Conditional Feature Sensitivity: A Unifying View on Active Recognition and Feature Selection

Xiang Sean Zhou[1], Dorin Comaniciu[1], Arun Krishnan[2]
[1]*Siemens Corporate Research* [2]*Siemens Medical Systems*
{xzhou, comanici}@scr.siemens.com, arun.krishnan@siemens.com

Abstract

The objective of active recognition is to iteratively *collect the next "best" measurements (e.g., camera angles or viewpoints), to maximally reduce ambiguities in recognition. However, existing work largely overlooked feature interaction issues. Feature selection, on the other hand, focuses on the selection of a subset of measurements for a given classification task, but is not context sensitive (i.e., the decision does not depend on the current input). This paper proposes a unified perspective through conditional feature sensitivity analysis, taking into account both current context and feature interactions. Based on different representations of the contextual uncertainties, we present three treatment models and exploit their joint power for dealing with complex feature interactions. Synthetic examples are used to systematically test the validity of the proposed models. A practical application in medical domain is illustrated using an echocardiography database with more than 2000 video segments with both subjective (from experts) and objective validations.*

1. Introduction

Active object recognition or active vision [2][9][15] [17][20][22][23][25] (or as called in robotics, active sensing/localization/navigation [18][10]) deals with a specific object or scene, searching for the next action, operator [13], or viewpoint [23][25], to optimize some objective function. Although these topics are intimately related to feature selection in machine learning [4][7][14][16], two key issues raised in the latter field has not been consciously considered by the former, namely, the necessity of an induction algorithm [14], and the possibility of complex feature interactions (e.g., in-class dependencies) [16]. As a result, an active vision system based on ad hoc heuristics may fail to fully reveal potential feature contributions. For example, most existing systems implicitly assume *feature independence* (which translates to *viewpoint independence* for object recognition using an active camera). However, in many cases two or more views are required to discriminate one class of objects from others.

Feature selection [14] for classification has recently been very active, with papers reporting notable progress [24]. Feature selection is essentially a search for the most sensitive feature subset for the purpose of improved classification accuracy and a significantly reduced feature set. However, these studies did not deal with a specific test input or *case-in-question* along with a *context.*

This paper attempts to bridge the two research fields by presenting a general framework of *conditional feature sensitivity analysis*. We assume all features for a given case to be uncertain but to different degrees—a measured feature (e.g., the visible patterns from the current camera angle) contains lower uncertainty, while a missing feature (e.g., the unseen or self-occluded parts of an object) has maximal uncertainty. Then, the question is: "given an induction algorithm, a labeling on a training set, and some *contextual information* for the case-in-question, what is the relative sensitivity for *all* features?" In other words, if we were to take more measurements, either on unmeasured features, or *to increase the accuracy of measured features,* what should we measure? Note that this framework subsumes both active vision problems and feature selection (when we have zero contextual information) (see Section 6). A key issue is how to deal with uncertainty in the contextual features, for which we devise several treatment models and exploit their joint power for dealing with complex feature interactions. We also put an emphasis on efficient implementation using sampling for likelihood approximation. Our sampling-based algorithms can trade off computation with effectiveness in detecting feature dependencies. We use real-world applications in medicine to underline the practical value of the proposed scheme.

We begin in Section 2 by outlining the proposed framework. Sections 3 and 4 contain the models, analysis, and some implementation issues. Experiments are presented in Section 5. Section 6 describes related work. Section 7 concludes the paper and lists future work.

2. Problem Formulation

The general system diagram is presented in Figure 1. The conditional feature sensitivity analysis module takes

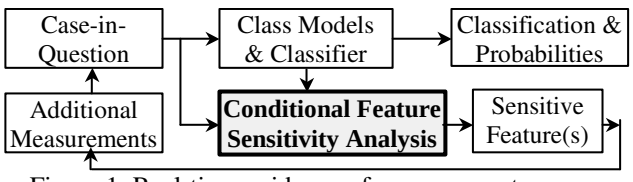

Figure 1. Real-time guidance of measurement process

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 IEEE

as inputs all the available information about the case in question (both missing and uncertain features) together with the class models (e.g., likelihood maps) provided by a given induction algorithm. The output is a list of sensitive features that should be measured next, to *maximally reduce uncertainty in classification*.

2.1 Defining Context

A case **S** has a measurement vector Z with M features, each of which has a prior distribution $p_0(z)$ over the whole population, a true value z for the current case, and a posterior distribution $p(z)$ after a measurement operation. From the algorithm viewpoint, a *feature with uncertainty* models both a *missing feature* and a *measured feature*. In other words, all features will be treated as uncertain features, with different degree of uncertainty parameterized in $p(z)$.

The ultimate goal is probabilistic classification, i.e., given a case $\mathbf{S}_i$, to obtain probabilities of its membership in all classes: $\{P(C_i = c_1), P(C_i = c_2), \ldots, P(C_i = c_K)\}$, where c_k, $k = 1, 2, \ldots, K$, are the class labels.

We use y to represent the feature under current study, $\{y\} \subseteq Z$. Without loss of generality we will assume y is 1-D for simplicity unless otherwise noted. The remaining features are represented by a vector X, $X = Z \setminus \{y\}$. We call y the *current feature(s)*, and X the *contextual features*. We denote the *context*—or what we know about the current case—by χ and ζ, representing the distributions of X and y, respectively. We abuse the expression "$y \in \zeta$" to mean "y has a distribution ζ" or "y is restricted to ζ." A particular sample drawn from ζ is denoted by y_i; and from χ, x_j. Note that when y is in 1-D, x_j is a $(N-1)$-dimensional vector. We will sometimes use the lowercase x (without a subscript) to represent a subset of Z.

2.2 Likelihood Sampling

A prerequisite for our later analysis is a way to deal with missing or uncertain features for both training and testing. A principled way of treating a missing feature is to sample (for training) or integrate (for testing) over its value range (or, its conditional distribution), an idea that parallels the EM and Bayesian treatments of missing data [11]. A traditional formulation [1] is as follows:

$$P(C = c_k \mid X \in \chi, y \in \zeta) = \frac{\iint_{\chi,\zeta} P(X, y) P(C = c_k \mid X, y) dX dy}{P(X \in \chi, y \in \zeta)} \quad (1)$$

$P(C \mid X, y)$ is obtained from the classifier. Here, one also needs to estimate the joint distribution of the features, $P(x, y)$. Since we will assume available a likelihood function for every class in the original feature space $P(X, y \mid C)$ (these can be approximated efficiently and we will discuss this issue in Section 0), the joint distribution is implied and we use the following formula:

$$P(C = c_k \mid X \in \chi, y \in \zeta) = \frac{P(C = c_k) \iint_{\chi,\zeta} P(X, y \mid C = c_k) dX dy}{\sum_k P(C = c_k) \iint_{\chi,\zeta} P(X, y \mid C = c_k) dX dy} \quad (2)$$

Here $P(C = c_k)$ is the prior probability of the k^{th} class.

3. Conditional Feature Sensitivity Analysis

We define the concept of conditional feature sensitivity as follows: *Given the context* $\{x \in \chi, y \in \zeta\}$ *for the case in question, further measurement of which feature(s) can maximally reduce uncertainty in classification?*

Although other criteria exist, the best gauge for uncertainty is entropy [6]. The reduction of entropy is the mutual information or information gain ([6], p. 18). Indeed, if one only considers the class label C and the current feature y, maximizing information gain corresponds to minimizing conditional entropy, and this in turn minimizes a bound on classification error according to *Fano's inequality* [6]. This is the foundation behind some early (unconditional) feature selection algorithms [3][4] and was in [13].

With contextual information coming into play, mutual information between C and y alone cannot in general reveal the potential information gain; and one shall appeal only to the *information gain* criterion for the right answer.

Since we have uncertainty in the contextual features, it is not a trivial task to formulate an information gain strategy directly. Based on different treatments of contextual uncertainties, we propose three models: mean-imputation model, integral model and sample-expectation model; or M-model, I-model, and S-model, respectively.

3.1 Mean-Imputation Model (M-Model)

The most straightforward way is to assign the mean values to the contextual features while working on the current feature. The information gain of y, IG_y, is defined as:

$$IG_y(C, X = \bar{x}, y \in \zeta) = H(C \mid X = \bar{x}, y \in \zeta) - \int_\zeta P(y \mid X = \bar{x}) H(C \mid X = \bar{x}, y) dy \quad (3)$$

where

$$H(C \mid X = \bar{x}, y \in \zeta) = -\sum_k P(C = c_k \mid X = \bar{x}, y \in \zeta) \log P(C = c_k \mid X = \bar{x}, y \in \zeta) \quad (4)$$

and

$$P(C = c_k \mid X = \bar{x}, y \in \zeta) = \frac{P(C = c_k) \int_\zeta P(X = \bar{x}, y \mid C = c_k) dy}{\sum_k P(C = c_k) \int_\zeta P(X = \bar{x}, y \mid C = c_k) dy} \quad (5)$$

The mean-imputation model is the simplest and most ef-

ficient. It can be practically very useful in the following scenario: when all measurements are done (with a mean value and a small variance), the doctor wants to know which feature is more sensitive than others, i.e., whether perturbations (due to, say, human or machine error) in one feature will cause more fluctuation in the final diagnosis than those of other features.

However, the M-model is less attractive when there are multiple missing features. Because of the simplified representation of the context, inevitably it can miss contextual structures and thus miss certain sensitive features.

3.2 Integral Model (I-Model)

A better way is to consider the full range of the contextual features:

$$IG_y(C, X \in \chi, y \in \zeta) = H(C \mid X \in \chi, y \in \zeta) - \int_{\zeta} P(y \mid X \in \chi) H(C \mid X \in \chi, y) dy \quad (6)$$

where

$$H(C \mid X \in \chi, y \in \zeta) = -\sum_k P(C = c_k \mid X \in \chi, y \in \zeta) \log P(C = c_k \mid X \in \chi, y \in \zeta) \quad (7)$$

Here, $P(C \mid X \in \chi, y \in \zeta)$ is evaluated according to the strategy discussed in Section 2.2.

$H(C \mid X \in \chi, y)$ is defined in a similar fashion.

The conditional probability can be expressed in terms of the likelihood as follows:

$$\begin{aligned} P(y \mid X \in \chi) &= \frac{P(y, X \in \chi)}{P(X \in \chi)} \\ &= \frac{\sum_k P(C = c_k) P(y, X \in \chi \mid C = c_k)}{\int_{\zeta} \sum_k P(C = c_k) P(y, X \in \chi \mid C = c_k) dy} \\ &= \frac{\sum_k P(C = c_k) \int_{\chi} P(y, X \mid C = c_k) dX}{\int_{\zeta} \sum_k P(C = c_k) \int_{\chi} P(y, X \mid C = c_k) dX dy} \end{aligned} \quad (8)$$

IG_y is now expressed in terms of $P(y, X \mid C)$, the prior distributions that we obtain through our generative classifier, and $P(C)$, the prior probability for each class.

All the integrals can be estimated either analytically if closed-form models are available, or by sampling within the uncertain range. In terms of sampling, we prefer randomized sampling to a deterministic sampling for the serendipity that could be brought forth in the randomness; in other words, random sampling has the capability of capturing unexpected irregularity in distributions.

An efficient sampling-based algorithm, CFS-I, that goes through every sample only once is presented in Appendix.

3.3 Sample-Expectation Model (S-Model)

For the S-model, the question to answer for the current feature is: "assuming that *we knew the context,* on average how much information could we gain from measuring the current feature?" The formula is:

$$\begin{aligned} EIG_{y|X}(C, X \in \chi, y \in \zeta) &= \mathrm{E}_X[IG_y(C, X, y \in \zeta)] \\ &= \int_{\chi} P(X \mid y \in \zeta) IG_y(C, X, y \in \zeta) dX \\ &= \int_{\chi} P(X \mid y \in \zeta)(H(C \mid X, y \in \zeta) - \int_{\zeta} P(y \mid X) H(C \mid X, y) dy) dX \end{aligned} \quad (9)$$

A sampling-based implementation, CFS-S, is presented in Appendix.

We can also define the sample-expectation model more generally but in a similar fashion in the form of $EIG_{y|x}$, where x is only a subset of X. (The formula and algorithm are analogous but omitted here due to space limit.)

4. Analysis and Implementation

In the following we analyze the working conditions for the three models and discuss implementation issues.

4.1 Within-Class Disjunctive Relationship

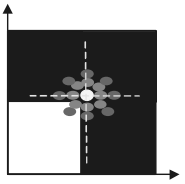

Figure 2. Type I masking

The M-model takes only the mean of each contextual feature. Its output will be biased whenever the means are biased representations of the context. An apparent example is shown in Figure 2, where one of the classes is defined by a disjunctive relationship, $C = (x > 0.5) \vee (y > 0.5)$. For a case in question with $\{x \in [0.4\ 0.65], y \in [0.4\ 0.65]\}$, it is sitting on the class boundary with both features sensitive (in that variations in one or both will alter the decision entropy). But the M-model will check only along the dashed lines thus report 0 sensitivity for both features. The mean value as the representation for one feature has "masked" out the sensitivity of the other. We call this *type I masking effect*. M-model will report 0 or low sensitivity for features under type I masking.

4.2 Within-Class Feature Dependency

When class boundary is defined *jointly* by more than one feature, the I-model may in some cases give significantly lower sensitivity estimate. Figure 3 shows some discrete or continuous examples, including the XOR relationship (i.e., $C = x \oplus y$, see [14]), and other feature dependencies within class(es). In the first three cases, with probability one, an integral along any line parallel to a coordinate axis will yield class distributions exactly the same as the prior class distribution, thus render the corresponding feature completely insensitive. The last case on

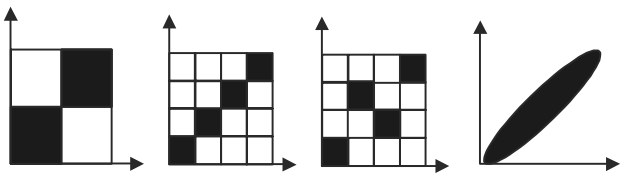

Figure 3. Feature dependency within classes

the right is the continuous counterpart for the second case.

This reveals the limitation of some existing information gain formula for feature selection [3][7] and active vision [9][13][25] since they are special cases of the I-model. However, our S-model will capture this type of sensitivity to its fullest extent due to the sampling of the context.

4.3 Prior Feature Dependency

If y is sensitive but it depends on another feature x, i.e., $y = \boldsymbol{g}(x)$ where $\boldsymbol{g}$ is a deterministic function; or more generally, if $p(C \mid y) \neq p(C)$ but $p(C \mid x, y) = p(C \mid x) \neq p(C)$, y is usually called a redundant feature, or a feature with weak relevance [14].

S-model will not report the weak relevance of y, even if x has not been measured. This is due to restricted sampling range for y given each sampled value of x. M-model cannot but I-model can detect such weak relevance.

We call this *type II masking effect* because the assumed values of a redundant contextual feature *mask* out the effect of the current feature when they are dependent.

4.4 Relationship between I-model and S-model

The I-model does not assume specific values for contextual features. It answers the question: "how much is the information gain if we measure the current feature?" The S-model assumes specific values before the information gain is calculated, and takes expectation afterwards. Thus, it answers a subtly different question: "If we knew the values of the contextual features, how much could the current feature provide in terms of information gain?"

For two (subsets of) features x and y, the following properties hold for the I-model and the S-model:

Property I: *The sum of IG_x and IG_y is not always equal to $IG_{x,y}$.*

Property II: *The sum of $EIG_{x|y}$ and $EIG_{y|x}$ is not always equal to $EIG_{x,y|-} = IG_{x,y}$. (by "-" we mean an empty feature set).*

Property III: $IG_x + EIG_{y|x} = EIG_{x,y|-} = IG_{x,y}$

The first two are merely rephrased statements of the preceding analysis. The example scenarios that support these two claims are the XOR relationship and the prior feature dependency, respectively.

We give a brief proof for Property III:

$$IG_x + EIG_{y|x} = H(C) - H(C \mid x) + \mathrm{E}_x[IG_{y|x}]$$

$$\equiv H(C) - \int_{\chi} P(x)H(C \mid x)dx + \int_{\chi} P(x)(H(C \mid x) - \int_{\zeta} P(y \mid x)H(C \mid x, y)dy)dx$$

$$= H(C) - \int_{\chi} P(x)(\int_{\zeta} P(y \mid x)H(C \mid x, y)dy)dx$$

$$= H(C) - \int_{\chi}\int_{\zeta} P(x, y)H(C \mid x, y)dydx \equiv IG_{x,y} \qquad (10)$$

4.5 Joint Analysis of Feature Dependencies

Based on the properties of Section 4.4, we can devise a testing scheme that combines outputs from multiple models to expose complicated feature dependencies. By examining the outputs of both the I-model and the S-model we could detect different feature sensitivity or relevance scenarios. The following table shows an example for two features, x and y:

Table 1. An example for joint analysis of two features

IG_x	IG_y	$EIG_{x\|y}$	$EIG_{y\|x}$	IG_{xy}	*Relevant features*	Notes
0	0	0	0	0	-	x, y: weak or no
0	0	+	+	+	*x,y*	Both strong, e.g., $C=x \oplus y$
0	+	0	+	+	*y*	y: Strong; x: weak or no
+	0	+	0	+	*x*	x: Strong; y: weak or no
+	+	0	0	+	*x,y*	Both weak, e.g., $x=f(y)$
+	+	+	+	+	*x,y*	Both strong

It is worth noting that for the first five columns in Table 1, we only need to calculate three because of the constraint implied by Property III.

Joint analysis of two features can only expose dependency involving less than three features. In case there are dependencies between more than two features, we will need to consider joint analysis of more than two features. For example, if $C = x \oplus y \oplus z$, and we have another three redundant features, $x' = x$, $y' = y$, and $z' = z$, analyzing two features at a time and we will arrive at the first row of Table 1. Only joint three-feature analysis such as $IG_{x,y,z}$ can reveal the contributions of x or x', y or y', and z or z'.

An easy implementation for $IG_{x,y,z}$ or $EIG_{x,y,z|-}$ is through nested calls to the single-feature IG function based on Property III in Section 4.4. For example,

$$IG_{x,y,z} = IG_x + EIG_{y,z|x} = IG_x + \sum_i P(x_i)IG_{y,z|x=x_i}$$

$$= IG_x + \sum_i P(x_i)(IG_{y|x=x_i} + EIG_{z|x=x_i,y})$$

$$= IG_x + \sum_i P(x_i)(IG_{y|x=x_i} + \sum_j P(y_j)IG_{z|x=x_i,y=y_j}) \qquad (11)$$

$$= IG_x + \sum_i P(x_i)(IG_{y|x=x_i} + \sum_j P(y_j)IG_{z|x=x_i,y=y_j}) \qquad (12)$$

4.6 Induction Algorithm Implementation

Because we aim at probabilistic classification, and our models take full advantage of likelihood functions, we shall choose an induction algorithm that can learn probabilistic models from the training data. We use kernel dis-

criminant analysis [19] combined with generative modeling for this purpose. The kernel discriminant has been shown to have comparable performance as SVM [19]. In addition, it can provide a low dimensional, non-linearly transformed subspace, in which simple probability models can be built. We use RBF kernel with an empirically determined spread.

In real world applications, it is often necessary to expect missing feature values in the training data. We apply data imputation (through sampling) to fill in missing (or uncertain) values for feature(s) y, based on $p(y \mid x)$ where x represents the remaining features with known values. We estimate $p(y \mid x)$ using the training set. Robust estimates are used to reduce the influence of outliers.

5. Experiments

In this section we evaluate the proposed schemes on both synthetic data and real-world medical data.

5.1 Synthetic Data

To gain more insights, we decouple the comparison of the models from the evaluation of the induction algorithm. Two sets of experiments are compared: one uses an ideal induction algorithm or ideal likelihood maps (Figure 4), the other uses kernel discriminant analysis and likelihood back-projection to learn the maps from training sets (Table 2, Figure 5).

In Figure 4, cases I to III show that relative feature sensitivity changes with the context. Case II also shows that more uncertainty in measurement does not necessarily lead to higher sensitivity.

Some scenarios in Figure 4 are carefully selected to reveal "blind-spot" of our models: for cases IV, V, and VI, M-model gave biased estimates. Also note the subtle difference in context for cases V and VI with yet a large change in M-model outputs. This indicates the lack of robustness of the M-model as compared to the other two. I-model failed to capture feature sensitivity in case II (refer to analysis in Section 4.2). The S-model gives reasonable results on all these cases.

Figure 5 is based on the data sets listed in Table 2. The results correlate largely to that of Figure 4, demonstrating that kernel machines can reliably capture nonlinear configurations without over-fitting (e.g., x_3), and can support meaningful conditional feature sensitivity analysis.

Table 2. Synthetic training sets

Set I	**Class A (8 points)**	**Class B (8 points)**
x_1	0.2 0.2 0.4 0.4 0.6 0.6 0.8 0.8	0.2 0.2 0.4 0.4 0.6 0.6 0.8 0.8
x_2	0.2 0.4 0.2 0.4 0.6 0.8 0.6 0.8	0.6 0.8 0.6 0.8 0.2 0.4 0.2 0.4
x_3	0.2 0.4 0.6 0.8 0.2 0.4 0.6 0.8	0.2 0.4 0.6 0.8 0.2 0.4 0.6 0.8

Set II	**Class A (8 points)**	**Class B (8 points)**
x_1	0.2 0.2 0.4 0.45 0.2 0.2 0.35 0.6	0.55 0.6 0.8 0.8 0.65 0.8 0.8 0.4
x_2	0.2 0.4 0.2 0.4 0.6 0.8 0.6 0.2	0.6 0.8 0.6 0.8 0.4 0.2 0.4 0.8
x_3	0.2 0.4 0.6 0.8 0.2 0.4 0.6 0.8	0.8 0.6 0.4 0.2 0.6 0.4 0.2 0.8

Set III	**Class A (4 points)**	**Class B (12 points)**
x_1	0.2 0.2 0.4 0.4	0.2 0.2 0.4 0.4 0.6 0.6 0.8 0.8 0.6 0.6 0.8 0.8
x_2	0.2 0.4 0.2 0.4	0.6 0.8 0.6 0.8 0.2 0.4 0.2 0.4 0.6 0.8 0.6 0.8
x_3	0.2 0.4 0.6 0.8	0.2 0.4 0.6 0.8 0.2 0.4 0.6 0.8 0.2 0.4 0.6 0.8

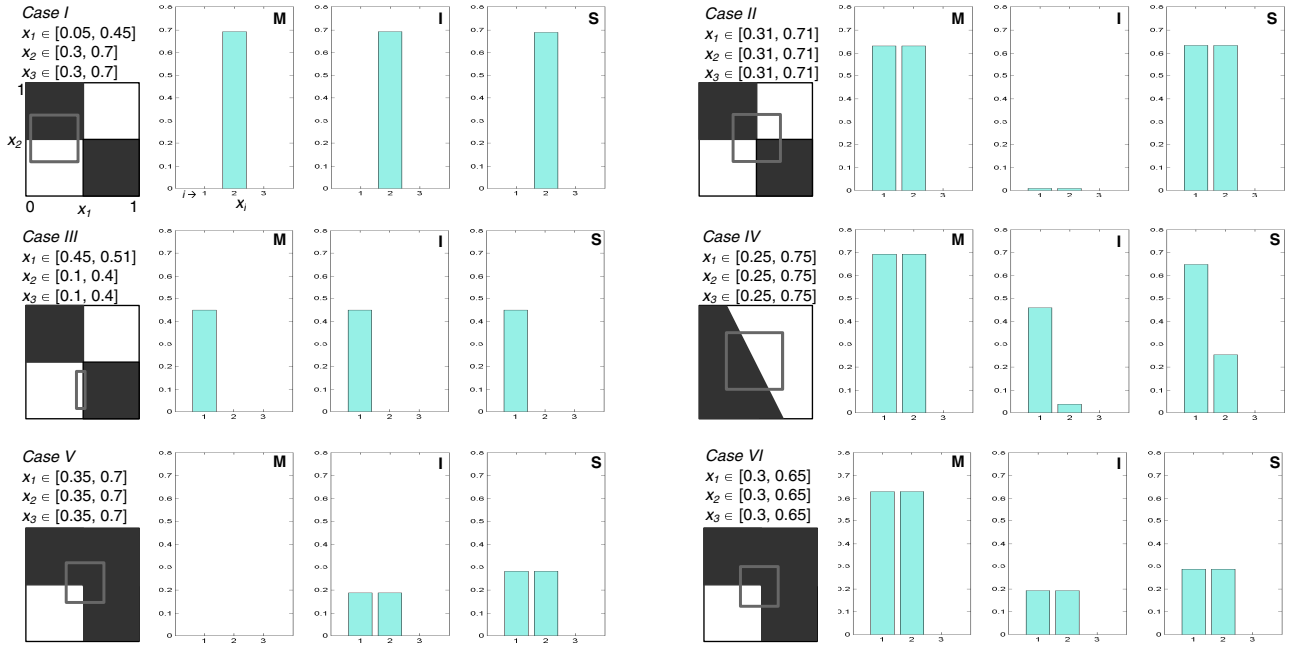

Figure 4. Comparing the three models using ideal likelihood maps. The boxes outline the *context*. "M", "I", and "S" indicate the outputs of the three models, respectively. x_3 is a irrelevant dummy variable for compatibility with Figure 5. For Case II, we used [0.31 0.71] instead of [0.3 0.7] to avoid ambiguity on the class boundary, which can affect the output of the M-model.

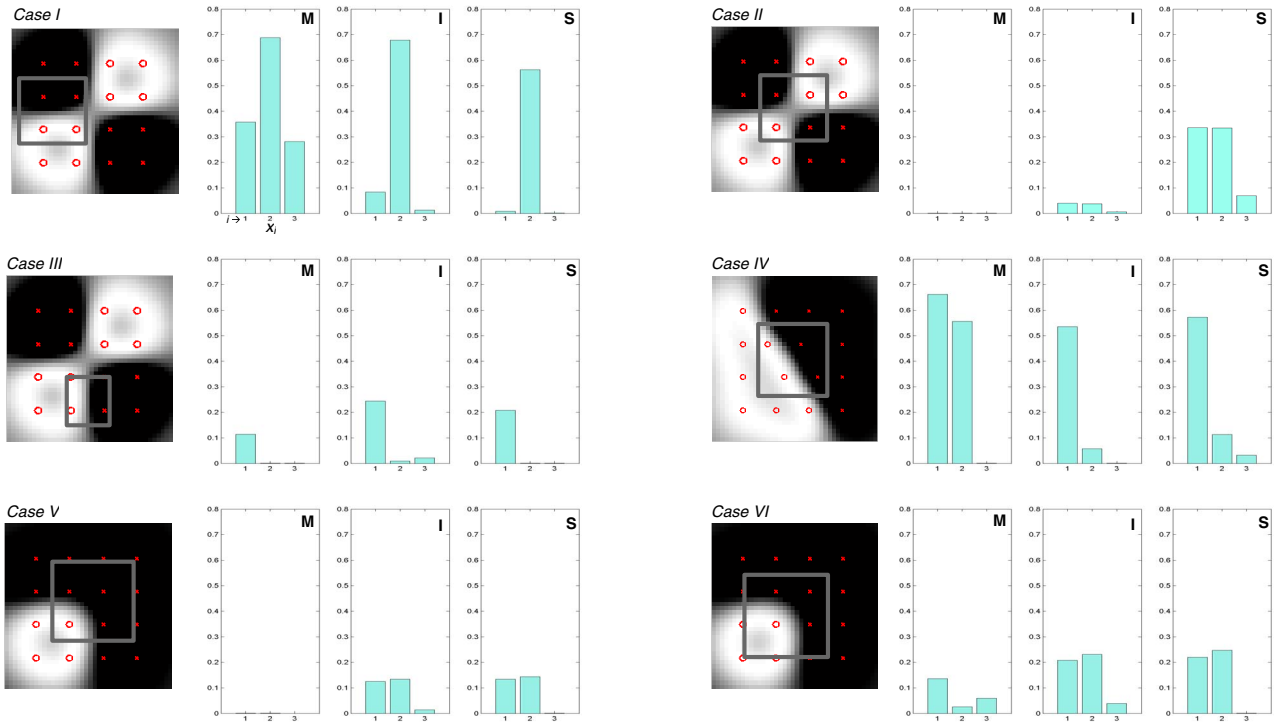

Figure 5. Comparing the three models using likelihood maps learned using kernel discriminant analysis (RBF kernel with $\sigma = 0.1$)

5.2 Real-time System for Echocardiography

Due to page limit, we will omit experiments on pseudo-real-world datasets such as the Columbia object database [23], instead we present a more practical and interesting active recognition problem in medicine, namely, the *on-line* recognition of a diseased heart using visual descriptors *during the data acquisition process*. During an echocardiograph exam, the number of possible measurements is in the hundreds [21], but a typical echo study at clinics in the United States only contains about ten different measurements on average. The appropriate selection of additional measurements requires extensive training and field experience and is therefore subjective and error-prone. It would be very helpful if a machine could provide context-sensitive real-time guidance as to what additional feature(s) should be measured for the current case. A feature sensitivity analysis module also provides guidance and schedules for a pool of automatic vision algorithms: e.g., border detector and motion tracker.

We have developed a system for real-time diagnosis support in echocardiography with optical flow-based contour tracking for calculating heart volume, ejection fraction, and ischemia regions, etc. The system uses a database with about 2000 echocardiography videos, labeled by medical expert into different types and stages of cardiomyopathies. Figure 9 shows the system at work for diagnosing DCM or nonDCM: after LVED (left ventricle end diastolic dimension) is measured to 7.6cm (95%th percentile of healthy heart is <5.8cm according to [21] Apendix II), the system outputs a probability of DCM at 86.2%, but recommended further measurements on LVES (LV end systolic dimension), wall thickness, EF (ejection fraction), etc., in that order.

The features recommended by the system were verified by medical experts and were found to be in accordance with medical textbook descriptions.

To objectively measure the savings brought forth by the system, we designed an experiment to simulate the real-time interaction process. We selected 28 DCM, 32 nonDCM, and 4 borderline cases, most of which have a full measurement vector of ten components. For each case, we start out by assuming no measurements and ask the system to recommend the first feature to measure;[1] And then fill in the real measurement for that feature and iterate until the system *reliably* reaches a probability assessment of the case within ε of that obtained from all N measured features. Let P_k denote the DCM probability using the first k features recommended by the system, we record $n_\varepsilon = \boldsymbol{min}\{n \mid P_N-\varepsilon<P_{n+i}<P_N+\varepsilon,\ i=0,1,\ldots,N-n\}$ for every case and report the averages in Table 3.

The saving is evident, especially for extreme cases (Θ_1), i.e., either very healthy or very diseased cases.

[1] For DCM vs. nonDCM, LVED is the most sensitivity feature given no context. However, subsequent feature sensitive ordering is case or context-dependent, assuming nonlinear class boundaries.

Table 3. Average number of necessary measurements

Case subset: Θ	$\overline{N}$	$\overline{n}_{10\%}$	$\overline{n}_{5\%}$	$\overline{n}_{2\%}$	$\overline{n}_{1\%}$
Θ_0={**S**}, $\|\Theta_0\|$=64	9.44	3.30	4.11	5.02	5.41
Θ_1={**S** | P_N<5% or P_N>95%}, $\|\Theta_1\|$=41	9.49	2.66	3.44	4.32	4.68
Θ_2={**S** | 5%≤P_N≤95%}, $\|\Theta_2\|$=23	9.35	4.43	5.30	6.26	6.70
Θ_3={**S** | **S** =DCM or unsure}, $\|\Theta_3\|$=32	9.38	2.88	3.72	4.69	5.09
Θ_4={**S** | **S** is nonDCM}, $\|\Theta_4\|$=32	9.50	3.72	4.50	5.34	5.72

6. Related Work

Much research in active vision and robotics has used similar heuristics for active selection of best features ([10][15][17][18][20][22][23]). For example, [2] and [25] used reduction of entropy to guide the selection of viewpoints; [9] proposed optimal sensor parameter selection for iterative state estimation in static systems by maximizing mutual information; and [13] proposed information gain-based selection of "imaging operators", taking into account also operation costs. However, none of above formally addressed the role of an induction algorithm for feature analysis as well as the issue of feature interaction. They are at best equivalences of the I-model. The general framework presented here can provide principled extensions for these applications.

Since our analysis parallels some of the latest developments in feature selection research, it is beneficial to also compare the proposed framework with feature selection [3][4][7][14][16]. This field has seen notable advances in recent years, with methods categorized into either a filter model [8] that traditionally treats feature selection solely as a preprocessing step for later induction algorithm design; or a wrapper model [14] that performs cross validation of an induction algorithm on the training set. There have also been efforts to link these two models [8][16]. We can regard conditional feature sensitivity analysis as a *local* feature selection problem based on the *context*. However, existing algorithms are not directly applicable. For example, the wrapper approach [14] relies on cross-validation but oftentimes we will not have sufficient training samples to cross-validate in the neighborhood defined by the context—especially when more than a few features have been measured; on the other hand, most variants of the filter approach do not address the *context* issue, and often ignore the induction algorithm altogether.[2]

Our analysis combines the essence of both the wrapper model and the filter model and puts an explicit emphasis on the modeling of *contextual features*. This new viewpoint may also shed some lights on feature selection because it subsumes feature selection: when there is nothing to condition upon (i.e., no measurement or prior information), our I-model reduces to a mutual information-based feature selection algorithm ([3], *cf.* Definition 1 in [14]). However, after this reduction, the explicit consideration of the context will be lost because the integral over X's full range completely marginalizes X out (Eq. (2) and Eq. (8)). On the other hand, at its extreme case, our S-model reveals an algorithm for feature selection that has not been explored in the feature selection literature.

Notice also the different stop criteria: feature selection process stops only when all relevant features are found; conditional feature sensitivity analysis stops when a case can be classified with sufficient confidence—the feature subset examined can be much less than the relevant set.

One can also find some related research on belief networks (e.g., Bayesian nets or influence diagrams) regarding *value of information* [12][22][5] with, however, different viewpoints or limited treatments.

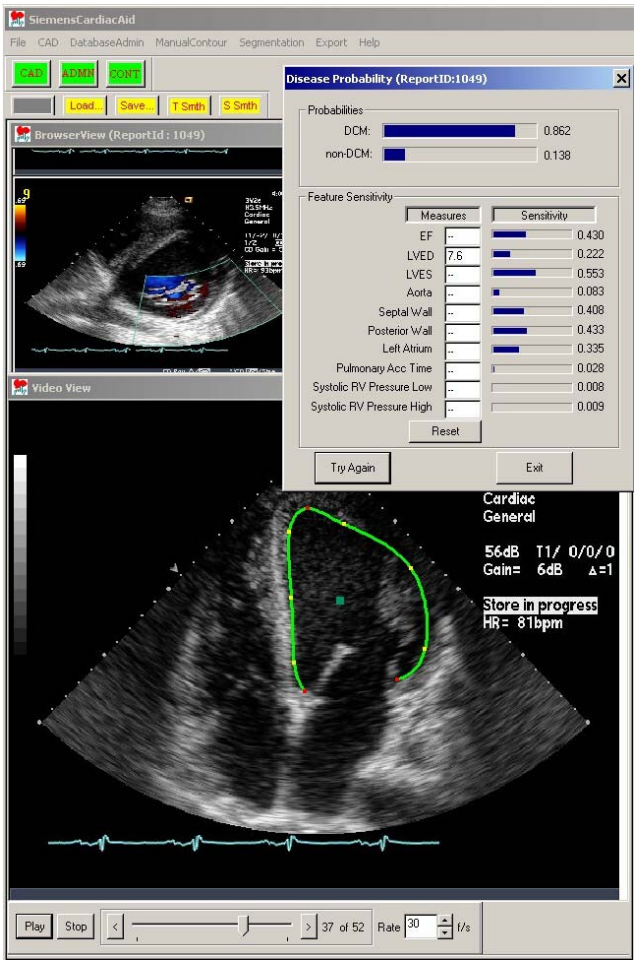

Figure 6. Computer-aided diagnosis and data acquisition guidance system for cardiac ultrasound.

7. Conclusion and Future Work

We presented a general framework for *conditional fea-*

[2] Consulting an induction algorithm is necessary during the course of feature evaluation, because the most sensitive feature is *not* necessarily the one that leads to the most variability in labels (which may lead to minimal empirical error on the training data but large error on test data); the best feature shall lead to the most *systematic and predictable* variability in labels [14].

ture sensitivity analysis that provides feature sensitivities under a case context, incorporates rigorous analysis of feature interaction issues, and finds practical applications for real-time active vision in medicine. Future work shall include consideration of measurement costs [13].

Acknowledgement

We would like to thank Visvanathan Ramesh from Siemens Corporate Research for fruitful discussions.

APPENDIX

Algorithm 1: CFS-I

For every feature of interest y:
- $P_x = 0$; $Sum\_H_{yi} = 0$; $L_k = 0$, $k = 1, \ldots, K$;
- For every sample of y, y_i :
  - $L_{yi,k} = 0$, $k = 1, \ldots, K$;
  - For every sample of X, X_j :
    Accumulate $L_{yi,k} += P(X_j, y_i \mid C = c_k)$;
  - Calculate Bayesian posterior, $P_{yi,k}$, from $L_{yi,k}$;
  - Calculate Entropy H_{yi} over $P_{yi,k}$;
  - Calculate $P_{yi} = \sum_k L_{yi,k} P(C = c_k)$;
  - Accumulate $Sum\_H_{yi} += P_{yi} H_{yi}$;
  - Accumulate (marginalize): $P_x += P_{yi}$;
  - Accumulate $L_k += L_{yi,k}$;
- Calculate Bayesian posterior, P_k, from L_k;
- Calculate Entropy H_y over P_k;
- Calculate information gain: $IG_y = H_y - Sum\_H_{yi}/P_x$.

Algorithm 2: CFS-S

For every feature of interest y:
- $P_y = 0$; $Accu\_IG_y = 0$;
- For every sample of X, X_j:
  - $P_x = 0$; $Sum\_H_{yi} = 0$; $L_k = 0$, $k = 1, \ldots, K$;
  - For every sample of y, y_i :
    Calculate $L_{yi,k} = P(X_j, y_i \mid C = c_k)$;
    Calculate Bayesian posterior, $P_{yi,k}$, from $L_{yi,k}$;
    Calculate Entropy H_{yi} over $P_{yi,k}$;
    Calculate $P_{yi} = \sum_k L_{yi,k} P(C = c_k)$;
    Accumulate (i.e., marginalize): $P_x += P_{yi}$;
    Accumulate $Sum\_H_{yi} += P_{yi} H_{yi}$;
    Accumulate $L_k += L_{yi,k}$;
  - Calculate $P_{xj} = \sum_k L_k P(C = c_k)$;
  - Accumulate (i.e., marginalize): $P_y += P_{xj}$;
  - Calculate Bayesian posterior, P_k, from L_k;
  - Calculate Entropy H_y over P_k;
  - Accumulate the information gain:
  - $WeightedSum\_IG_y += P_{xj}(H_y - Sum\_H_{yi}/P_x)$.
- $IG_y = WeightedSum\_IG_y / P_y$.

References

[1] S. Ahmad and V. Tresp, "Some solutions to the missing feature problem in vision," *Advances in Neural Information Processing Systems(NIPS)* 5, pp. 393-400, 1993.

[2] T. Arbel, F. P. Ferrie, "Viewpoint selection by navigation through entropy maps," *ICCV*, Corfu, Greece, Sept. 1999.

[3] R. Battiti, "Using the mutual information for selecting features in supervised neural net learning," *IEEE trans. on neural networks*, 5(4), pp. 537-550, 1994.

[4] C. Cardie and N. Howe, "Improving minority class prediction using case-specific feature weights," in *Proc. ICML*, pp. 57-65, 1997.

[5] H. Chan and A. Darwiche, "When do numbers really matter?" *Journal of AI Research*, 17, pp. 265-287, 2002.

[6] T. Cover and J. Thomas, *Elements of Information Theory*, John Wiley & Sons, New York, 1991.

[7] W. Daelemans, A. Van den Bosch, and T. Weijters, "IGTree: using trees for compression and classification in lazy learning algorithms," *AI Review*, 11, pp.407-423, 1997.

[8] S. Das, "Filters, wrappers, and a boosting-based hybrid for feature selection," in *Proc. ICML*, 2001.

[9] J. Denzler and C. Brown, "Information theoretic sensor data selection for active object recognition and state estimation," *IEEE trans. PAMI*, 24(2), pp. 145-157, Feb. 2002.

[10] G. DeSouza and A. Kak, "Vision for mobile robot navigation: A survey," *IEEE trans. PAMI*, 24(2), Feb. 2002.

[11] Z. Ghahramani and M. Jordan, "Supervised learning from incomplete data via an EM approach," in *NIPS 6*, 1994.

[12] D. Heckerman, E. Horvitz, and B. Middleton, "An approximate nonmyopic computation for value of information," *IEEE Trans. on PAMI*, 15(3), pp.292-298, 1993.

[13] R. Isukapalli and R. Greiner, "Efficient interpretation policies," in *Proc. IJCAI,* 2001.

[14] G. John, R. Kohavi, and K. Pfleger, "Irrelevance features and the subset selection problem," in *Proc. ICML*, 1994.

[15] H. J. Kappen, M. J. Nijman, and T. van Moorsel, "Learning active vision," in *Proc. ICANN*, Oct. 1995.

[16] D. Koller and M. Sahami, "Toward optimal feature selection," in *Proc. ICML*, 1996.

[17] E. Marchand and F. Chaumette. "Active vision for complete scene reconstruction and exploration," *IEEE trans. PAMI*, 21(1), pp. 65-72, Jan. 1999.

[18] L. Mihaylova, et al., "Active sensing for robotics – A survey," in *Proc. 5th Int'l Conf. On Numerical Methods and Applications*, 2002.

[19] S. Mika, G. Ratsch, and K.-R. Muller. "A mathematical programming approach to the kernel Fisher algorithm," in *NIPS* 13, 2001.

[20] S. Moorehead, R. Simmons, and W.L. Whittaker, "Autonomous exploration using multiple sources of information," *IEEE Int'l Conf. Robotics and Automation*, 2001.

[21] J. Oh, J. Seward, and A. Tajik, *The Echo Manual*, Lippincott Williams & Wilkins, Philadelphia, 1999.

[22] L. Ortiz and L. Kaelbling, "Sampling methods for action selection in influence diagrams," in *Nat'l Conf. AI*, 2000.

[23] B. Schiele and J. Crowley, "Transinformation for active object recognition," *ICCV*, 1998

[24] N. Slonim, G. Bejerano, S. Fine, and N. Tishby, "Discriminative feature selection via multiclass variable memory Markov model," in *Proc. ICML*, 2002.

[25] M. Sipe and D. Casasent, "Feature space trajectory methods for active computer vision," *IEEE trans. PAMI*, 24(12), pp. 1634-1643, Dec 2002.