close to zero, singularity issue can be evaded which will lead to a more appropriate behavior around decision boundaries as well as on regions far from object’s true values as shown in [20] and [21].

Figure 1. Classification decision: no smoothing (left), a 0.01 Laplace smoothing (right)

Figure 2. Classification decision under multiple Laplace smoothing

The main drawback of this solution is that it is essentially an approximation with a domain-specific control parameter (e). Figure 2 illustrate the decision characteristics under various Laplace smoothing parameter. It’s worth noting that a very small smoothing tolerate a false decision at regions far from true values, while large smoothing may saturate weak likelihood preventing true decisions. Statistical representation of measured samples also inherits an approximation error. A polynomial of high order will probably over fit the likelihood, while an unsupervised Gaussian fitting may result
3. PROBLEMS WITH TRADITIONAL NAÏVE BAYESIAN CLASSIFIER

Naïve Bayesian classifier is classifier that assumes independency between all input features [23]. There have been two main formulation in the literature for naïve Bayesian classifier. Let’s assume \( o_i \) is object of interest, \( f_1 \sim f_n \) are measured features, \( P(f_j | O = o_j) \) is conditional likelihood of feature \( j \) given that it was measured from object \( o_j \), and \( P(O = o_1 | f_1 \sim f_n) \) is posterior probability we are interested in computing. Using Bayesian theorem and the assumption of independency we can write:

\[
P(O = o_1 | f_1 \sim f_n) = \frac{P(O = o_1)P(f_1 \sim f_n | O = o_1)}{P(f_1 \sim f_n)}
\]

\[
= \frac{P(O = o_1)\prod_{j=1-n}P(f_j | O = o_j)}{P(f_1 \sim f_n)}
\]

The first formulation commonly used assumes that denominator is irrelevant since it has nothing to do with the measured object. So, classification can be done as follows:

\[
P(O = o_1 | f_1 \sim f_n) \propto P(O = o_1) \prod_{j=1-n}P(f_j | O = o_j)
\]

Thus, object is classified by finding

\[
\arg\max_{o_1} \left\{ P(O = o_1) \prod_{j=1-n}P(f_j | O = o_j) \right\}
\]

This is also known as maximum a posteriori likelihood estimate (MAP). The assumption that features’ probabilities have nothing to do with objects is not accurate. There is another way to drive a value of the denominator from likelihoods and prior probabilities using total probability theorem as follows:

\[
P(f_1 \sim f_n) = P(f_1 \sim f_n | o_1)P(O = o_1) + \ldots + P(f_1 \sim f_n | o_k)P(O = o_k)
\]

\[
= \sum_{k=1}^{K} P(f_1 \sim f_n | o_k)P(O = o_k)
\]

For conservatism, let’s assume worst case. All prior probability are zero except for the target object and another object that happens to have likelihoods that best fit the measurement:

\[
P(f_1 \sim f_n, \text{worst}) = P(f_1 \sim f_n | o_1)P(O = o_1) + \arg\max_{o_k}(P(f_1 \sim f_n | o_k)P(O = o_k))
\]

\[
\text{thus, } P(O = o_1 | f_1 \sim f_n, \text{worst}) = \frac{P(f_1 \sim f_n | o_1)P(O = o_1) + \arg\max_{o_k}(P(f_1 \sim f_n | o_k)P(O = o_k))}{1 + \arg\max_{o_k}(P(f_1 \sim f_n | o_k)P(O = o_k))}
\]

By adding laplace smoothing, assuming independencies and no prior knowledge \( P(O = o_j) = 1 - P(O \neq o_j) = 0.5 \)

\[
P(O = o_1 | f_1 \sim f_n) = \frac{1}{1 + \arg\max_{o_k}(\prod_{j=1-n}P(f_j | O = o_j) + \epsilon)}
\]

The above equation is commonly used as one of the best interpretation of naïve Bayesian classifier. It is a powerful formula that can produce very reasonable results as shown in [24]. It has, however, many drawback such as:

- Feature smoothing is a necessity
- Cascaded multiplication of likelihoods may not be tractable computationally without hitting zero. (log cannot be used when computing posterior probabilities)
- Not scalable, performance drop with the increase of number of features. It accumulates their uncertainties and becomes computationally impossible to keep track of cascaded multiplication of likelihoods
- Very sensitive to irrelevant features
- Ignores conditional relationship of individual features with measured objects. Once a feature likelihood hits zero, no matter how uncorrelated it is to measured object, it dictates every other feature no matter how strongly correlated they may be.

That been said, most of these weaknesses can be evaded by simply applying a feature selection stage before applying naïve Bayesian. Many researches, for efficient feature selection, have been conducted in pair with naïve Bayesian. These approaches, however, still ignore the risk of having a poor feature dictating the decision. For that purpose, researchers started migrating ideas from Bayesian network and tree-augmented naïve Bayesian network is widely adapted as the natural extension of naïve Bayesian classifier [25]

4. PROPOSED ADAPTIVE BAYESIAN NETWORK FRAMEWORK

4.1 Bayesian Network Inferences Model

In a Bayesian Network, a probability can be inferred in any direction once a node is instantiated [26]. In a classification problem, we are interested in backward (diagnostic) inference. We may drive backward inference of a Bayesian Network using total probability theorem as follows:

\[
P(O = o_1) = P(O = o_1 | f_1, f_2, f_3, f_4)P(f_1, f_2, f_3, f_4)
\]

\[
+ P(O = o_2 | f_1, f_2, f_3, f_4)P(f_1, f_2, f_3, f_4)
\]

\[
+ \ldots
\]

\[
+ P(O = o_k | f_1, f_2, f_3, f_4)P(f_1, f_2, f_3, f_4)
\]

The above equation consists of two terms: probability of a feature measurement to be true or false, which can be interpreted as a likelihood, and a prior knowledge of the causal effect of an object on the feature set, represented by the terms:

\[
P(O = o_1 | f_1, f_2, f_3, f_4)P(0 = o_2 | f_1, f_2, f_3, f_4) \ldots P(O = o_k | f_1, f_2, f_3, f_4)
\]

These prior knowledge can be computed using the same formulation of naïve Bayesian classifier (equation 1). The main difference between these prior knowledge and the posterior probabilities computed by traditional naïve Bayesian classifier...
(despite their similar formulas) is in the way likelihood probabilities are obtained. In a traditional naïve Bayesian classifier, likelihood probabilities are computed for an actual measurement from the scene. For our prior knowledge, however, we don’t know what the measurement will turn out to be yet. So, likelihood probability is computed according to our database and expectation of what these measurement should look like under current environment. This will be discussed in detail after the introduction of TRUE region (section 4.3). A Bayesian conditional probability table is formed from these prior knowledge. This table represent our belief of the cause-effect relationship between an object and the set of measurable features used to recognize it.

Table 2. Bayesian conditional probability table

<table>
<thead>
<tr>
<th>h₁</th>
<th>h₂</th>
<th>h₃</th>
<th>O₁</th>
<th>O₂</th>
<th>Oₙ</th>
</tr>
</thead>
<tbody>
<tr>
<td>False</td>
<td>False</td>
<td>P(O = o₁₁₁₁₁₁)</td>
<td>P(O = o₂₁₁₁₁₁)</td>
<td>P(O = oₙ₁₁₁₁₁)</td>
<td></td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>P(O = o₁₁₁₁₁₁)</td>
<td>P(O = o₂₁₁₁₁₁)</td>
<td>P(O = oₙ₁₁₁₁₁)</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>True</td>
<td>True</td>
<td>P(O = o₁₁₁₁₁₁)</td>
<td>P(O = o₂₁₁₁₁₁)</td>
<td>P(O = oₙ₁₁₁₁₁)</td>
<td></td>
</tr>
</tbody>
</table>

Since Network graph along with this table represent explicitly our knowledge of an object, they may be defined manually without any loss of generality. However, for the sake of robustness, we will present an approach to drive the values of this table from statistical likelihood representations of the features and will use this to update the graph autonomously during online acquisition.

While this inference model is general, we would call it an “AND” inference. We will introduce another way to infer independent features using “OR” inference. It is intuitive for human brain to think of evidences in terms of AND/OR relationships. While a group of evidences may infer the existence of an object, they are as good as their weakest evidence. Thus, a more reliable model would be to incorporate a set of various evidence groups. If a bad evidence corrupted a decision of one group, other groups may still survive and dictate the correct final decision. We will refer to each group as a “Sufficient Condition” that is composed from a set of features in a typical Bayesian network with a final probability inference of AND. An object classification may result from multiple of these sufficient conditions. Thus, we combine the results of these branches using OR inference and we refer to this framework as “Evidences Structure” (figure 3). In this paper, we will assume complete indecencies of features among each sufficient condition for simplicity. The way we define “OR” inference is by a weighted voting scheme. If an object exist in a scene, branches of sufficient conditions should infer high probability if their reliability is high. Thus by allowing a weighted voting scheme, we can simply avoid noisy or poor feature measurement. This is simply done as follows:

\[
P(O = o₁)_{set1} OR P(O = o₂)_{set2} = P(O = A[f₁ \in set1]) + P(O = A[f₂ \in set2])
\]

Where \( P(O = A[f₁ \in set1]) \) and \( P(O = A[f₂ \in set2]) \) represents reliability of target object to be detected using sufficient condition of feature set 2. This reliability is updated online according to environmental changes as will be shown in section 4.5.

4.2 TRUE/FALSE Likelihood Regions

One major problem in using a knowledge-based model, such as Bayesian network, is the way its parameters are obtained. In the early work on Bayesian network classifier, this issue was overlooked and heuristic offline coefficients were given to the network resulting in poor inferences. Since we introduced various ways to adapt and update the conditional likelihood distributions in [6], [7], [15], [27], we can estimate the expected likelihood distribution of a feature given an object. We would like to use this knowledge to estimate the expected posterior probabilities, use them to select optimal features sets, restructure the Bayesian network, drive the inference coefficients of the Bayesian network and compute the reliabilities of each sufficient condition for OR inferences. In order to do that, we define TRUE/FALSE regions as labels used to divide feature space into fixed binary tuples. Using this notion, we can estimate the effect of a feature on decision. The more likelihood of an object within TRUE region with respect to other objects, the more this feature is effective to identify that object, and vice versa. In order to keep the consistency of our definitions, we used this notion to measure features’ evidence probabilities given a particular measurement. This is done through a sigmoid natural distribution within fixed TRUE regions. There can be many ways to obtain boundaries of TRUE/FALSE regions. Since they represent ground truth of a measurement, ideally, they should be defined by the manufacturer of the object of interest. For example, when a factory defines the thickness of a smartphone as 9mm with tolerance of 1mm, TRUE region can be defined from 8-10mm and FALSE region is everywhere else. In our work, however, we use statistical representation of feature likelihood gathered in the database to fix TRUE/FALSE regions.

4.3 Estimation of Bayesian Conditional Probabilities Table

Now, since we have introduced TRUE region, we may propose a statistical method for obtaining Bayesian table coefficients. To do that, we may use the same derivation of naïve Bayesian classifier. However, there’s a small important difference. In naïve classifier, the likelihoods probability \( P(f₁∥O = o₁) \) is computed according to features’ evidence measurement. In here, this probability should only be driven from prior information. And since we have defined TRUE and FALSE regions, we may redefine it as \( P(f₁ ∈ TRUE∥O = o₁) \):

\[
P(f₁ ∈ TRUE∥O = o₁) = \int_{TRUE} P(f₁∥O = o₁) = 1 - \int_{FALSE} P(f₁∥O = o₁)
\]
Where $P(f_j|O = o_i)$ represents our updated feature conditional likelihood density distribution. Equation 3 can be used here with little modification as follows:

$$P(O = o_i|f_j, \tilde{O})_{\text{worst}}$$

$$= \frac{1}{1 + \frac{\varepsilon + \arg \max \{(\Pi_j \text{f}_{\text{TRUE}} P(f_j|O = o_i) | (\Pi_k \text{f}_{\text{FALSE}} P(f_k|O = o_i))\}}{\varepsilon + (\Pi_j \text{f}_{\text{TRUE}} P(f_j|O = o_i)) (\Pi_k \text{f}_{\text{FALSE}} P(f_k|O = o_i))}}$$

Computationally, we use Abramowitz and Stegun approximation of error function to compute the integration of likelihood pdf. Figure 4 illustrate the idea of exploiting pre-defined TRUE/FALSE regions to drive Bayesian network coefficients statistically, which is a major contribution to this work.

![Figure 4](image)

**Figure 4. Illustration of using TRUE region to determine feature’s strength**

### 4.4 Estimation of Expected Posterior Probabilities and Features Discrimination Strengths Table

In section 4.3 we compute a conditional probability used in Bayesian network inference. This coefficient may also be interpreted as an estimation of the expected posterior probability of finding the object under the worst case, given a set of successful and failure feature set. By worst case, we mean that another conditional term is assumed, that is, the negative object is the closest object in feature description to the measurement. Hence the argmax operator in the formula. In this section, we are not only interested in the worst case, but also in every possible case. And we would like to study the effect of each feature separately. Thus, we would like to compute:

$$P(O = o_i|f_j, \tilde{O} = o_r) = \frac{1}{1 + \frac{\varepsilon + \int_{\text{TRUE}} P(f_j|O = o_r)}{\varepsilon + \int_{\text{TRUE}} P(f_j|O = o_i)}}$$

That is, the estimated expectation of posterior probability of finding object $o_i$ given that we are only using feature $f_j$, measurement of feature $f_j$ will come true, and that the only other object we are comparing with is $o_r$. Now, by computing this for every possible case, we can form the following expected posterior probabilities, which we can also be interpret as estimated features discrimination strengths, for target object:

**Table 3. Estimated Features Discrimination strengths for Target Object $o_1$**

<table>
<thead>
<tr>
<th>Features</th>
<th>$a_1$</th>
<th>$a_2$</th>
<th>...</th>
<th>$a_n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_1$</td>
<td>$P(0 = a_1</td>
<td>f_1, 0 = o_1)$</td>
<td>$P(0 = a_1</td>
<td>f_1, 0 = a_2)$</td>
</tr>
<tr>
<td>$f_2$</td>
<td>$P(0 = a_1</td>
<td>f_2, 0 = o_1)$</td>
<td>$P(0 = a_1</td>
<td>f_2, 0 = a_2)$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
<td>...</td>
</tr>
<tr>
<td>$f_n$</td>
<td>$P(0 = a_1</td>
<td>f_n, 0 = o_1)$</td>
<td>$P(0 = a_1</td>
<td>f_n, 0 = a_2)$</td>
</tr>
</tbody>
</table>

### 4.5 Estimation of Sufficient Condition Reliability

Using the above table we can estimate the reliability of a single feature using total probability theorem as follows:

$$P(O = A|f_j) = \sum_{i=1-N}^N P(O = A|f_j, \tilde{O} = o_i)P(\tilde{O} = o_i)$$

Now, let’s assume a feature set is determined. We are interested in finding this feature set reliability. Using Naive Bayesian formulation discussed before:

$$P(O = A|f_1, f_2, f_3, ..., f_n) = \frac{1}{1 + \prod_{i=1-n}^n P(f_i|O \neq A) - 1}$$

Now single feature reliability can be formulated using naive Bayesian too as follows:

$$P(O = A|f_j) = \frac{1}{1 + \frac{P(f_j|O \neq A)}{P(f_j|O = A)}}$$

$$P(f_j|O \neq A) = \frac{1}{P(O = A|f_j)} - 1$$

So, we can estimate the reliability of the feature set as follows:

$$P(O = A|f_1, f_2, f_3, ..., f_n) = \frac{1}{1 + \prod_{i=1-n}^n \left(\frac{1}{P(O = A|f_j)} - 1\right)}$$

### 4.6 Optimal Features Sets Selection and Evidence Structure Update

By determining a minimum required discrimination strength between target object and other objects, we use an iterative algorithm to keep accumulating next-best-feature in optimal feature-set bag until they mutually meet the required discrimination criteria. This algorithm can find the optimal feature (if exist) set for discriminating a target object given current environmental variations. We, however, are interested in finding multiple feature sets to form an evidence structure of multiple sufficient conditions. So, after finding the optimal feature set, we deliberately reduce discrimination strength of each feature and rerun the algorithm to...
find second, third, … best optimal features sets as shown in figure 5. The overall Tree Augmented Naïve Bayesian Network is formed by simply putting selected ANDs through an OR to take a decision.

Figure 5. Flowchart for selection of multiple sufficient conditions

Finally, the above system has been implemented with the following overall flowchart:

Figure 6. Overall system flowchart

5. EVALUATION AND EXPERIMENTAL RESULTS

We have trained our system on 10 different objects, each with multiple alignments (pitch/yaw), each with 16 orientation (roll) as shown in figure 7. For the sake of comparison, we have developed an adaptive Naïve Bayesian classifier [27]. An object (milk box) is chosen as a target object. Experimental environment consists of a table with a sliding platform that allows an object to slide from 50cm to 3m distance away from sensor. An Asus xtion pro RGBD sensor, and a Bumblebee2 stereo camera are used for data acquisition. For the sake of illumination experiments, two levels (bright: lights on, dark: lights off) were used. Two experimental results will be presented: 1) Qualitative: will show how proposed system behaves under various conditions. 2) Statistical evaluation and comparison with adaptive Naïve Bayesian classifier.

5.1 Experiments for Adaptability

At first, target object (milk box) is placed in front of the camera [28] and figure 8. Recognition system used octree for segmentation and found out there’s a candidate object at a distance of 73cm that is about 32% bright and has no occlusion. System updates the likelihoods distributions of every feature of every object in the database, then system construct discrimination strength table and decided that at these conditions, 80% chances of discrimination can be achieved. It then picks up optimal feature set (height, middle width, and SIFT) and constructed their conditional probability table. System also computes the reliability associated with this sufficient condition (99.7%) thus, final decision has inherited only 0.3% uncertainty. Finally, the system went ahead and measured these 3 features and computed their probability to match the target object and inferred their likelihoods throughout the evidence structure and concluded that this candidate is indeed the target object with 97%. These steps took the system 187ms to compute. 100 cycles were computed and shows that mean probability is 95% with standard deviation of 3%. The target object is then gradually moved away from the sensor and continuous readings were made along the way as shown in figure 9. System detects the distance changes every frame and updates the likelihoods distributions accordingly. System also reduces chances of discriminations, allowing itself to pick up 5 different sufficient conditions. It’s worth noticing that even when features measurement fails, final decision probability is about 50% with very high uncertainty. There are two
more things to notice in figure 9: 1) In “Interpretation Summary” chart, probabilities of this candidate being other object from database (not milkbox) are increased to about 50% since system is not really certain what this object is at this distance. 2) “history” chart shows the last 200 cycles’ probabilities. It shows the decline of probability along with distance. As expected, the decline shows a characteristic of an exponential function responding the exponential modeling of the distance variations used to update features likelihoods distributions. On the other hand, Naïve Bayesian classifier could not adapt to this extreme condition. Measurement of top width, top shape, and middle width were contaminated. The resulting negative to positive likelihood ratio exceeded the classification boundary and the object was falsely classified as a yellow cup. The opposite is also shown in figure 10. Another non-target object (Pringles) is placed at far distance and gradually moved closer to the camera. As expected, final probability is reduced from about 50% to 0% while decision uncertainty vanishes.

Figure 10. Non-target object at far distance (272cm. left) and close distance (92cm. right)

Figure 11. Effect of Illumination intensity on final decision. Non-target object / normal light: decision is 0% with 0.3% uncertainty. Non-target object / dark condition: decision becomes 25% with 48% uncertainty. Target object / normal light: decision is 92% with 0.1% uncertainty. Target object / dark condition: decision becomes 59% with 34% uncertainty.

Similarly for intensity, when environment becomes too dark or too bright, decision uncertainty increases and final probability becomes closer and closer to 50% as shown in figure 11.

5.2 Evaluation of Performance
A 100 sample per object per pose per orientation is collected to construct the database. This database is used for both adaptive Naïve Bayesian classifier and proposed system. A 1,000 measurement of each object at various environmental condition is collected. Table 4 states the result. Figure 12 shows a comparison between 3 systems: non-adaptive Naïve Bayesian classifier, adaptive Naïve Bayesian classifier, and proposed adaptive Tree-Augmented Naïve Bayesian Network under various distances. The results obtained from 120~170cm shows that proposed system decided that distance is not sufficient for decision and computed a near 50% probability. By providing better environmental modelling, this behavior can be adjusted. It’s worth noting that the main achievement the proposed system has over other systems is the very low false positive rate under very poor conditions. This is a result of conservatively fixing the TRUE/FALSE region definition as discussed in 4.2. As mentioned in related work, other approaches relying on machine learning are more aggressive resulting in higher false positive rates. It’s worth investigating of setting TRUE/FALSE regions free to follow peaks in multimodal likelihood distribution and the ramifications of this on the final results compared to current proposed fixed regions system results.

6. CONCLUSION
We have presented in this work the problems in classifications as well as other researchers’ efforts to overcome these problems. Since we developed a Naïve Bayesian classifier, we showed the problems associated with this approach and introduced Bayesian Network approach. We addressed the problem of environmental changes by modeling the characteristics of environmental parameters on the likelihood of each feature. We introduced a new probability inference operator “OR” and defined TRUE/FALSE regions in feature’s space. We have also addressed the problems of obtaining network inference parameters and network structure from statistical dataset and formulated probabilistic expressions for estimating expected posterior probability and feature information.
gain under an environmental condition. Finally, we have shown the result of the system under various environmental situations and validated and compared the results with other non-structural classifier.

7. ACKNOWLEDGMENTS
This work is partially supported by MEGA science research and development projects, funded by Ministry of Science, ICT and Future Planning (2013M1A3A02042335), and partially by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2010-0020210).

8. REFERENCES