Visual Explanation of Evidence in Additive Classifiers

Brett Poulin, Roman Eisner, Duane Szafron, Paul Lu, Russ Greiner, D.S. Wishart, Alona Fyshe, Brandon Pearcy, Cam MacDonell and John Anvik

Department of Computing Science, University of Alberta
Edmonton, Canada
{poulin, eisner, duane, paullu, greiner, dwishart, alona, bpearcy, cam, janvik }@cs.ualberta.ca

Abstract

Machine-learned classifiers are important components of many data mining and knowledge discovery systems. In several application domains, an explanation of the classifier's reasoning is critical for the classifier's acceptance by the end-user. We describe a framework, ExplainD, for explaining decisions made by classifiers that use additive evidence. ExplainD applies to many widely used classifiers, including linear discriminants and many additive models. We demonstrate our ExplainD framework using implementations of naïve Bayes, linear support vector machine, and logistic regression classifiers on example applications. ExplainD uses a simple graphical explanation of the classification process to provide visualizations of the classifier decisions, visualization of the evidence for those decisions, the capability to speculate on the effect of changes to the data, and the capability, wherever possible, to drill down and audit the source of the evidence. We demonstrate the effectiveness of ExplainD in the context of a deployed web-based system (Proteome Analyst) and using a downloadable Python-based implementation.

Introduction

A *classifier* assigns a label to an unlabeled query item based on patterns learned from labeled examples. Such *machine-learned* classifiers are important components in data mining and knowledge discovery systems. They have been used in many applications, including protein function prediction (Szafron et al. 2004), fraud detection (Fawcett and Provost 1997; Fawcett and Provost 1999), medical diagnosis (Kononenko 2001), and text classification (Joachims 2002; Dhillon, Mallela and Kumar 2002).

There are a variety of techniques for learning classifiers (Alpaydin 2004; Hastie, Tibshirani, and Friedman 2001) including decision trees, naïve Bayes (NB), support vector machines (SVM), logistic regression, linear discriminant analysis and artificial neural networks (ANN). Whatever the classification technology, it is desirable for the resulting system to be able to transparently explain its predictions to help the user to identify possible causes of errors and misclassifications. Swartout (1983) emphasized:

> Trust in a system is developed not only by the quality of the results but also by clear description of how they were derived. … a user … should be able to ask for a description of the methods employed and the reasons for employing them.

For classifiers in particular, and for data mining and knowledge discovery systems in general, users often want to validate and explore the classifier model and its output. Often, there are conflicting and complicated units of evidence that contribute to the classification decision. To address these issues, the classification system should have a sound, intuitive, and interactive explanation capability.

Graphical explanations help users understand the evidence for a classification decision, thus engendering trust in the decision. Graphical explanation also helps users to efficiently visualize the evidence and to drill down to the source of the evidence (especially where classifications are unexpected or erroneous), providing an audit of the classifier and the raw data that was used to train it.

Related Work

Some classification techniques, such as decision trees, offer immediate explanation of germane decision factors. However, decision trees, are not appropriate for all classification tasks and not always the most accurate. We focus on classifiers whose results are not so easily explained. There has been some previous research in this area. For example, Madigan, Mosurski, and Almond (1996) described Bayesian belief networks using the useful idea of 'weights of evidence', which relates closely to our work. While there are some graphical explanation facilities for naïve Bayes classification (Becker, Kohavi, and Sommerfield 1997; Mozina et al. 2004; Ridgeway, Madigan, and Richardson 1998), there are many classifier techniques with no published explanation capabilities (graphical or otherwise). ExplainD has more visualization and auditing capabilities than previous systems.

Nomograms (Mozina et al. 2004) have effective explanation capabilities for NB classification and logistic regression that are closely related. Our framework, however, offers some advantages over nomograms. Our ExplainD, for example, allows explanation of a broader range of classifiers, including linear SVM.

Contributions

- We derive graphical explanation capabilities, including decision, decision evidence, decision speculation, ranks of evidence, and source of evidence.
- We provide a framework to support these capabilities for several widely used classifiers, including naïve Bayes, SVMs, and many additive models.
- We provide two implementations of the framework:

  1) A Python implementation of our generic explanation system (downloadable from http://www.cs.ualberta.ca/~bioinfo/explain) explains NB, linear SVM, and logistic regression classifiers.

  2) An NB version of the ExplainD system has been deployed in the Proteome Analyst (PA) system (Szafron et al. 2004) (http://www.cs.ualberta.ca/~bioinfo/PA) since August 2002. PA is a web-based bioinformatics tool used by hundreds of molecular biologists.

Framework

In this section, we derive a set of basic explanation capabilities and a simple explanation framework. This framework facilitates graphical explanation of classifier decisions, evidence for those decisions, and insight into the classifier as a whole. More information can be found at www.cs.ualberta.ca/~bioinfo/explain.

Classifiers as Additive Models

A classifier maps an object described by a set of feature values to one of the possible mutually exclusive class labels. This assignment can be based on a set of discriminant real-valued functions, one function $g_k(x)$ for each possible class label k (Alpaydin 2004). Given a query instance x (with feature values $x_1 \ldots x_j \ldots x_m$),, each discriminant provides a score for assigning a class label. The instance is classified with the label of the highest scoring class given the instance.

$$\text{classification of } x = \arg\max_{k \quad labels} g_k(x)$$

When considering only two mutually exclusive classes, such as *positive* and *negative*, a single discriminant function may be used.

$$g(x) = g_+(x) \quad g \ (x)$$

This leads to a simple decision function.

$$\text{classification of } x = \begin{cases} + & \text{if } g(x) > 0 \\ & \text{otherwise} \end{cases} \qquad (1)$$

For a certain class of discriminant models, the value of the discriminant changes linearly with the value x_j of each feature j of the instance x. The score from such a linear discriminant model is the sum of the intercept b and the contributions of each feature, which is the product of the feature value x_j and the feature weight w_j.

$$g(x) = b + \sum_{j=1}^{m} w_j x_j$$

These w_j values are the parameters of this model. We will see that there are many ways to learn good parameter values from labeled training examples.

We can generalize this linear model to an additive model (Hastie, Tibshirani, and Friedman 2001) in which the feature contributions are functions of the feature values.

$$g(x) = b + \sum_{j=1}^{m} f_j(x_j) \qquad (2)$$

Each contribution $f_j(x_j)$ is the contribution of evidence from feature j with value x_j to the score of the discriminant function $g(x)$. We show below that, when a classifier corresponds to an additive model, the classifier and the classification results can each be explained in terms of the components of the model. The basic structure of additive models forms the basis for our explanation framework.

For brevity, we describe a framework for two-class classification tasks (*positive* or *negative* for a particular class label). The framework may be extended to multiple classes in a straightforward manner.

Classification Explanation Capabilities

Our graphical explanation framework includes five capabilities. Each successive capability increases the user's ability to understand and audit an aspect of the classification process, based on the evidence. These explanation capabilities are centered on the user's ability to explain particular classifier decisions. They also facilitate visualization of the entire classifier and the training data.

1. *Decision:* Represent a predicted classification graphically (Figure 1 LHS).
2. *Decision Evidence*: Represent the relative strength of potential classification decisions and the contributions of each feature to the decisions (Figure 1 RHS). Confidence intervals may be displayed where possible.
3. *Decision Speculation*: An interactive 'what-if?' analyses by changing feature values (Figure 2 LHS).
4. *Ranks of Evidence*: Represent feature evidence in the context of the overall classifier (Figure 2 RHS).
5. *Source of Evidence*: Represent (where possible) the data supporting evidence contributions (Figure 3).

Each explanation capability is discussed in detail in the next sections. We illustrate ExplainD using an example of the diagnosis of obstructive coronary artery disease (CAD) based on a classification model actually used by physicians (Gibbons 1997; Morise *et al*. 1992). A physician uses the classifier to predict the probability that an undiagnosed 35-year-old male has CAD. We also use examples from Proteome Analyst (Szafron *et al*. 2004), a web-based proteome prediction tool used by molecular biologists, which contains a full implementation of ExplainD.

Capability 0 – decision: The *decision* capability represents the outcome visually (Figure 1 LHS). This baseline explanation capability presents the 'native' outcome of each classifier. Depending on the classifier, this may be a binary indicator, a score, or a probability.

In the CAD example, the *decision* is whether to predict CAD, based on patient history and test results (Figure 1

LHS-top) or, for a probabilistic classifier, the *probability* that the patient has CAD, based on the evidence (Figure 1 LHS-bottom). While this visualization allows a physician to conclude that the patient does not have CAD, it does not provide other information or evidence.

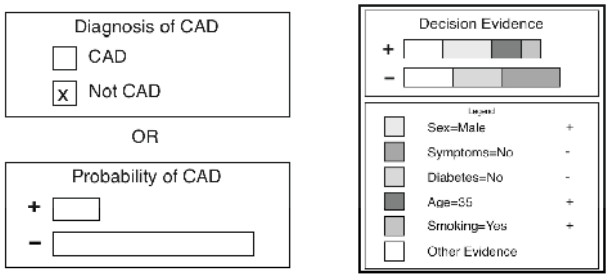


Figure 1: Capability 0 – *decision* (LHS) and Capability 1 – *decision evidence* (RHS)

Capability 1 – decision evidence: The *decision evidence* represents the strength of the classification decision and the relative contributions of each feature to the decision. The classifier is considered an additive model (2).

We can represent the overall score of the discriminant function $g(x)$ as two stacked bars (Figure 1 RHS). The evidence in each bar is composed of the individual evidence contributions $f_j(x_j)$ based on the value of each feature. All positive evidence ($f_j(x_j) > 0$) is combined in one bar, and all negative evidence is combined in the other (Figure 1 RHS-top). An intercept b (constant) can also be displayed in the appropriate bar depending on its sign. A legend notes the feature corresponding to each bar segment (Figure 1 RHS-bottom). The difference between the two bars is the overall score of the discriminant function.

The longer of the positive or negative bars indicates the classification – equivalent to (1):

$$\text{classification of } x = \begin{cases} + & \text{if (score for positive)} > \text{(score for negative)} \\ - & \text{otherwise} \end{cases}$$

This visualization simultaneously represents the strength of the predicted classification (magnitude of the difference between bars) and the relative contribution of each feature to the predicted classification. For classifiers that supply them, confidence values may be displayed for each bar.

In the diagnosis of CAD, the overall negative evidence is greater than the overall positive evidence, indicating that it is more likely that the patient does not have CAD. The physician might also observe which factors give the most classifier evidence for CAD in this patient (Figure 1 RHS).

When there are many features, it is helpful to show the evidence contributions from only a subset of 'focus' features. The contributions of 'non-focus' features are combined into 'aggregate' terms that are displayed as 'Other Evidence' segments – one positive and one negative (Figure 1 RHS). The focus features may be selected interactively or by some specified criteria (such as those that give the most evidence for the classification). Since the 'Other Evidence' segments may be large compared to the focus features, it may also be helpful to zoom in on the portion of the chart that shows the focus features.

Capability 2 – decision speculation: One of the key components to facilitate understanding of the classifier and classification decisions is the ability to examine how the classification would change if feature values changed. This 'what-if?' analysis allows speculation about the effect that a change in feature values would have on the decision. This capability allows the user to interactively change the feature values of an instance and visually audit changes in classification decisions. This capability can help users to explore and better understand the classification model and help experts examine the model for unexpected behaviors.

In the CAD example, *decision speculation* allows the physician (or patient) to view the effects of risk factors on the diagnosis (Figure 2 LHS). They can explore the effects of changing lifestyle factors such as smoking. If the patient were 20 years older and still smoking, we would observe a prediction change to 'CAD' (Figure 2 LHS-middle). However, if the patient quits smoking, the prediction would change back to 'not CAD' (Figure 2 LHS-bottom). The 'what-if?' analysis can be used as an educational tool.

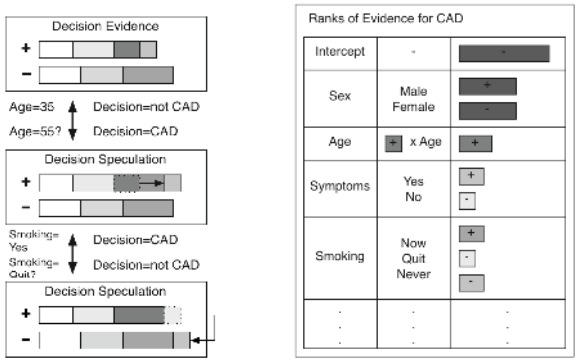


Figure 2: Capability 2 – *decision speculation* (LHS) and Capability 3 – *ranks of evidence* (RHS)

Decision speculation should not be confused with the capability to speculate on the effects that changes to the *training data* can have on classifier decisions (such as removal of outliers, over-represented or suspicious data). In general, 'Training Speculation' may require complete re-training of the classifier. This capability may be useful in some cases, but it is not included in our framework.

Capability 3 – ranks of evidence: Since the classifier is defined by the components of the additive model (the feature contribution functions $f_j(x_j)$ and the intercept b), we can display the information that defines the *entire classifier* with visual representations of the intercept and each of the feature contribution functions (Figure 2 RHS).

To effectively present the classifier, we display the contribution functions in a meaningful sorted order. Users are often most interested in contributions of features that have the most effect on classification. By displaying the *ranks of evidence*, the explanation system displays all the features in the context of the whole classifier.

The visualization and ranking measure of each contribution function $f_j(x_j)$ is based on its form. If the function is a linear function, $f_j(x_j) = w_j x_j$, we represent the

function by a bar whose length is proportional to w_j (Figure 2 RHS-Age). Such functions can be compared visually and ranked easily (by the value of w_j). If the contribution function is a two-case conditional, it has the form:

$$f_j(x_j) = \begin{cases} v_a & \text{if } x_j \text{ satisfies condition } a \\ v_b & \text{otherwise} \end{cases}$$

We display bars corresponding to each case (Figure 2 RHS-Symptoms). This graphical representation allows visual comparison and simple ranking (by the difference between the two evidence bars). Other function types may require other representations and ranking functions. This capability is limited to additive models with functions that can be represented graphically and ranked meaningfully.

For CAD diagnosis, the *ranks of evidence* (Figure 2 RHS) indicate the overall importance that the model assigns to each patient risk factor. Ideally, the highest ranked features will be the most important overall risk factors and advance the physician's trust in the classifier.

Capability 4 – source of evidence: The *source of evidence* capability assists users to explore the reasoning and data behind classifier parameters. Where possible, this capability represents how the evidence contributions of each feature relate to the training data. If the training data is available (and regardless of whether or not a calculation from that data can be shown), an explanation system can facilitate exploration of the data behind the classifier.

To audit the relationship between the decision label and a feature, the training data is sliced by label value and sliced (or sorted) by feature, allowing users to inspect whether classifier parameters are compatible with their expectations, based on training data. A data summary shows the totals of each slice and allows the user to drill down to the training data. The *source of evidence* capability is especially useful for auditing anomalies in training data that lead to unexpected predictions. For CAD diagnosis, the *source of evidence* capability allows the physician to drill down to original study data as shown in Figure 3 and view how classifier parameters are calculated.

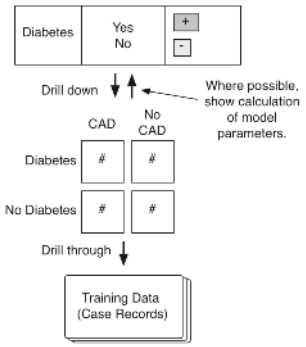


Figure 3: Capability 4 - *source of evidence*.

Applications

The current ExplainD framework can be applied to any linear model. To date, it has been applied to three classification techniques: naïve Bayes (NB), linear support vector machine (SVM), and logistic regression (LR). The NB explanation facility has been deployed, since August 2002, as an integral part of the Proteome Analyst (PA) system (Szafron *et al.* 2004), a web-based application for predicting protein function and protein subcellular location from protein sequence. In the five-month period between October 2005 and March 2006, molecular biologists have used PA to analyze 552,873 proteins. Users view the PA output by examining web pages. ExplainD's importance is illustrated by the fact that 12% of all web pages examined were from the ExplainD sub-system. The percentage of ExplainD web pages varied from 0% to as high as 29% for some users, with 38.6% of PA users accessing ExplainD.

The NB, SVM, and LR explanation facilities have been implemented using ExplainD. The prototype system is implemented (using Python [http://python.org]) with a simple class interface for explainable classifiers. The linear SVM classifier uses the LIBSVM software (Chang and Lin 2001), with weights extracted by the LIBSVM tool (LIBSVM site). The logistic regression classifier uses the BioPython [biopython.org] logistic regression module. Application data is processed from text files having a common format. New classifiers (such as neural networks without hidden layers or linear discriminant analysis) can be incorporated into the framework by creating a Python class that adheres to a simple interface.

We illustrate the application with a simple text classification example from PA. PA predicts a protein's function and/or subcellular location from that protein's amino acid sequence. A preprocessing step involves a database search that maps each sequence to text keywords that describe proteins closely related to the target protein. Therefore, the classification step can be regarded as a classification on text keywords (for details see Szafron *et al.* 2004). Each protein instance is associated with a vector of binary features, each corresponding to a text keyword or phrase (encoded as '+1' if the corresponding keyword is associated with the protein and '-1' if it is not). For the subcellular location classifier used in this paper, our training set contains 3904 protein sequences.

In this paper, we focus on a classifier that predicts whether or not the protein of interest is found in the 'inner membrane' of a bacterial cell. In our example, the keywords associated with the query protein are '*nitrogen fixation*', '*transmembrane*', '*plasmid*', *'plasma membrane*', and '*complete proteome*'. The true label for our example protein is 'inner membrane' (vs. '*not* inner membrane).

Naïve Bayes

The naïve Bayes classifier (NB) is a widely used and effective probabilistic classifier. A NB classification is a simple Bayes classifier that assigns each instance x to the class k with the largest posterior probability:

$$\text{NB classification of } x = \arg\max_k P(k \mid x) = \arg\max_k P(k)P(x \mid k)$$

Since the model assumes independence of a feature given the class label, we can formulate the NB classifier as:

$$\text{NB classification of } x = \arg\max_k P(k) \prod_{j=1}^{m} P(x_j \mid k) \quad (3)$$

Each probability in the model is estimated by counting the relevant occurrences in the training data. Written in the typical form shown in (3), the NB model does not seem to fit a linear or additive model. For a two-class problem, however, it is well known (Duda, Hart, and Stork 2000) that we can use the log-odds ratio of the two classes to represent the classifier as a discriminant.

$$\text{NB classification of } x = \begin{cases} + & \text{if } P(k=+ \mid x) > P(k= \mid x) \\ & \text{otherwise} \end{cases}$$

$$= \begin{cases} + & \text{if } \log \frac{P(k=+ \mid x)}{P(k= \mid x)} > 0 \\ & \text{otherwise} \end{cases}$$

This formulation of the classifier yields the discriminant:

$$g_{NB}(x) = \log \frac{P(k=+ \mid x)}{P(k= \mid x)}$$

$$= \log \frac{P(k=+) \prod_{j=1}^{m} P(x_j \mid k=+)}{P(k=) \prod_{j=1}^{m} P(x_j \mid k=)}$$

$$= \log \frac{P(k=+)}{P(k=)} + \sum_{j=1}^{m} \log \frac{P(x_j \mid k=+)}{P(x_j \mid k=)}$$

This discriminant fits an additive model (2), where:

$$b = \log \frac{P(k=+)}{P(k=)}$$

$$f_j(x_j) = \begin{cases} \log \frac{P(x_j = +1 \mid k=+)}{P(x_j = +1 \mid k=-)} & \text{if } x_j = +1 \\ \log \frac{P(x_j = -1 \mid k=+)}{P(x_j = -1 \mid k=-)} & \text{if } x_j = \ \ 1 \end{cases} \quad (4)$$

ExplainD uses this formulation to explain naïve Bayes classification. The weights of the classifier correspond to Good's 'weights of evidence' (Madigan, Mosurski, and Almond 1996). In addition to evidence contributions, confidence intervals can also be calculated and displayed for a NB classifier (Van Allen, Greiner, and Hooper, 2001; Mozina *et al*. 2004; M. Mozina, personal communication).

Capability 0, the *decision*, can be represented graphically for the NB classifier by using probabilities. Capability 1, *decision evidence*, is visualized as described in Section 2. For both PA (Figure 4) and the prototype system (Figure 5), the user can see that the classifier (correctly) predicts inner membrane as the protein label.

From the display of focus features, we observe that the presence of keywords '*plasma membrane*' and '*transmembrane*' give critical evidence for the decision to label the query protein as '*inner membrane*'. Due to the noisiness of the biological data set, we see that the query protein, contrary to our expectations, is not associated with the keyword '*inner membrane*'. The lack of '*inner membrane*' gives negative evidence for the decision 'inner membrane'. Fortunately, other keywords, give enough evidence to overcome this omitted information.

From Capability 2, *decision speculation* (Figure 6), we observe that the decision would change if the keyword '*transmembrane*' were not associated with this protein. The lack of '*transmembrane*' (new value $x_{transmembrane} = -1$) would give sufficient negative evidence for the 'inner membrane' decision that the negative-response bar would be longer than the positive one. The difference between changing a feature value (from +1 to -1) and ignoring a feature is important. If the feature '*transmembrane*' were ignored in the training and testing instances, it would be equivalent to completely removing the '*transmembrane*' segment from the evidence bars. This follows from the additive nature of the model. In this case, the original positive decision for the inner membrane label would hold.

By viewing the NB classifier's Capability 3, *ranks of evidence*, the user can see the most important features overall for classification of 'inner membrane'. For example, Figure 7 shows that the most important features is*: 'inner membrane'* which contributes f_positive('*inner membrane*') = 5.44 to a positive decision if the feature is present in a query protein and contributes f_negative('*inner membrane*') = 1.77 to a negative decision if it is absent from the query protein.

To capture the rank of this feature we compute a ranking score – the difference between the evidence given by a positive (present) feature value compared with a negative (absent) feature value. For *'inner membrane'* the ranking score is (-1) (-1.77) + (+1) (+5.44) = +7.21. Each feature is ranked by the ranking score of its evidence. The features with second and third ranks of evidence for this classifier are *'cytoplasmic'* with score (-1) (+1.33) + (+1) (-5.35) = -6.68 and *'protein biosynthesis'* with score (-1) (+0.63) + (+1) (-5.59) = -6.62. In fact, *'transmembrane'* is ranked thirteenth, with score (-1) (-2.16) + (+1) (+2.69) = +4.85, but it is important enough (as well as the features ranked above it) to reverse the decision if it did not appear as a feature in the query protein.

Capability 4, the *source of evidence*, can be shown by two methods. For the NB classifier we can display the actual calculation of each evidence contribution at the user's request. For example, Figure 9 shows the computation for the contribution of the '*transmembrane'* feature to a 'cytoplasm' classifier that, similarly to our 'inner membrane' classifier, decides whether or not a protein is in the cytoplasm of a cell. Figure 9 shows that there were 5 instances in the 'cytoplasm' training set that included the feature '*transmembrane'* and were labeled 'cytoplasm' out of a total of 2465 training instances that were labeled 'cytoplasm'. It also shows that there were 706 training instances that included the feature '*transmembrane'* and were labeled 'not cytoplasm' out of a total of 1437 training instances that were labeled 'not cytoplasm'. Given this information, the evidence contribution is calculated based on the NB formula in (4).

If training data is available, we can also show how the instances are sliced by label and feature. For example, Figure 8 shows how the data is sliced for label 'inner membrane' and feature '*transmembrane*'. The user may then continue to drill through to view the actual training data by clicking on the numerical hyperlinks in the table.

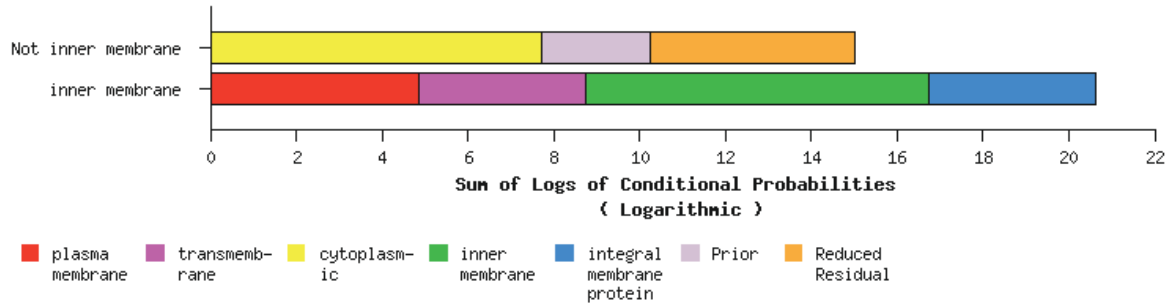
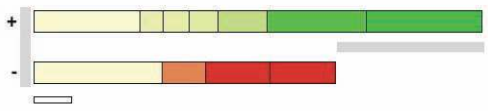


Figure 4: Naive Bayes application of capability 1 – *decision evidence* in Proteome Analyst.

Decision Evidence for 'inner membrane' classification using NaiveBayes



Color	Feature	Value	Contribution
	plasma membrane (SE) (RE)	1	3.14499
	transmembrane (SE) (RE)	1	2.69372
	inner membrane (SE) (RE)	-1	-1.77431
	prior (SE) (RE)	constant	-1.76159
	cytoplasmic (SE) (RE)	-1	1.33019
	integral membrane protein (SE) (RE)	-1	-1.18103
	plasmid (SE) (RE)	1	0.80206
	nitrogen fixation (SE) (RE)	1	0.70888
	protein biosynthesis (SE) (RE)	-1	0.62916
	+ Aggregate (130)	+	2.86909
	- Aggregate (265)	-	-3.47801
	Decision	+	3.98314

Figure 5: Naïve Bayes application of capability 1 – *decision evidence* in the ExplainD prototype.

Decision Evidence for 'inner membrane' classification using NaiveBayes

Color	Feature	Value	Contribution
	plasma membrane (SE) (RE)	1	3.14499
	transmembrane (SE) (RE)	-1	-2.16327
	inner membrane (SE) (RE)	-1	-1.77431

Figure 6: Naïve Bayes application of capability 2 - *decision speculation* in the ExplainD prototype.

Classifier Ranks of Evidence for 'inner membrane'

inner membrane	Training Data sliced by label 'inner membrane' and feature 'inner membrane'.		
	-1	= -1.77431	
	1	= 5.44131	
cytoplasmic	Training Data sliced by label 'inner membrane' and feature 'cytoplasmic'.		
	-1	= 1.33019	
	1	= -5.35363	
protein biosynthesis	Training Data sliced by label 'inner membrane' and feature 'protein biosynthesis'.		
	-1	= 0.62916	
	1	= -5.59310	
light-harvesting polypeptide	Training Data sliced by label 'inner membrane' and feature 'light-harvesting polypeptide'.		
	-1	= -0.11596	
	1	= 5.90183	
bacteriochlorophyll	Training Data sliced by label 'inner membrane' and feature 'bacteriochlorophyll'.		
	-1	= -0.11596	
	1	= 5.90183	
antenna complex	Training Data sliced by label 'inner membrane' and feature 'antenna complex'.		
	-1	= -0.11596	
	1	= 5.90183	
thylakoid membrane	Training Data sliced by label 'inner membrane' and feature 'thylakoid membrane'.		
	-1	= -0.10235	
	1	= 5.78405	
	Training Data sliced by label 'inner membrane' and feature 'thylakoid'.		

Figure 7 Naïve Bayes application of capability 3 - *ranks of evidence* in the ExplainD prototype.

Source of Evidence: Training Instances sliced by 'inner membrane' and 'transmembrane'

	transmembrane	not transmembrane
inner membrane	511	61
not inner membrane	200	3130

Figure 8: Naïve Bayes application of capability 4 – *source of evidence* in the ExplainD prototype (showing only the relevant slices of the training data).

Color	Feature	Value	Contribution	Detail
	transmembrane (SE) (RE)	1	-5.30833	= log (P('transmembrane' | cytoplasm)) / (P('transmembrane' | not cytoplasm)) = log (c('transmembrane',cytoplasm)/c(cytoplasm)) / (c('transmembrane',not cytoplasm)/c(not cytoplasm)) = log (5/2465) / (706/1437) = log (0.00243) / (0.49131) = log 0.00495 = -5.30833

Figure 9: Naïve Bayes application of capability 4 – *source of evidence* in the ExplainD prototype (showing detailed calculation of the parameters).

Support Vector Machines (SVM)

Support vector machines (SVMs) are an important type of classifier based on linear discriminants. SVMs are often considered a 'black box' since their formulation is less intuitive for those not familiar with machine learning. However, the primal formulation of the linear SVM decision function is a linear discriminant (Vapnik 1995):

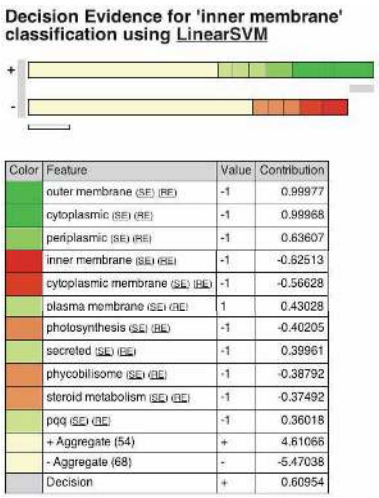
$$y = sign \left(\sum_{j=1}^{m} w_j x_j \right) - b$$

Thus, ExplainD applies to an SVM classifier of this form.

The SVM model we present does not take advantage of the dual formulation, and thus cannot use the 'kernel trick' (Boser, Guyon, and Vapnik 1992) that allows for more complex representations of the input data (radial basis functions, polynomial expansions, etc.). It is currently unclear how ExplainD could be used with kernels to explain the predictions of more complex SVM classifiers.

For our prototype, Capability 0 indicates only the class. It would be helpful to indicate prediction strength – corresponding to the probabilistic output of a Naïve Bayes classifier. Unfortunately, SVM classifiers do not normally produce probabilities. However, the distance of an instance from the decision boundary in an SVM is related to how confident the corresponding prediction is. Therefore, the output of an SVM can be mapped into a posterior probability, where the SVM is more confident of predictions that lie farther from the decision boundary. This is accomplished by fitting a sigmoid function to the training data, which maps the SVM output into the [0,1] range (Platt 2000). An explanation system could display these probabilities to improve the user's understanding of how confident the SVM is in its predictions and make the decision visualization more informative. Our prototype does not yet support this approach. However, the SVM classified the protein (correctly) as 'inner membrane'. Capability 1, *decision evidence*, shows that although the linear SVM made the same decision as the NB classifier, the most important evidence for that decision was not the same (compare Figure 10 with Figure 5).

In contrast with the NB classifier, the SVM classifier does not change the classification decision if the keyword '*transmembrane*' is removed as a feature using *decision speculation* (not shown). This is due to the fact that 'transmembrane' is ranked very low in the ranks of evidence for the SVM classifier. The *ranks of evidence* for the SVM classifier are represented as continuous linear functions and are ranked by the absolute value of the weight w_j. The ranks from the SVM classifier show a markedly different set of high-ranked features compared to the NB classifier (Figure 11 and Figure 7). For example, the feature *'inner membrane'* has dropped from first to fourth rank. This is partly because the SVM classifier depends on the absence of the feature *'outer membrane'* to make a positive prediction for 'inner membrane'.



Color	Feature	Value	Contribution
	outer membrane (SE) (RE)	-1	0.99977
	cytoplasmic (SE) (RE)	-1	0.99968
	periplasmic (SE) (RE)	-1	0.63607
	inner membrane (SE) (RE)	-1	-0.62513
	cytoplasmic membrane (SE) (RE)	-1	-0.56628
	plasma membrane (SE) (RE)	1	0.43028
	photosynthesis (SE) (RE)	-1	-0.40205
	secreted (SE) (RE)	-1	0.39961
	phycobilisome (SE) (RE)	-1	-0.38792
	steroid metabolism (SE) (RE)	-1	-0.37492
	pqq (SE) (RE)	-1	0.36018
	+ Aggregate (54)	+	4.61066
	- Aggregate (68)	-	-5.47038
	Decision	+	0.60954

Figure 10: Linear SVM application of capability 1 – *decision evidence* in the ExplainD prototype.

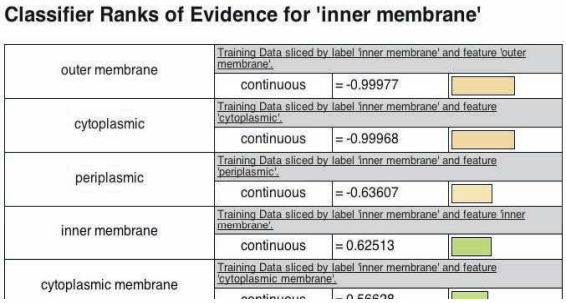


Figure 11: Linear SVM application of capability 3 – *ranks of evidence* in the ExplainD prototype.

For capability 4, *source of evidence*, the SVM is not as intuitive as the NB model. While we can still slice the training data using features and labels, we cannot show the direct calculation between the training instances and the evidence parameters, w_j and b, since their values depend on the global training set in a complex manner.

This example highlights that despite commonalities in the explanation system, NB and SVM classifiers can remain different in their classification decisions and supporting evidence. This example shows that a good explanation system can be used to reveal and highlight the differences between such classifiers.

Logistic Regression

Logistic regression classifiers model the probabilities of the potential class with the logistic function. Due to their simplicity and the fact that many real world phenomena are logistic in nature, they are widely used for making binary decisions (*positive* or *negative*). The response variable is specified with a log-odds transformation—the log of the odds-ratio of the *positive* outcome against the *negative* (Alpaydin 2004; Hastie, Tibshirani, and Friedman 2001).

$$g(x) = \log \frac{P(y=+\mid x)}{P(y=\mid x)} = b + \sum_{j=1}^{m} w_j x_j$$

This classifier form fits our explanation framework (b is the intercept and w_j are weights). The *decision* is expressed as a probability, as with the NB classifier. However, all other explanation capabilities are displayed in a manner similar to the SVM framework. The feature contributions are continuous (as opposed to a choice of discrete values). As with the SVM classifier, we cannot represent a direct calculation of the weights from the training data since this is done through optimization (such as the Newton-Raphson algorithm used by ExplainD). Despite this complication, once training has been done and the weights have been obtained, we can still allow the user to drill through to the training data. Our CAD example is based on a model that was obtained using logistic regression (Morise *et al.* 1992) and is included in our prototype explanation system.

Conclusion

We described a framework for visually explaining the decisions of several widely used machine-learned classifiers and the evidence for those decisions. We described how to apply the framework to any classifier that is formulated as an additive model and showed (using our prototype) how to implement this framework in the context of three models: naïve Bayes, linear SVM and logistic regression classifiers. In addition, we have implemented these ideas within Proteome Analyst (Szafron *et al.* 2004), a working bioinformatics application that has been used by hundreds of biologists. This explanation facility has helped several users to find errors in their training data.

This framework offers both experienced and inexperienced users a straightforward graphical explanation of classification. These transparent explanation capabilities can cultivate user confidence in the classifier decision, enhance the user's understanding of the relationships between the feature values and the decisions, and help users to visually audit the classifier and identify suspicious training data. This framework extends the elements provided by other explanation systems (Madigan, Mosurski, and Almond 1996; Becker, Kohavi, and Sommerfield 1997; Mozina *et al.* 2004) that have been used for Bayesian classifiers and provides the potential for extension to many other classifier models.

References

Alpaydin, E. 2004. *Introduction to Machine Learning*. Cambridge, MA: MIT Press.

Becker, B., Kohavi, R., and Sommerfield, D. 1997. Visualizing the Simple Bayesian Classifier. In *KDD Workshop on Issues in the Integration of Data Mining and Data Visualization*.

Boser, B., Guyon, I., and Vapnik, V. 1992. A Training Algorithm for Optimal Margin Classifiers. *Computational Learning Theory*, 144-152.

Chang, C., and Lin, C. 2001. LIBSVM: library for Support Vector Machines, www.csie.ntu.edu.tw/~cjlin/libsvm.

Dhillon, I., Mallela, S., and Kumar, R. 2002. Enhanced Word Clustering for Hierarchical Text Classification. In *8th ACM SIGKDD*, Edmonton, AB, 191-200.

Duda, R., Hart, P., and Stork, D. 2000. *Pattern Classification*, Wiley Interscience.

Fawcett, T., and Provost, F.J. 1997. Adaptive Fraud Detection. *Data Mining and Knowledge Discovery* 1(3): 291-316.

Fawcett, T., and Provost, F.J. 1999. Activity Monitoring: Noticing Interesting Changes in Behavior. In *5th ACM SIGKDD*, San Diego, CA, 53-62.

Gibbons, R. 1997. ACC/AHA Guidelines for Exercise Testing. *J. Amer. College of Cardiology*, 30(1): 260-315.

Hastie, T., Tibshirani, R., and Friedman, J. 2001. *The Elements of Statistical Learning*. Springer.

Joachims, T. 2002. Optimizing Search Engines Using Clickthrough Data. In *8th ACM SIGKDD*, Edmonton, AB.

Kononenko, I. 2001. Machine Learning for Medical Diagnosis: History, State of the Art and Perspective. *Artificial Intelligence in Medicine*, 23(1): 89-109.

Madigan, D., Mosurski, K., and Almond, R. 1996. Explanation in Belief Networks. *J. Comp. and Graphical Stat.* 6:160–181.

Morise, A.P., Detrano, R., Bobbio, M., and Diamond, G.A. 1992. Development and validation of a logistic regression-derived algorithm for estimating the incremental probability of coronary artery disease before and after exercise testing. *J. Amer. College of Cardiology*, 20(5):1187-1196.

Mozinam M., Demsar, J., Kattan, M., and Zupan, B. 2004. Nomograms for Visualization of Naive Bayesian Classifier. In *PKDD-2004*, Pisa, Italy, 337-348.

Platt, J. 2000. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. *Advances in Large Margin Classifiers*, A. Smola, P. Bartlett, B. Scholkopf, and D. Schuurmans, Eds. Cambridge, MA: MIT Press.

Ridgeway, G., Madigan, D., and Richardson, T. 1998. Interpretable Boosted Naïve Bayes Classification. In *4th ACM SIGKDD*, 101-104.

Swartout, W. 1983. XPLAIN: A system for Creating and Explaining Expert Consulting Programs. *Artificial Intelligence*. 21:285-325.

Szafron, D., Lu, P., Greiner, R., Wishart, D.S., Poulin, B., Eisner, R., Lu, Z., Anvik, J., Macdonell, C., Fyshe, A., and Meeuwis, D. 2004. Proteome Analyst: Custom Predictions with Explanations in a Web-based Tool for High-Throughput Proteome Annotations. Nucleic Acids Research, 32:W365-W371.

Vapnik, V. 1995. *The Nature of Statistical Learning Theory*, Springer.

Van Allen, T., Greiner, R., and Hooper, P. 2001. Bayesian Error-Bars for Belief Net Inference. In *17th Conference on Uncertainty in Artificial Intelligence*, Seattle, WA.