MIS Quarterly

RESEARCH ARTICLE

EXPLAINING DATA-DRIVEN DOCUMENT CLASSIFICATIONS[1]

David Martens

Department of Engineering Management, Faculty of Applied Economics, University of Antwerp, Prinsstraat 13, 2018 Antwerp, BELGIUM {david.martens@ua.ac.be}

Foster Provost

Department of Information, Operations and Management Sciences, Stern School of Business, New York University, 44 West 4th Street, New York, NY 10012-1126 U.S.A. {fprovost@stern.nyu.edu}

Many document classification applications require human understanding of the reasons for data-driven classification decisions by managers, client-facing employees, and the technical team. Predictive models treat documents as data to be classified, and document data are characterized by very high dimensionality, often with tens of thousands to millions of variables (words). Unfortunately, due to the high dimensionality, understanding the decisions made by document classifiers is very difficult. This paper begins by extending the most relevant prior theoretical model of explanations for intelligent systems to account for some missing elements. The main theoretical contribution is the definition of a new sort of explanation as a minimal set of words (terms, generally), such that removing all words within this set from the document changes the predicted class from the class of interest. We present an algorithm to find such explanations, as well as a framework to assess such an algorithm's performance. We demonstrate the value of the new approach with a case study from a real-world document classification task: classifying web pages as containing objectionable content, with the goal of allowing advertisers to choose not to have their ads appear on those pages. A second empirical demonstration on news-story topic classification shows the explanations to be concise and document-specific, and to be capable of providing understanding of the exact reasons for the classification decisions, of the workings of the classification models, and of the business application itself. We also illustrate how explaining the classifications of documents can help to improve data quality and model performance.

Keywords: Document classification, instance level explanation, text mining, comprehensibility

Introduction

Document classification systems classify text documents automatically, based on the words, phrases, and word combinations therein (hereafter, words). Business applications of document classification are becoming increasingly widespread, especially with the introduction of low-cost micro-outsourcing systems for annotating training corpora. Prevalent applications include sentiment analysis (Pang and Lee 2008), spam identification (Attenberg et al. 2009), web page classification (Qi and Davison 2009), legal document classification (Tseng et al. 2007), medical document triage (Wallace et al. 2010), and document classification for topical web search (Pant and Srinivasan 2005), just to name a few. Classification models are built from labeled data sets that encode the frequencies of the words in the documents. Importantly for this paper, and different from many data mining

[1]Vijay Vaishnavi was the accepting senior editor for this paper. Vijay Khatri served as the associate editor.

The implementation of the methods described in this paper is available at www.applieddatamining.com.

The appendices for this paper are located in the "Online Supplements" section of the *MIS Quarterly*'s website (http://www.misq.org).

applications, the document classification data representation has very high dimensionality, with the number of words and phrases typically ranging from tens of thousands to millions.

The main contribution of this paper is to examine in detail an important aspect of the business application of document classification that has received little attention in the research literature. Specifically, organizations often need to understand the exact reasons why classification models make particular decisions. The need comes from various perspectives, including those of managers, customer-facing employees, and the technical team. To understand these needs more deeply, in the next section we extend an existing theoretical model from the Information Systems literature to include these various perspectives.

As a concrete illustration, consider an application that has received substantial interest in online advertising: keeping ads off of objectionable web content (eMarketer 2010). Having invested substantially in their brands, firms cite the potential to appear adjacent to nasty content as the primary reason they do not spend more on online advertising. To help reduce the risk, document classifiers are applied to web pages along various dimensions of objectionability, including adult content, hate speech, violence, drugs, bomb-making, and many others. However, because the online advertising ecosystem supports the economic interests of both advertisers *and* content publishers, black-box models are insufficient. Managers cannot put models into production that might block advertising from substantial numbers of non-objectionable pages without understanding the risks of incorporating them into the product offering. Customer-facing employees need to explain why particular pages were deemed objectionable by the models. And the technical team needs to understand the exact reasons for the classifications made, so that they can address errors and continuously improve the models.

Popular techniques to build document classification models include naive Bayes, linear and nonlinear support vector machines (SVMs), classification-tree based methods (often used in ensembles, such as with boosting; Schapire and Singer 2000), K-nearest neighbor (Han et al. 2001), and many others (Hotho et al. 2005). Because of the massive dimensionality, even for linear and tree-based models, it is very difficult to understand exactly how a given model classifies documents. It is essentially impossible for a nonlinear SVM or an ensemble of trees. Understanding the classifications requires *concise* explanations—explanations that refer to only a very small fraction of the total vocabulary—in contrast to existing explanation approaches, which in most cases include large fractions of the vocabulary.

Understanding particular classifications also provides other important benefits. Along with improved understanding of the classification model, the explanations also can provide a novel lens into the complexity of the business domain. For example, in Explanation 1 (shown below; described fully in the section "Instance-Level Explanations" later in this article), the word "welcome" as an indication of adult content initially seems strange. Upon investigation/reflection we understand that in some cases an adult website's first page contains a phrase similar to "*Welcome to....By continuing you confirm you are an adult and agree with our policy*." The explanation brings this complexity to light.

Explanation 1: An example explanation why a web page is classified as having adult content.

If words (welcome fiction erotic enter bdsm adult) are removed, then class changes fro adult to non-adult.

We introduce this problem, tying it in to the existing literature on explanations for decision systems and extending the relevant theory to account for modern, data-driven modeling. In line with this theory, we then introduce the first (to our knowledge) technique that directly addresses the explanation of the decisions made by document classifiers. The technique focuses on explaining why a document is classified as a specific class of interest (e.g., "objectionable content" or "hate speech"). Finally, we present a case study based on data from a real application to the business problem of safe advertising discussed above, and an empirical follow-up study on benchmark data sets (from news classification). These studies demonstrate that the methods can be effective, and also flush out additional important issues in explaining document classifications, such as the need for hyper-explanations.

Explanations and Statistical Classification Models

Explaining the decisions made by intelligent decision systems has received both practical and research attention for decades, and a complete review is well beyond the scope of this paper. Nonetheless, there are important results from prior work that help to frame, motivate, and explain the specific gap in the current state of the art that this paper addresses.

Model-Based Decision Systems and Instance-Specific Explanations

Starting as early as the celebrated MYCIN project in the 1970s studying intelligent systems for infectious disease diagnosis (Buchanan and Shortliffe 1984), the ability for intelligent systems to explain their decisions was understood to be necessary for effective use of such systems and therefore was studied explicitly. The document classification systems that are the subject of this paper are an instance of decision systems (DSs): systems that either (1) support and improve human decision making (as with the characterization of decision-support systems by Arnott 2006), or (2) make decisions automatically. The focal application of this paper's case study falls in the second category: billions of attempts to place advertisements are made each day, and each decision is made in a couple dozen milliseconds. Model-based decision systems have seen a steep increase in development and use over the past two decades (Banker and Kauffman 2004). We focus on models produced by large-scale automated statistical predictive modeling systems (Shmueli and Koppius 2011), for which generating explanations can be particularly problematic.

Different applications impose different requirements for understanding. Consider three different application scenarios, both to add clarity in what follows, and so that we can rule out one of them. First, in some applications it is important to understand every decision that the DS *may possibly make*. For example, for many applications of credit scoring (Martens et al. 2007), regulatory requirements stipulate that every decision be justifiable, and often this is required in advance of the official "acceptance" and implementation of the system. Similarly, one could easily see that a medical decision system may need to be completely transparent in this respect. The present paper, about individual case-specific explanations, is not intended to apply to systems such as these.[2]

In contrast, consider applications where one needs to explain the specific reasons for some subset of the individual decisions (cf., the theoretical reasons for explanations summarized by Gregor and Benbasat (1999), discussed below). Our case study falls into this category. Often, this need for individual case explanations arises because particular decisions need to be justified after the fact. For example, a customer may question a decision or a developer may want to examine model performance on historical cases. Furthermore, to reveal *problems* with the classification of documents it may be more efficient for an analyst to study concise explanations than the documents themselves. Alternatively, a developer may be exploring decision-making performance by giving the system a set of theoretical test cases. In both scenarios, it is necessary for the system to provide explanations for specific individual cases.[3] Other examples in the second scenario include fraud detection (Fawcett and Provost 1997), many cases of targeted marketing, and all of the document classification applications listed above.

In a third application scenario, every decision that the system actually makes must be understood. This often is the case with a classical decision-support system, where the system is aiding a human decision maker, for example for forecasting (Gönül et al. 2006) or auditing (Ye and Johnson 1995). For such systems, again, it is necessary to have case-specific explanations.

Cognitive Perspectives on Model Explanations

Gregor and Benbasat (1999) provide a survey of empirical work on explanations from intelligent systems. They find that explanations are important to users when there is some specific reason and anticipated benefit, when an anomaly is perceived, or when there is an aim of learning. Their theoretical analysis brings to the fore three ideas that are critical for our context. First, they introduce the reasons for explanations: to resolve perceived anomalies, a need to better grasp the inner workings of the intelligent system, or the desire for long-term learning. Second, they describe the type of explanations that should be provided: they emphasize the need not just for general explanations of the model, but for explanations that are context-specific. Third, Gregor and Benbasat emphasize the need for justification-type explanations, which provide a justification for moving from the grounds to the claims, in contrast to rule-trace explanations. In statistical predictive modeling, the rule trace often entails simply the application of a mathematical function to the case data, with the result being a score representing the likelihood of the case belonging to the class of interest, with no justification of why. There is little existing work on methods for explaining modern statistical models extracted from data that satisfy these latter two criteria, and none (to our knowledge)

[2]The current prevailing interpretation of this requirement for complete transparency argues for a globally comprehensible predictive model. Indeed, in credit scoring generally, the only models that are accepted are linear models with a small number of well-understood, intuitive variables. Such models are chosen even when nonlinear alternatives are shown to give better predictive performance (Martens et al. 2007).

[3]Individual case-specific explanations may also be sufficient in many applications. For this paper, it is only important that they be necessary.

that provide such explanations for the very high-dimensional models that are the focus of this paper.

An important subtlety that is not brought out explicitly by Gregor and Benbasat, but which is quite important in our context, is the difference between (1) an explanation as intended to help the user to understand how *the world* works, and thereby help with acceptance of the system, and (2) an explanation of how *the model* works. The latter case can be further subdivided into (a) how the model works in general, and (b) how the model works on a particular instance. The explanation thereby either can help with acceptance, or can focus attention on the need for improving the model. When the model reflects reality, then this also will support understanding how the world works.

The Three-Gap Framework

In order to examine more carefully why explanations are needed and their impact on decision model understanding, long-term learning, and improved decision making, we turn to the recent work by Kayande et al. (2009). This work focuses on the same context as we do in our case study, specifically where data are voluminous, the link between decisions and outcomes is probabilistic, and the decisions are repetitive. They presume that it is highly unlikely that decision makers can consistently outperform model-based DSs in such contexts.

Prior work has suggested that when users do not understand the workings of the DS model, they will be skeptical and reluctant to use the model, even if the model is known to improve decision performance (see, for example, Arnold et al. 2006; Kayande et al. 2009; Lilien et al. 2004; Limayem and DeSanctis 2000; Umanath and Vessey 1994). Further, decision makers need impetus to change their decision strategies (Todd and Benbasat 1999), as well as guidance in making decisions (Silver 1991). Kayande et al. introduce a "three-gap" framework (Figure 1) for understanding the use of explanations to improve decision making by aligning three different models: the user's model, the system's model, and reality. Their results show that guidance toward improved understanding of decisions, combined with feedback on the potential improvement achievable by the model, induces decision makers to align their mental models more closely with the decision model, leading to deep learning. This alignment reduces the corresponding gap (Gap 1), which in turn improves user evaluations of the DS. It is intuitive to argue that this then improves acceptance and increases use of the system. Under the authors' assumption that the DS's model is objectively better than the decision maker's (large Gap 3 compared to Gap 2), this then would lead to improved decision-making performance (see Todd and Benbasat 1999). Expectancy theory suggests that this will lead to higher usage and acceptance of the DS model, as users will be more motivated to actually use the DS if they believe that a greater usage will lead to better performance (DeSanctis 1983).

An Extended Gap Framework

The framework of Kayande et al. is incomplete in two important ways, which we now will address in turn. First, Kayande et al. do not address the use of explanations (or other feedback) to improve the DS model. Technically this incompleteness is not an incompleteness in their three-gap framework, because improving the model fits as closing Gap 2. Indeed, the authors note specifically that "to provide high-quality decision support, the gap between the DSS model and the true model must be small (Gap 2…)" (p. 529). However, in their paper, Kayande et al. focus their attention on closing Gap 1 between the user's mental model and the DS model. They justify this with the explicit assumption "that the DSS model is of high objective quality (small Gap 2) and that it is of better quality than the user's mental model (large Gap 3)" (p. 529). Even when the model's performance generally is much better than the user's, in many applications there still are plenty of cases where the user is correct when the model is wrong. True mistakes of the model, when noticed by a user, can jeopardize user trust and acceptance.

Generally, we need research that focuses on a user-centric theoretical understanding of the production of explanations with a primary goal of improving data-driven models based on feedback and iterative development. This is important because as model-based systems increasingly are built by mining models from large data, users may have much less confidence in the model's reasoning than with hand-crafted knowledge-based systems. There are likely to be many cases where the decisions are erroneous due either to biases in the process, or to over-fitting the training data (Hastie et al. 2001). As pointed out by Gregor and Benbasat, a user will want an explanation when she perceives an anomaly. The resultant explanation may help the user to learn about how the world works (Kayande et al. 2009), and thereby improve acceptance. However, it alternatively may lead to the identification of a flaw in the model, and lead to a development effort focused on improving the model. At a higher level, this ability for the users and the developers to collaborate on fixing problems with the system's decision making may also improve user acceptance, because the user sees herself as an active, integral part of the system development, rather than a passive recipient of explanations as to why she is wrong about

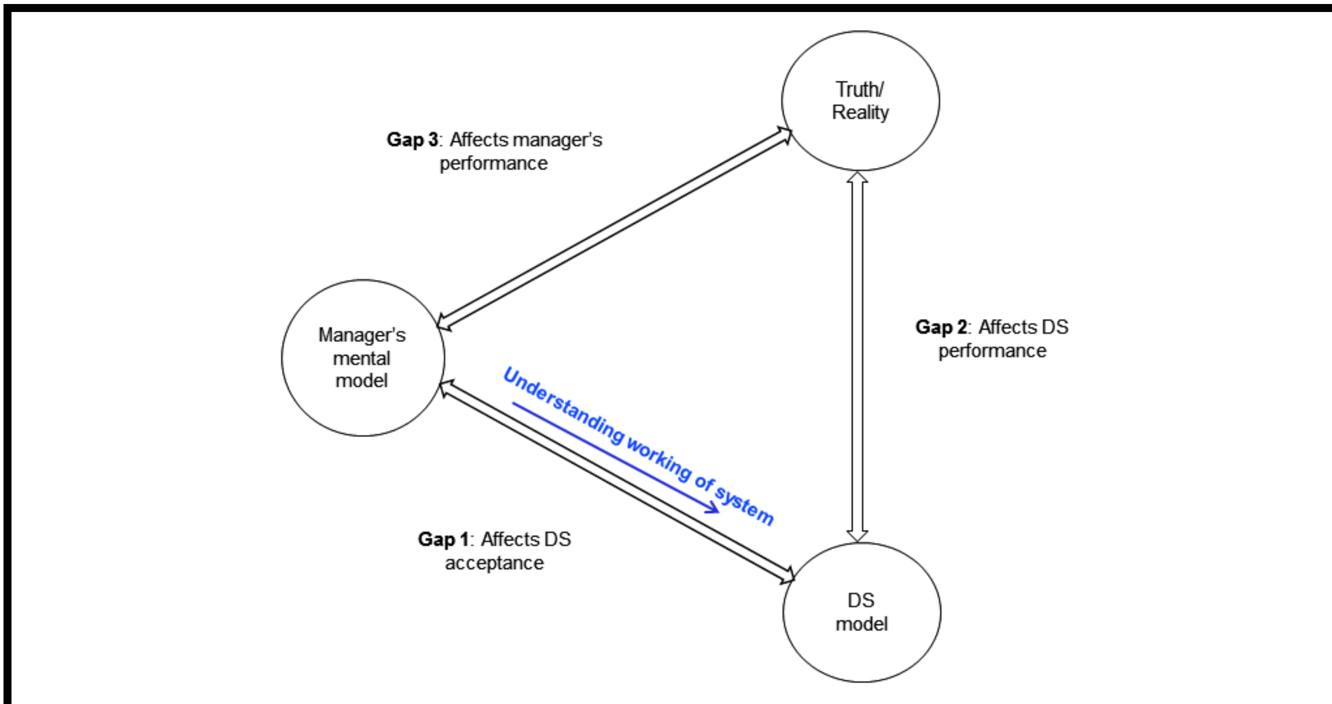


Figure 1. The Three-Gap Framework (adapted from Figure 1 in "How Incorporating Feedback Mechanisms in a DSS Affects DSS Evaluations," U. Kayande, A. De Bruyn, G. L. Lilien, A. Rangaswamy, and G. H. Van Bruggen, *Information Systems Research* (20:4), 2009)

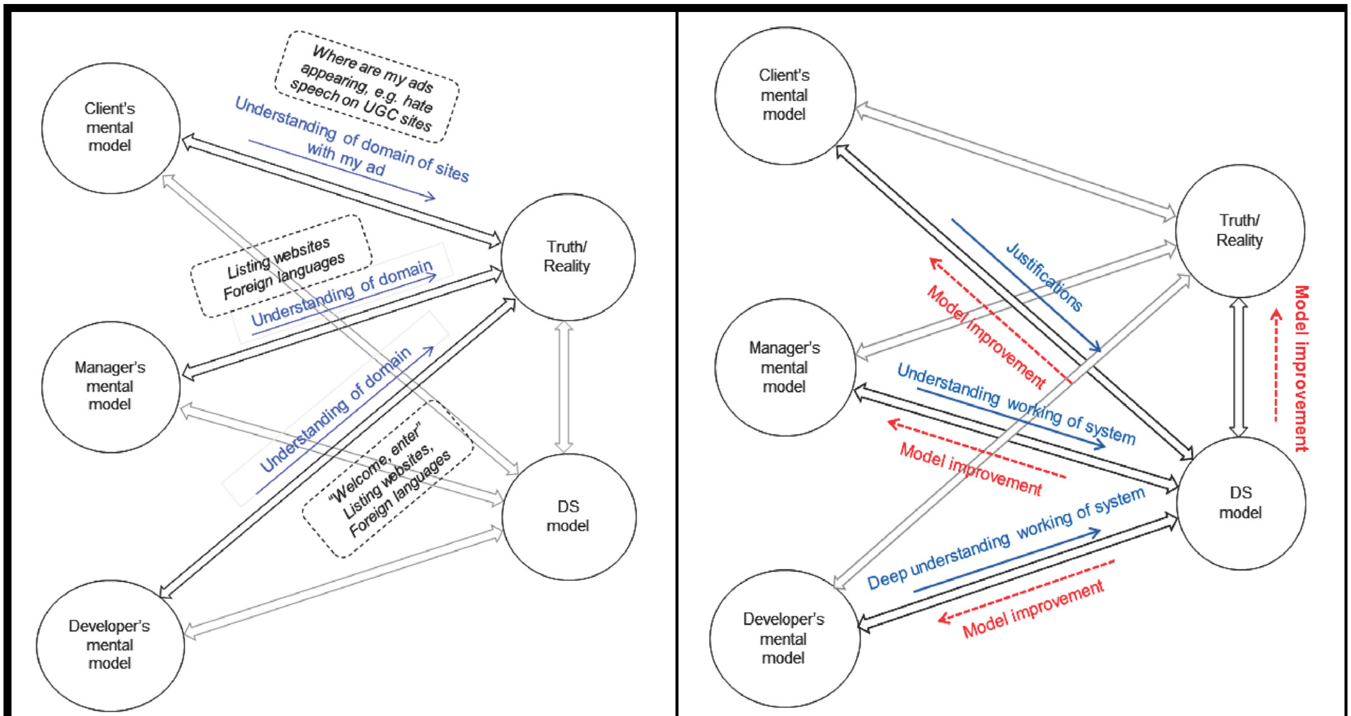
the world. *Therefore, our first extension to the three-gap framework is that* ***explanations*** *can be used to improve the model—closing Gap 2 (and Gap 1 in the other direction)—as well as to improve user understanding.*

This leads us to the second important incompleteness in the framework of Kayande et al. The three-gap framework considers a single, monolithic "user" of the decision system. We contend that to better understand the uses of explanations in the context of practices within contemporary organizations, we need to differentiate between different roles of people who interact with the decision system.[4] In order to understand how explanations are or should be used, there are at least three different roles that are important to distinguish: developers, managers, and customers.

Figures 2a and 2b present a seven-gap extension to Kayande et al.'s framework. The extended framework makes three novel contributions. First, it clarifies the bidirectional nature of the gap closing that can be achieved via explanations: explanations can lead to changes in user mental models; they also can lead to changes in the DS model. Second, the extended framework distinguishes three different user roles. Each different role has different needs and uses for explanations, as will be illustrated in the context of our case study. Third, the extended framework distinguishes between two quite different sorts of user understanding, both of which are important: understanding reality better, and understanding the DS model better.

Specifically, Figure 2a illustrates how the extended model breaks apart the closing of the gap between the different user roles and reality. In each case, explanations can give the user better understanding of the domain. However, although customers, managers, and developers all need to accept the DS model, *acceptance* means different things for each. In our case study application of web page classification for safe advertising, explanations of why ads are blocked on certain pages can increase a *customer*'s understanding of the sorts of pages on which her ads are being shown (a difficult task in modern online display advertising). If these include hate speech pages on user-generated content sites, this may substantially increase the user's acceptance of the need in the first place for the DS. *Managers* seeing explanations of blocked

[4]We discuss different roles rather than different sorts of people, because in some contexts the same person may play more than one of the roles.



(a) Proposed seven-gap model highlighting the gaps between the users' models and reality. Understanding document classifications can close these gaps, helping users to understand the world better, thereby improving acceptance of the system.

(b) Proposed seven-gap model highlighting the gaps between the users' models and the DS's model. These gaps can be closed in either direction: improving users' understanding of how the DS model works, or helping to improve the DS model. Improving the DS model, in turn, helps close the vertical gap between the DS model and reality.

Figure 2. Seven-Gap Extension of Kayande et al's Three-Gap Framework (The figure shows that (1) explanations can close more than just the gap between the user's mental model and the DS model, and (2) the extension of a single user to three relevant user roles: client, manager, and developer.)

pages can better understand the landscape of objectionable content, in order to better market the service. *Developers* can better understand the need for focused data collection, in order to ensure adequate training data for the classification problems faced (Attenberg et al. 2011; Attenberg and Provost 2010). In sum, assuming (as do Kayande et al.) that the DS model is relatively close to reality, a better understanding of the domain should improve: acceptance by customers and managers, marketing and sales by managers, and efficiency and efficacy of developers.

Figure 2b highlights the gaps between the users' mental models and the DS model. The arrows moving from the mental models toward the DS model break apart different sorts of understanding that underlie the gap closing that explanations may provide, inherent in the treatment by Kayande et al. In the case of data-driven statistical models, all of the different user roles may need to achieve some level of understanding of the decision system in order to improve acceptance (in line with prior research). At the top of the figure, clients/customers may need to have specific decisions of the system justified. As represented by the middle gap, managers need to understand the workings of the DS model: customer-relationship managers need to deal with customer queries regarding how decisions are made. Even in applications for which black-box systems are deployed routinely, such as fraud detection (Fawcett and Provost 1997), *managers* still need to have confidence in the operation of the system (middle gap) and may need to explain to customers reasons for particular classifications when errors are made. Operations managers need to "sign off" on models being placed into production. Such managers prefer to understand how the model makes its decisions, rather than simply to trust the technical/data science team. Development managers need to understand specific decisions when they are called into question by customers or business-side employees. Finally, (bottom gap) the data science developers themselves need to understand the reasons for decisions in order to be able to debug/improve the models (discussed next). Holistic views of a model and aggregate statistics across a test set may not give sufficient guidance as to what exactly is wrong and how the model can and should be improved.

The dashed arrows (emanating from the DS model) represent gap-closing in the other direction, by *improving the DS model*. The explanation methods introduced in this paper can have a substantial impact on improving document classification models from the users' perspectives. Despite the stated goals of early research on data mining and knowledge discovery (Fayyad et al. 1996), very little work has addressed support for the process of building acceptable models, especially in business situations where various parties must be satisfied with the results. There has been increasing focus in research and in practice on using advanced statistical models that mimic behavior, without understanding the meaning of those behaviors (Norvig 2011). The design we introduce provides support for such understanding. The DS model can move closer to the mental models of people playing each of the different user roles, to the extent that they were correct on the specific flaws that were improved upon. Presumably these gap closings also would improve acceptance. Possibly equally important for acceptance would be the increase in the users' perception that the model *can be* improved when necessary.

Note that, when improved, the model is likely also to move closer to reality (the vertical, dashed arrow). We say "is likely to" because since there is a gap between each user's mental model and reality, it may be that moving the model closer to the mental model of some user actually moves it further away from reality. We will not examine that possibility in this paper.[5] The extended gap model also highlights the existence of the vertical gaps between user roles. Closing these gaps also is important to DS development (see, for example, Barki and Hartwick 2001; Sambamurthy and Poole 1992). For example, to avoid conflicts, managers and developers should have similar mental models. Producing good explanations may address these gaps indirectly, as closing the gaps between the user roles and reality and between the user roles and the DS model may act naturally to close these vertical gaps between user mental models. We do not address these vertical gaps directly in this paper.

[5]We have omitted the possibility that reality can move closer to the DS model in our treatment. However, this is not necessarily out of the question. The "true" classifications of documents are subjective in certain domains, and it may be that a broadly used classification system changes the accepted subjective class definitions. Further, in dynamic domains the production of documents may coevolve with system development and usage. Authors may write documents differently based on their knowledge of the algorithms used to find or process them. Such issues are beyond the scope of this paper.

Explaining Documents' Classifications

Prior research has examined two different sorts of explanation procedures for understanding predictive models: global explanation and instance-level explanation (Baehrens et al. 2010; Craven and Shavlik 1997; Martens et al. 2007; Robnik-Sikonja and Kononenko 2008; Štrumbelj and Kononenko 2010; Štrumbelj et al. 2009). Global explanations provide improved understanding of the complete model, and its performance over the entire space of possible instances. Instance-level explanations provide explanations for the model's classification of an individual instance.

In the previous section, we presented reasons for preferring instance-level explanations over global explanations, drawing on prior IS research. We now present additional reasons why existing methods are not ideal (or not suitable) for explaining classifications of documents in particular, and then we present a new approach that addresses the drawbacks.

Key Aspects of Document Classification

We focus on textual document classification, where a score is produced representing the predicted likelihood (or strength of belief) of the document belonging to some discrete class or category, based on the values of a large number of independent variables representing the words.[6] There are several ways in which document classification differs from traditional data mining for common applications such as credit scoring, medical diagnosis, fraud detection, churn prediction, and response modeling. First, the data instances have less structure. Technically, one can engineer a feature-vector representation from the sequence or bag of words, but this leads us to our second main difference. Second, in a feature-vector representation of a document data set, the number of variables is often orders of magnitude larger than in the standard classification problems presented above. Third, the values of the variables in a text mining data set denote the presence, frequency of occurrence, or some positively weighted frequency of occurrence of the corresponding word (see below).

[6]Technically, text document classification applications generally use terms that include not only individual words, but phrases, metadata terms, n-grams, etc. For this paper, we call all of these words. Cases where the terms are not comprehensible to a human present a limitation of our approach.

These three aspects of document classification all are critical for the explanation of classifier decisions. The first two combine to render existing explanation approaches relatively useless (as we discuss in detail next). The third, however, presents the basis for the design of the solution we propose. Specifically, with all such document classification representations, removing words always corresponds to reducing the value of the corresponding variable or setting it to zero.

A few technical details of document classification are important here. All non-textual symbols, such as punctuation, are removed from each document, unless they are specifically included for their semantic relationship to the classification task. For a set of n documents and a vocabulary of m words, an $n \times m$ data set is created with the value tf_{ij} on row i and column j denoting the frequency of word j in document i (term frequency). As most of the words available in the vocabulary will not be present in any given document, most values will be zero, and a sparse representation typically is used. Often a weighting scheme is applied to the frequencies, where the weights reflect the importance of the word for the specific application (Hotho et al. 2005). A commonly used data-driven weighting scheme is *tfidf*: $x_{ij} = \mathrm{tf}_{ij} \times \mathrm{idf}_j$ where the weight of a word is the *inverse document frequency*, which describes how uncommon the word is: $idf(w_j) = log(n / n_j)$ with n_j the number of documents that contain word w_j.

Classification models are built using a training set of labeled documents, meaning we know the value of the target variable being predicted/estimated. The resultant classification model, or classifier, maps any document to one of the predefined classes. Specifically, the classifier maps the document to a score representing the likelihood of belonging to the class; this score is compared to a threshold for classification. Based on an independent test set, the performance of the model can be assessed by comparing the true labels with the predicted labels.[7]

Global Explanations

The most common approach to understanding a predictive model is to examine the coefficients of a linear model. Unfortunately such an approach is impracticable for a model with 10^4 to 10^6 variables. For such applications, the most common approach for a linear model is to list the variables (words in our case) with the highest weights. To understand more complex models such as neural networks (Bishop 1996) and nonlinear support-vector machines (SVMs) (Vapnik 1995), the principal approach is rule extraction: rules or trees are extracted that mimic the black box as closely as possible (Craven and Shavlik 1997; Martens et al. 2007). The motivation for using rule extraction is to combine the desirable predictive behavior of nonlinear techniques with the comprehensibility of decision trees and rules. Previous benchmarking studies have revealed that when it comes to predictive accuracy, nonlinear methods often outperform traditional statistical methods such as multiple regression, logistic regression, naive Bayesian, and linear discriminant analysis (see Baesens et al. 2003; Lessmann et al. 2008).

These rule extraction approaches are not suitable for our present problem for several reasons. Not all classifications are explained by these rule extractions. For some instances that seem to be explained by the rules, more refined (and therefore more accurate) explanations exist. In addition, often one is only interested in the explanation of the classification of a single data instance—for example, because it has been brought to a manager's attention as a classification error or simply because additional information is required for this case (to address a perceived anomaly, or for other learning).

In addition, global explanations do not provide much insight for document classification anyway, because of the massive dimensionality. For a classification tree to remain readable, it cannot include thousands of variables (or nodes). Similarly, listing all of these thousands of words with their corresponding weights for a linear model will not provide much insight into individual decisions. Considering our running example of web page classification for safe advertising, what we want to know is, *why did the model classify this particular web page as containing objectionable content?*

Instance-Level Explanations

Over the past few years, explanation methods have been introduced that explain the predictions for individual instances (Baehrens et al. 2010; Robnik-Sikonja and Kononenko 2008; Štrumbelj and Kononenko 2010; Štrumbelj et al. 2009). Generally, these methods provide an explanation as a vector with a real-valued score for each of the variables, indicating the extent to which it contributes to the classification. This makes sense for many classification problems, which have relatively few variables (e.g. , the median number of variables for the popular UCI benchmark datasets is 18.5; Hettich and Bay 1996). However, due to the high dimensionality of the data, this sort of explanation is not ideal for document classi-

[7]Note that latent semantic analysis (LSA) (Deerwester et al. 1990) is sometimes used for indexing and information retrieval (e.g., Sidorova et al. 2008). Its clustering over the identified concepts can provide improved understanding, but is different from making or explaining prediction models based on labeled data.

fication—possibly not useful at all. Considering our safe-advertising problem, an explanation for a web page's classification as a vector with thousands of non-zero values generally will not be comprehensible. The words with the highest contribution scores will have the biggest impact on the classification; however, we still will not know which (combination of) words actually led to any given classification.

Aside from the unsuitable format of these previous explanations, previous instance-based explanation approaches are unable to handle high-dimensional data computationally. The sample-based approximation method of Štrumbelj and Kononenko (2010) is reported to be able to handle up to about 200 variables, albeit requiring hours of computation time. Štrumbelj and Kononenko acknowledge that for such data sets other approaches should be introduced:

> Arguably, providing a comprehensible explanation involving a hundred or more features is a problem in its own right and even inherently transparend [sic] models become less comprehensible with such a large number of features (p. 13).

Because of this inability to deal with the high dimensionality of document data, these methods are not applicable for explaining documents' classifications.

In focusing on document classification, we take advantage of three main observations to define a slightly different explanation task from that addressed by prior work. This new task will address the motivating business needs and we will be able to solve it efficiently. The first observation is that, in many document classification problems, there really are two quite different explanation tasks, and we often are interested specifically in one of them: why documents were classified as a particular focal class (a "class of interest"). The other task is to explain why a document was not classified as a particular class. Considering our web page classification setting, we will focus primarily on explaining why a page has received (rightly or wrongly) a positive classification of containing objectionable content. The asymmetry is due to the negative class being a default class: if there is no evidence of the class of interest (or of any of the classes of interest), then the document is classified as the default class. In this paper we will not treat in detail the other explanation task. The question of why a particular page has *not* received a positive classification can be important as well, but reflection tells us that it is indeed a very different task. Often the answer is "the page did not exhibit any of the countless possible combinations of evidence that would have led the model to deem it objectionable." The problem here generally is: "How do I *fix* the model given that I believe it has made an error on this document?" This is a fundamentally different problem and thereby should require a very different solution—for example, an interactive solution where users tell the system why the page should be a positive, for example using dual supervision (Sindhwani and Melville 2008), or a relevance feedback/active learning system where chosen cases are labeled and then the system is retrained (Attenberg et al. 2011). These are important problems, but are beyond the scope of this paper.

The second important observation is that in contrast to the individual variables in many predictive modeling tasks, individual words can be quite comprehensible. The innate comprehensibility of the words often will immediately give deep intuitive understanding of the explanation. As we will see, when it does not it can indicate problems with the model.

The third observation is that in document classification, removing all occurrences of a word always sets the corresponding variable's value to zero. This allows us to formulate an optimization problem for which we can find solutions fast.

Explaining the Classification of Documents

The question we address is: *Why is this document classified as a non-default class?* The technique(s) we introduce will provide an explanation as a set of words present in the document such that removing these words causes a change in the class. Only when all words in the explanation are removed does the class change (the set is minimal).

To define the explanation formally (see Definition 1), we need to recall that a document $D \in \mathrm{D}$ is a bag (multiset) of words. Let W_D be the corresponding set of words. We presume that classifications are based on a classifier C_M, which is a function from documents to classes. Later, our heuristic algorithm will presume that C_M incorporates at least one scoring function f_{C_M}; classifications will be based on scores exceeding thresholds (in the binary case), or choosing the class with the highest score (in the multiclass case). The majority of classification algorithms operate in this way, including all that we discuss in this paper.

DEFINITION 1. Consider a document D consisting of m_D unique words W_D from the vocabulary of m words: $W_D = \{w_i, i = 1, 2, \ldots, m_D\}$, which is classified by classifier C_M: $\mathrm{D} \to \{1, 2, \ldots, k\}$ as class c. An *explanation for document D's classification* is a set E of words such that removing all words in E from the document leads C_M to produce a different classification. Further, an explanation E is minimal in the sense that removing any subset of E does not yield a change in class. Specifically,

E is an explanation for $C_M(D) = c$
1. $E \subseteq W_D$ (the words are in the document),
2. $C_M(D \setminus E) \neq c$ (the class changes), and
3. $\nexists E' \subset E : C_M(D \setminus E') \neq c$ (E is minimal).

$D \setminus E$ denotes the result of removing the words in E from document D.

Definition 1 is specifically tailored to document classification. It provides intuitive explanations in terms of words present in the document, and we will be able to produce such explanations even in the massively dimensional input spaces typical of document classification. Specifically, Definition 1 differs from those of prior approaches in that the explanation is a set of words rather than a vector. It also defines the size of the explanation as the cardinality of E. Our empirical analysis will reveal that explanations typically are quite small (often about a dozen words) as compared to the size of the vocabulary, and as such the technique is able to effectively transform the high-dimensional input space to a low-dimensional explanation. This is of crucial importance to satisfy a manager's or a customer's need to understand a classifier's decision, to obtain better understanding of the domain, or to improve the classifier's performance.

The desire to be model-independent is important and worth discussing further. Some firms use different model types for different document classification problems. Complicated nonlinear models are often used, such as nonlinear SVMs (Joachims 1998) or boosted trees (Schapire and Singer 2000). These models are incomprehensible globally. Explaining the individual decisions made by such models to a client, manager, or subject-matter expert is a natural application of our approach. When a *linear* model is being used, one could argue simply to list the top k words that appear in the document with the highest positive weights as an explanation for the class (assuming we are explaining class 1 versus class 0). The choice of k can be set to 10, for example. A more suitable choice for k would follow our definition and be the minimal number of top words such that removing these k words leads to a class change. This is exactly what our approach would provide with a linear model. Finally, although they are often cited as producing comprehensible models, classification trees for document classification do not provide the sort of explanations we need (as in Definition 1): they do not explain what words actually are responsible for the classification. All words from the root to the specific leaf for this document may be important for the classification, but some of these words are likely not present in the document (the path branched on the absence of the word) and we do not know which (minimal) set of words actually is responsible for the given classification. Appendix C discusses relations to inverse classification and to K-nearest neighbor approaches in more detail.

Finding Document Classification Explanations

Now we can present the problem more precisely from an optimization perspective. Unlike the settings in prior work, here we are looking for the shortest paths in the space defined by word *presence*, based on the effect on the surface defined by the document classification model, which is in a space defined by more sophisticated word-based features (e.g., frequency or tfidf, as described above). Given a document vocabulary with m words, consider a *mask vector* to be a binary vector of length m, with each element of the vector corresponding to one word in the vocabulary. An explanation E can be represented by a mask vector μ_E with $\mu_E(i) = 1 \Leftrightarrow w_i \in E$ (otherwise, $\mu_E(i) = 0$). Recall that the size of the explanation is the cardinality of E, which becomes the L1-norm of μ_E. Then $D \setminus E$ is the Hadamard product of the feature vector of document D (which may comprise frequencies or tfidf values) with the one's complement of μ_E.

Thus, finding a minimal explanation corresponds to finding a mask vector μ_E such that $C_M(D \setminus E) \neq C_M(D)$, but if any bit of μ_E is set to zero forming E', then $C_M(D \setminus E') = C_M(D)$.

Objectives and Performance Metrics

Although Definition 1 is quite concise, the objectives for an algorithm searching for such explanations can vary greatly. A user may want to (1) find at least one minimum-sized explanation: an explanation such that no other explanation of smaller size exists; (2) find all minimal explanations; (3) find all explanations of size smaller than a given k; (4) find l explanations, as quickly as possible ($l = 1$ may be a common objective); (5) find as many explanations as possible within a fixed time period. Combinations of such objectives may also be of interest. To allow the evaluation of different explanation procedures for these objectives, we define a set of performance metrics:

Search effectiveness:
1. PE: Percentage of test instances explained

Explanation complexity:

2. AWS: Average number of words in the smallest explanation

Problem complexity:

3. ANS: Average number of smallest explanations produced
4. ANT: Average number of total explanations produced

Computational complexity:

5. ADF: Average time to find the first explanation
6. ADA: Average time to find all explanations

These performance metrics describe the behavior of a document explanation algorithm.[8] In a separate analysis, one can also employ a domain expert to verify the explanations. An interesting question that is beyond the scope of this paper is: If the explanations are counterintuitive, does that reflect on the explanation-finding method, or only on the underlying classification model that is being explained? We will show that some explanations reveal the over-fitting of the training data by the modeling procedure, which often is not revealed by traditional machine learning evaluations that examine summary statistics (error rate, area under the ROC curve, etc.).

Complete Enumeration of Explanations of Increasing Size

A straightforward approach to producing explanations is to conduct a complete search through the space of all candidate word combinations, starting with one word, and increasing the number of words until an explanation is found. The candidate word combinations are all combinations of words in the document (rather than in the vocabulary), for which a subset of the words was not already found to be an explanation. This algorithm starts by checking whether removing any one word w from the document would cause a change in the class label and, if so, producing the explaining rule "**if** word w is removed **then** the class changes." For a document with m_D words, this requires m_D evaluations of the classifier. If the class does not change based on one word only, the case of several words being removed simultaneously will be considered. The algorithm considers all word combinations of size 2, then 3, and so on. For combinations of two words, the algorithm makes $m_D \times (m_D - 1)$ evaluations, for all combination of three words $m_D \times (m_D - 1) \times (m_D - 2)$ evaluations and, generally, for a combination of k words, we need $m_D!/(m_D - k)! = O(m_D^k)$ evaluations. This complete search scales exponentially with the number of words in the document. Therefore, it is impracticable for all but the smallest documents. It could be used for naturally small documents, such as explaining the classifications of search queries, sentiment predictions for Twitter posts, or classifications based on nonstandard documents such as ad targeting classification based on collections of visited URLs. Note that if the goal of the search is to find *an* explanation, the complete search is almost certain not to exhaustively search the space. If a short explanation exists, then the complete search may be quite fast for such short documents. However, as the search will be impracticable for most document settings, including the domains of our experiments, we will not consider complete search further.

Explaining Documents' Classifications: A Heuristic Search Approach

The heuristic search approach, described in Algorithm 1, is designed to find one or more minimal explanations in reasonable time. It is not guaranteed to find all solutions or the shortest solution. (We will see that it is optimal in an important setting.) The approach is based on two notions:

1. **Heuristic search guided by local improvement**: We assume that the underlying classification model will always be able to provide a probability estimate or score[9] in addition to a categorical class assignment. We will denote this score function for classifier C_M by $f_{C_M}(\cdot)$. The algorithm starts by listing all potential explanations of one word, and calculating the class and score change for each. The algorithm proceeds as a straightforward heuristic best-first search. Specifically, at each step in the search, given the current set of word combinations denoting partial explanations, the algorithm next will expand the partial explanation for which the output score changes the most in the direction of class change. Expanding the partial explanation entails creating a set of new, candidate explanations, comprising all combinations with one additional word from the document (that is not yet included in the partial explanation).

[8] Note that explanation accuracy is not a major concern. An explanation, by definition, changes the predicted class; it is straightforward to ensure that explanations always are correct. What is important with regard to usefulness is how complex the explanations are and how long it takes for the algorithm to find them.

[9] No explicit mapping to [0, 1] is necessary; a score that ranks by likelihood of class membership is sufficient. The scores for different classes must be comparable in the multiclass case, so in practice scores often are scaled to [0,1]. For example, support-vector machines' output scores are often scaled to (0,1) by passing them through a simple logistic regression (Platt 1999).

Algorithm 1: SEDC: Search for Explanations for Document Classification (via Best-First Search with Pruning)

Inputs:
$W_D = \{w_i, i = 1, 2, \ldots, m_D\}$ % Document D to classify, with m_D words
C_M: D → $\{1, 2, \ldots, k\}$ % Trained classifier C_M with scoring function f_{C_M}
max iteration = 30 % Maximum number of iterations

Output:
Explanatory list of rules, R

$c = C_M(D)$ % The class predicted by the trained classifier
$p = f_{C_M}(D)$ % Corresponding probability or score
$R = \{\ \}$ % The explanatory list that is gradually constructed
combinations_to_expand_on = { }
P_combinations_to_expand_on = { }
for $i = 1 \rightarrow m_D$ **do**
    $c_{new} = C_M(D \setminus w_i)$ % The class predicted by the trained classifier if word w_i did not appear in the document
    $p_{new} = f_{C_M}(D \setminus w_i)$ % The probability or score predicted by the trained classifier if word w_i did not appear in the document
    if $c_{new} \neq c$ **then**
        $R = R \cup$ "*if* word w_i is removed *then* class changes"
    else
        combinations_to_expand_on = *combinations_to_expand_on* ∪ w_i
        P_combinations_to_expand_on = *P_combinations_to_expand_on* ∪ p_{new}
    end if
end for
for *iteration* = 1 → *max_ iteration* **do**
    combo = word combination in *combinations_to_expand_on* for which (*p* − *P_combinations_to_expand_on*) is maximal % The best first
    combo_set = create all expansions of *combo* with one word
    *combo_set*2 = remove combinations containing already found explanations of *R* from *combo_set* % The pruning step
    for all combos C_O in *combo_set*2 **do**
    $c_{new} = C_M(D \setminus C_O)$ % The class predicted by the trained classifier if the words in C_O did not appear in the document
    $p_{new} = f_{C_M}(D \setminus C_O)$ % The probability or score predicted by the trained classifier if the words in C_O did not appear in the document
    if $c_{new} \neq c$ **then**
            $R = R \cup$ "*if* words *Co* are removed *then* class changes"
        else
            combinations_to_expand_on = *combinations_to_expand_on* ∪ C_O
            P_ combinations_to_expand_on = *P_ combinations_to_expand_on* ∪ p_{new}
        end if
    end for
end for

2. **Search-space pruning**: For each explanation with *l* words that is found, we do not need to check combinations of size *l* + 1 with these same words, hence we can prune these branches of the search tree. For example, if the words *hate* and *furious* provide an explanation, we are not interested in explanations of three words that include these two words, such as *hate*, *furious*, and *never*. This search problem generally (including the complete

search solution) is an instance of unordered-set search. Unordered-set search is described in detail by Webb (1995) (and references therein), including optimizations that speed up the search substantially, while still allowing various guarantees, including this sort of search-space pruning. The pruning is somewhat different from the search-space pruning in similar set-enumeration algorithms, such as the Apriori association rule mining algorithm (Agrawal and Srikant 1994), in that it is based on set subsumption rather than coverage statistics.

For the case of a linear classifier with a binary feature representation, we might explain the classification by looking at the words with the highest weights that appear in the document. However, we would still want to know which words exactly are responsible for the classification. SEDC produces minimum-size explanations for linear models, which we discuss further next. Assuming again a class 1 versus class 0 prediction for document *i*, SEDC ranks all words appearing in the document according to the product $\beta_j x_{ij}$, where β_j is the linear model coefficient. The explanation with the top-ranked words is an explanation of smallest size.

LEMMA 1. *For document representations based on linear binary-classification model* $f_{C_M}(D) = \beta_0 + \sum \beta_j x_{ij}$ *with binary (presence/absence) features, the smallest explanation found by SEDC will be a minimum-size explanation. Specifically, for* E_1, E_2 *explanations, if* E_1 *is the smallest explanation found by SEDC,* $|E_1| = k \Rightarrow \nexists E_2$: $|E_2| < k$. *Furthermore, the first explanation found by SEDC will be of size k.*

Proof (by contradiction): If no explanation exists, then the theorem holds vacuously. Assume there exists at least one explanation. In the linear model, let the (additive) contribution w_{ij} to the output score for word j of document i be the linear model weight β_j corresponding to binary word-presence feature x^b_{ij} for those words that are present in document *i* (and zero otherwise).

Assume w.l.o.g. that the classification threshold is placed at $f_{C_M}(D) = 0$. SEDC will compose the first candidate explanation E^* by first selecting the largest w_{ij} such that the word is present in the document, $x^b_{ij} = 1$, and adding word *j* to the explanation. SEDC will then add to E^* the word with the next-largest such w_{ij}, and so on until $f_{C_M}(E^*) \le 0$. Thus, the first explanation E_1 by construction will consist of the *k* highest-weight words that are present in the document.

Now assume that there exists another explanation E_2 such that $|E_2| < k$; being an explanation, $f_{C_M}(E_2) \le 0$. Recall that explanations are minimal, so $\nexists S \subsetneq E_1$: $f_{C_M}(S) \le 0$. Thus E_2 must have at least one element $e \notin E_1$. Let Σ_E denote the sum of the weights corresponding to the words in an explanation E. For a linear model based on the (binary) presence/absence of words, $f_{C_M}(X \setminus Y) = f_{C_M}(X) - \Sigma_Y$. As noted above, E_1 comprises by construction the *k* words with the largest w_{ij}, so $\forall w_{ij} \in E_1, \forall w_e \notin E_1$: $w_{ij} \ge w_e$. Therefore, $\exists S \subsetneq E_1 : \Sigma_S > \Sigma_{E_2}$, which means that $\exists S \subsetneq E_1$: $f_{C_M}(D \setminus S) \le f_{C_M}(D \setminus E_2)$. But $\forall S \subsetneq E_1$: $f_{C_M}(D \setminus S) > 0$ and thus $f_{C_M}(D \setminus E_2) > 0$. Therefore, E_2 is not an explanation, a contradiction. ■

This optimality applies as well to monotonic transformations over the output of the linear model, as with the common logistic transform used to turn linear output scores into probability estimates. The optimality also applies generally for linear models based on numeric word-based features, such as frequencies, tfidf scores, etc., as detailed in the following theorem.

THEOREM 1. *For document representations based on linear models* $f_{C_M}(D) = \beta_0 + \sum \beta_j x_{ij}$ *with numeric word-based features, such as frequencies or tfidf scores, that take on positive values when the word is present and zero when the word is absent, the smallest explanation found by SEDC will be a minimum-size explanation. Specifically, for* E_1, E_2 *explanations, if* E_1 *is the smallest explanation found by SEDC,* $|E_1| = k \Rightarrow \nexists E_2$: $|E_2| < k$. *Furthermore, the first explanation found by SEDC will be of size k.*

Proof: Decompose each nonnegative word feature x_{ij} into the product $x^b_{ij} d_{ij}$ of a binary word presence/absence feature x^b_{ij} and a document-specific non-negative weight d_{ij}. The corresponding term in the linear model $\beta_j x_{ij}$ then becomes $\beta_j d_{ij} x^b_{ij}$. The proof then follows the previous proof directly, except with the additive contribution of each word being $w_{ij} = \beta_j d_{ij}$. ■

For nonlinear models, no such optimal solutions are guaranteed, in the sense that smaller explanations could exist. For multiclass classification problems, optimal solutions also are not guaranteed if one decomposes the problem into several binary classification problems (as in a one-versus-rest or one-versus-one approach), since the final classification of data instances now depends on several models with their own weights. This motivates our next optimization: applying local search on the obtained explanations.

SEDC Augmented with Local Search

The SEDC algorithm has two potential issues when applied to nonlinear models, addressed by two optimizations. First (and most importantly), seeing that the prediction space is nonlinear in the words, the obtained explanations might not contain a minimal subset of words, required by Definition 1 (requirement 3; E is minimal). It could be that removing a word from the explanation E still provides an explanation E'; hence, there exists an explanation $E' \subset E$: $C_M(D \setminus E') \neq c$. To address this concern, we extend the previously defined heuristic search procedure with a limited local search post-processing phase applied to the obtained explanations. This method will prune the explanation, if necessary, by verifying whether removing a word (or word combination) from an obtained explanation, E also provides an explanation E'. If that is the case, E is replaced by the smaller explanation E' containing a subset of the words of E. This guarantees minimality of the explanations (although in the empirical studies we never observed the need for such pruning).

The second potential problem with SEDC for nonlinear models is that potentially smaller explanations exist (with different words, making it different from the above optimization) than those obtained. Formally, there might exist an explanation E', where $E' \setminus E \neq \emptyset$ (E' has some word(s) that E does not), $|E'| < |E|$ (explanation E' is smaller than E), $C_M(D \setminus E') \neq c$ (E' also defines an explanation).

To investigate the extent of this potential problem, we define a second local search approach that is applied to the explanations found by the heuristic search method with the previously described optimizations. For each explanation, we replace two words by another word of the document, not yet in the explanation. Next, we attempt replacing three words of the explanation by two words of the document, not yet in the explanation, and so on. This yields a very large number of potential combinations to check: replacing a set of k words of an explanation for a document with m_D words yields $\binom{m_D - k}{k}$ combinations.[10] To deal with this huge number of new word combinations to check, we limit ourselves in our experiments up to $k = 5$ words, and a maximum of 5,000 combinations. If more exist, no attempt to optimize is undertaken. Within our empirical results, this local search addition provided an improvement of one word for only very few explanations (less than 1%), while requiring much more time (up to two hours per explanation, even with the limitation on the number of combinations). Seeing that the additional local search is so computationally expensive compared to the heuristic search procedure, with negligible improvements in explanation size, the results in the next section are provided without the local search.

SEDC with Branch-and-Bound

As described earlier ("Objectives and Performance Metrics"), there are various objectives one might have when finding explanations for document classifications. In the important case where one wants the shortest explanation, or the set of shortest explanations, the SEDC search can be improved by keeping track of the current shortest explanation found, and pruning from the search space all longer explanations (a simple branch-and-bound search), which can result in massive portions of the search space being discarded en masse once a first explanation has been found.[11]

Empirical Analysis

We now present an empirical case study of classifying web pages as containing adult content. A follow-up analysis is presented in Appendix A based on a suite of text classification problems (the 20 newsgroups) widely used in the research literature.

Explaining Web Pages' Classifications for Safe Advertising

The case study is based on data obtained from a firm that focuses on helping advertisers to avoid inappropriate adjacencies between online advertisements and web content, similar to our motivating example above. Specifically, the analysis is based on a data set of 25,706 web pages, labeled as either having adult content or not. The web pages are described by tfidf scores over a vocabulary chosen by the firm, including a total of 73,730 unique words. No stemming was conducted. The data set is balanced by class, with half of the pages containing adult content and half non-adult content. For this data set, the class labels were obtained from a variety of sources used in practice, including Amazon's Mechanical Turk. Given the variety of labeling sources, the quality of the labeling might be questioned (Sheng et al. 2008). Interestingly, the explanations indeed reveal that certain web pages are wrongly classified. No meta-data, links, or information on images is being used for this study; the inclusion of such data could improve the model further, but the focus of this paper is on textual document classification.

[10] To indicate how large these values can be, for $k = 3$ and $m_D = 100$ we have 147,440 combinations; for $k = 5$ and $m_D = 500$ we have 255,244,687,600 combinations.

[11] Unfortunately, for the general problem one cannot give nontrivial upper and lower bounds on explanation size given a partial explanation. For particular types of models, this may be possible, yielding more sophisticated branch-and-bound searches.

For this analysis, we built SVM document classification models with linear and RBF kernel functions.[12] The linear model is correct on 96.2 percent of the test instances, with a sensitivity (percentage of non-adult web pages correctly classified) of 97.0 percent, and a specificity (percentage of adult web pages correctly classified) of 95.6 percent. The nonlinear RBF kernel model has an accuracy of 93.3 percent, with a sensitivity of 89.0 percent and a specificity of 96.5 percent.

Global Explanations Are Not Satisfactory

As discussed above, rule extraction is the most researched and applied model explanation methodology. Trying to comprehend the SVM model, a tree can be extracted by applying the C4.5 tree induction technique (Quinlan 1993) on the aforementioned safe advertising data set with class labels changed to SVM predicted labels. Unfortunately, we could not get C4.5 to generate a small tree that models either SVM model (with linear or RBF kernel) with high-fidelity. A tree with 327 nodes models the classifier with a fidelity of only 87 percent. Pruning the tree further reduces the size, but further decreases fidelity.

As discussed above, an alternative method for comprehending the function of a linear document classifier is to examine the weights on the word features, as these indicate the effect that each word has on the final output score. As with the distinction between Lemma 1 and Theorem 1, we need to keep in mind that in a preprocessing step the data set is encoded in tfidf format. Hence for actual document explanations, the frequency is vital.[13] Figure 3 shows the weight sizes of all the words in the vocabulary; the weights are ranked smallest to largest, left to right. Many words show a high indication of adult content, while many others show a clear counter-indication of adult content. Looking deeper, Table 1 shows the highest (positive) weight words, as well as the words that give the highest mutual information (with the positive class) and information gain. We additionally list the top words when taking into account the idf weights, *viz*, based on the weights of the words multiplied with the corresponding idf values. The final column shows the words most frequently occurring in the explanations, which will be elaborated on below.

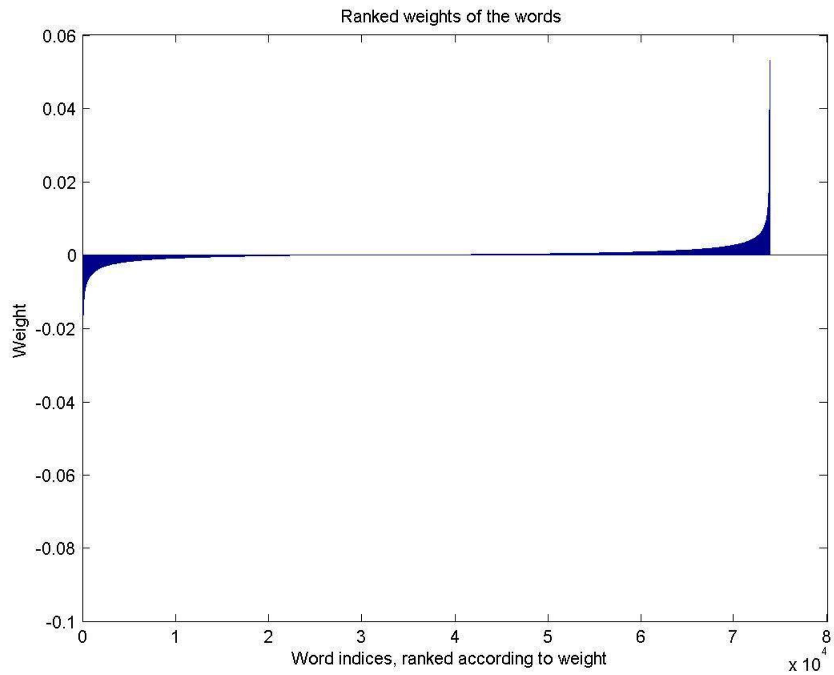
From Table 1 we see that most of the indicative words for adult content that are ranked highly using the mutual information criterion are very rare, unintuitive words. It may be possible to engineer a better information-based criterion, for example countering this over-fitting behavior by requiring a minimal frequency of the top-ranked words, but later results will show why such efforts ultimately are destined to fail to provide a comprehensive explanation. The top words provided by the other rankings on the other hand are quite intuitive. As stated before, even initially not-so-obvious words as *welcome*, *enter*, or *age* make sense once we realize that many positive examples are entrance pages of adult sites, which inform a visitor about the content of the website and require verification of age. Nevertheless, as we will see next, explanation of individual decisions simply requires too many individual words. Consider that we would have to produce a list of over 700 of the highest-weight words just to include *porn* and over 10,000 to include *xxx*.

Given the intuitiveness of the top-weighted words, we should consider how well a short list of such words explains the behavior of the model. Does the explanation of a web page typically consist of (some of) the top-100 or so words? It turns out that the content of web pages varies tremendously, even within individual categories. For "adult content," even though some strongly discriminative words exist, the model classifies most web pages as being adult content for other reasons. This is demonstrated by Figure 4, which plots the percentage of the classifications of the test instances that would be explained by considering the top-k words (horizontal axis) by weight (with and without idf correction), mutual information, and information gain. Specifically, if an explanation can be formed by any subset of the set of top-k words, then the document is deemed explained. So for example, if an explanation would be "if words (welcome enter) are removed then class changes," that explanation would be counted when $k \geq 2$.

We see from Figure 4 that we would need thousands of these top words before being able to explain a large percentage of the individual documents, as shown by the line with words ranked on the weight. More precisely, more than 2,000 top-weight words (3% of the vocabulary) are needed before even half of the documents are explained. Using the ranking based on mutual information requires even more words. This suggests either (1) that many, many words are necessary for individual explanations, or (2) the words in the individual explanations vary tremendously. The latter conclusion is also supported by the fact that the document-term matrix is very sparse even when the documents belong to the same topic. This motivates the use of an instance-level explanation algorithm not only for obtaining understanding of the individual decisions, but also for understanding the model overall.

[12]Using the LIBLINEAR (Fan et al. 2008) and LIBSVM (Chang and Lin 2001) packages, with 90% of the data used as training data, the remaining 10% as test data. SEDC was coded in Matlab and is available upon request from the first author. Experiments were run on an Intel Core 2 Quad (3 GHz) PC with 8GB RAM.

[13]The inverse document frequency is constant across documents, and could be incorporated in the model weights to facilitate global explanation.



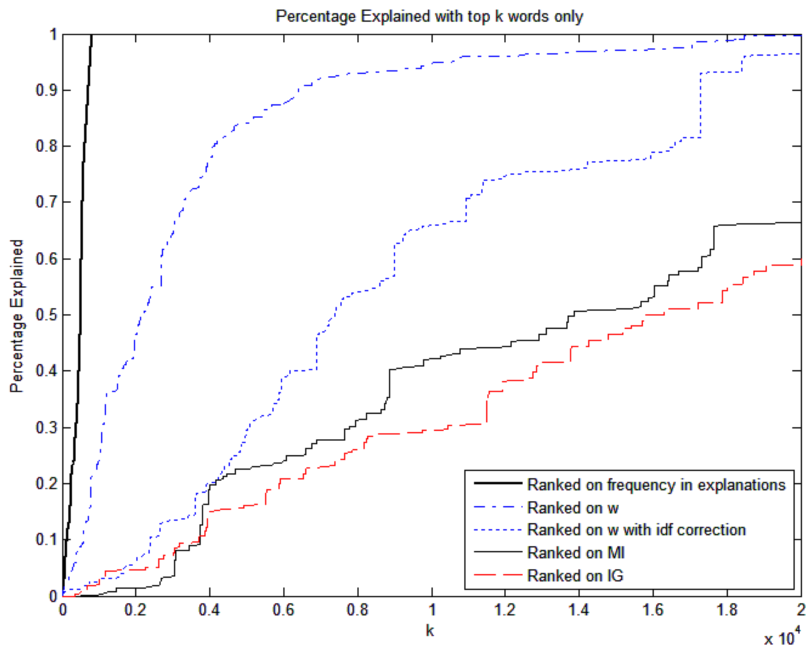
The size of the weights for all 73,730 words, ranked left to right according to increasing weight.

Figure 3. Size of Weight

Table 1. Global Explanations of the Model Produced by Listing the Top Words Providing Evidence for the Adult Class†

Ranking based on				
Mutual Information	**Information Gain**	**Size of weight**	**Size of weight with idf correction**	**Frequency of word occurring in our explanations**
primarykey	privacy	welcome	permanently	adult
sessionid	policy	enter	fw	age
youtubeid	home	adult	welcome	enter
webplayerrequiredgeos	us	permanently	compuserve	site
vnesfrsgphplitgrmxnlkrause	advertise	site	copyrightc	sex
videocategoryids	about	age	prostitution	years
usergeo	adult	usc	acronym	material
latestwebplayerversion	search	searches	tribenet	are
isyoutubepermalink	comments	over	amateurbasecom	sites
isyoutube	contact	erotic	gorean	hardcore

†Five rankings are considered, based on mutual information, information gain, the weights of the words in the model, the weights with idf correction (weight multiplied with word idf), and the frequencies of the words occurring in our explanations.



Percentage of 100 adult-classified test instances explained when considering only the top *k* words, ranked according to the frequency of occurrence in the explanations, the weights (w), the weights with idf correction, mutual information (MI), and information gain (IG).

Figure 4. Adult-Classified Test Instances

When we rank the words according to how often they occur in explanations, we obtain the line with the maximal area underneath. For the 100 classified instances, a total of 810 unique words are used in all the explanations (where we consider a maximum of 10 minimal explanations for a single data instance). This already suggests a wide variety of words are present in the explanations. The instance-based explanations can be aggregated to form a global explanation by listing the words that occur most frequently in the explanations, as shown in the final column of Table 1, which provides yet another benefit of the instance-level explanations.

Instance-Level Explanations Are Effective

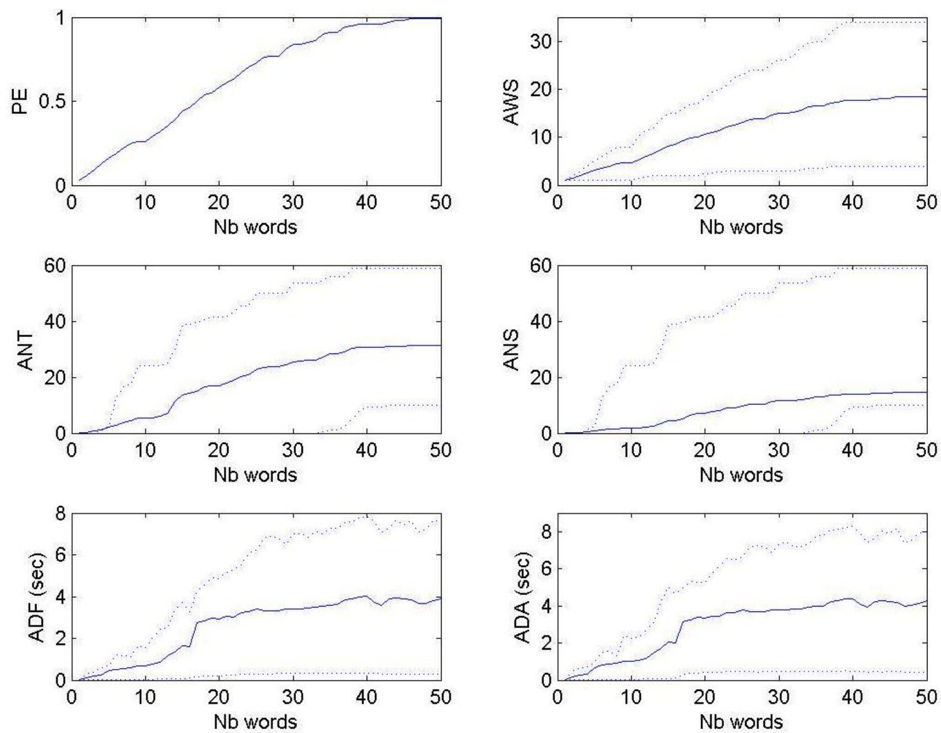
We will show now that SEDC is effective, and fast as well. We initially focus on the linear classification model. Explanation 2 shows typical explanations for classifications of several documents. We show the first three explanations of instances with explanations that are appropriate for publication. These explanations demonstrate several things. First, they directly address suggestion (1) just above: in fact, documents generally do not need many words to be explained. They also provide evidence supporting suggestion (2): the words in the explanations are quite different, including explanations in different languages.

We can examine the size of explanations more systematically by referring to the explanation performance metrics introduced earlier in "Objectives and Performance Metrics." The top-left plot in Figure 5 shows the percentage of the test cases explained (PE) when an explanation is limited to a maximum number of words (on the horizontal axis). We see that almost all of the documents have an explanation comprising fewer than three dozen words, and more than half have an explanation with fewer than two dozen words. In other words, each explanation is very concise, as it uses only about 0.01 percent of the words in the vocabulary. Note that even explanations containing dozens of words can easily give an understanding of why the classifier classified the document as the class of interest, as is discussed and shown later in the section "Hyper-Explanations Are Necessary." Figure 5 also shows that, not surprisingly, the number of words in the smallest explanation (AWS plot) and the (smallest and total) number of explanations (ANS, ANT plots) both grow as we allow larger and larger explanations.[14]

[14]In the experiments, we limit ourselves to searching for 10 explanations, if 10 or more explanations have been found, no further word expansions/iterations are attempted.

Table 2. Explanation Performance Metrics (for the false positives (FP) versus true positives (TP) of the linear model, allowing up to 30 words in an explanation)

	PE	AWS	ANS	ANT	ADF	ADA
FP	90.3%	9.2	12.0	35.2	2.3	3.1
TP	76.0%	15.3	13.4	25.5	2.9	3.3



Explanation performance metrics as a function of the maximal number of words allowed in an explanation. Both the performance and the complexity increase with the number of words. In addition to the averages, the 10th and 90th percentiles are also shown (dotted lines).

Figure 5. Explanation Performance Metrics

Table 2 presents the differences between the false and true positives (for the default threshold of zero). Interestingly, we find higher coverage, as well as more and smaller explanations for the web pages wrongly classified as adult (false positives, FP) versus those correctly classified as adult (true positives, TP). Seeing that FPs are classifications we are particularly interested in explaining (the perceived anomalies, as described by Gregor and Benbasat 1999), this suggests that the overall explanation metrics yield conservative estimates of practical performance for this case study.

More interesting, examining these performance metrics gives a view into how the classification model is functioning in this application domain. Specifically, the plots show that document explanation sizes vary quite smoothly and that there seem to be many different explanations for documents. The former observation suggests that the strength of the individual evidence varies widely: some cases are classified by aggregating many weak pieces of evidence, others by a few strong pieces of evidence (and some, presumably, by a combination of strong and weak). The latter observation suggests substantial redundancy in the evidence available for classification.

Figure 5 also shows that for this particular problem, explanations can be produced fairly quickly using SEDC. This problem is of moderate size; real-world document classification problems can be much larger, in terms of documents for training, documents to be classified, and the vocabulary. A brief word about scaling up can be found in Appendix B.

Explanation 2: Some explanations why a web page is classified as having adult content for web pages in the test set.

Explaining document 13 (class 1) with 61 features and class 1 ...

- Iteration 7 (from score 0.228905 to -0.00155753): If words (submissive pass hardcore check bondage adult ac) are removed then class changes from 1 to -1 (1 sec)
- Iteration 7 (from score 0.228905 to -0.00329069): If words (submissive pass hardcore check bondage adult access) are removed then class changes from 1 to -1 (1 sec)
- Iteration 7 (from score 0.228905 to -0.00182021): If words (submissive pass hardcore check bondage all adult) are removed then class changes from 1 to -1 (1 sec)

Explaining document 30 (class 1) with 89 features and class 1 ...

- Iteration 4 (from score 0.894514 to -0.0108126): If words (searches nude domain adult) are removed then class changes from 1 to -1 (1 sec)
- Iteration 6 (from score 0.894514 to -0.000234276): If words (searches men lesbian domain and adult) are removed then class changes from 1 to -1 (1 sec)
- Iteration 6 (from score 0.894514 to -0.00225592): If words (searches men lesbian domain appraisal adult) are removed then class changes from 1 to -1 (1 sec)

Explaining document 32 (class 1) with 51 features and class 1 ...

- Iteration 8 (from score 0.803053 to -0.0153803): If words (viejas sitios sexo mujeres maduras gratis desnudas de) are removed then class changes from 1 to -1 (1 sec)
  Translation: old mature women sex sites free naked of
- Iteration 9 (from score 0.803053 to -7.04005e-005): If words (viejas sitios mujeres maduras gratis desnudas de contiene abuelas) are removed then class changes from 1 to -1 (1 sec)
  Translation: old mature women free sites containing nude grandmothers
- Iteration 9 (from score 0.803053 to -0.00304367): If words (viejas sitios mujeres maduras gratis desnudas de contiene adicto) are removed then class changes from 1 to -1 (1 sec)
  Translation: old sites free naked mature women contains addict

Explaining document 35 (class 1) with 36 features and class 1 ...

- Iteration 6 (from score 1.04836 to -0.00848977): If words (welcome fiction erotic enter bdsm adult) are removed then class changes from 1 to -1 (0 sec)
- Iteration 6 (from score 1.04836 to -0.10084): If words (welcome fiction erotica erotic bdsm adult) are removed then class changes from 1 to -1 (1 sec)
- Iteration 6 (from score 1.04836 to -0.0649064): If words (welcome kinky fiction erotic bdsm adult) are removed then class changes from 1 to -1 (1 sec)

Table 3. Explanation Performance of SEDC (with and without branch-and-bound (B&B), explaining classifications of models from SVMs with a linear kernel and a radial basis function (RBF) kernel, allowing up to 30 words in an explanation)

kernel	PE	AWS	ANS	ANT	ADF	ADA
SEDC Linear SVM	84%	15.1	12	25	3	3
SEDC B&B Linear SVM	84%	15.1	12	12	3	3
SEDC Nonlinear RBF SVM	82%	11.1	18	28	169	187
SEDC B&B Nonlinear RBF SVM	82%	11.1	19	19	183	200

Table 3 shows SEDC's performance on both a linear SVM model and a nonlinear radial-basis function (RBF) SVM model, in each case allowing up to 30 words in an explanation. The percentage of instances explained is about the same for both the linear and nonlinear models, with the nonlinear model requiring slightly fewer words per explanation (AWS). A large difference is observed in the time needed to obtain an explanation: whereas for the linear model it takes on average four seconds to find an explanation, for the RBF model it takes almost three minutes. A deeper investigation into the reasons for the speed differences shows that processing the nonlinear models takes longer not because of the backtracking in the search. Rather, the nonlinear models simply run much slower, which has a crucial effect due to the repeated applications of the scoring function. Therefore, faster implementations of the nonlinear models could produce faster explanation performance. Please note that explanation times on the orders of minutes are not necessarily a cause for concern, depending on the context of application. In many of the application scenarios discussed above, explanation methods would be reserved for periodic development use or for tactical use when a concern arises over a particular case.[15]

Hyper-Explanations Are Necessary

Conducting the case studies brought to the fore some additional issues regarding explaining documents classifications. Specifically, a procedure for producing explanations of document classifications may provide no explanation at all. A document's explanation may be nonintuitive. There are several classes of reasons for these behaviors, which we group into *hyper-explanations*. Many of these are specifically helpful for improving the decision system's model (see the earlier section, "Explanations and Statistical Classification Models"), and for suggesting how to proceed (e.g., in light of a nonintuitive explanation).

Hyper-Explanations for the Lack of an Explanation

Let us distinguish between cases where the predicted class is the default class (hyper-explanation 1), and those where the predicted class is not the default class (hyper-explanation 2).

Hyper-Explanation 1a: No Evidence Present. The default class is predicted and no evidence in support of either class is present. For example, this would be the case when all words in the document have zero weights in the model or no words present are actually used in the model.

Technically, this case falls outside the scope of this paper's development, since we are specifically considering explaining why a document is classified as a non-default class. Nevertheless, this may be a practically important situation that cannot simply be ignored. For example, this case may have been brought to a manager's or developer's attention as a "false negative error"; that is, it should have been classified as a positive example. In this case the hyper-explanation explains exactly why the case was classified as being negative (there was no model-relevant evidence) and can be a solid starting point for a management/technical discussion about how to deal with it; for example, the model's vocabulary needs to be extended.

Hyper-Explanation 1b: No Evidence of Non-Default Class Present. The default class is predicted and only evidence in support of the default class is present. This is a minor variation to hyper-explanation 1a, and the discussion above applies regarding explaining false negatives and providing a starting point for discussions of corrective actions.

Hyper-Explanation 1c: Evidence for Default Class Outweighs Evidence for the Non-Default Class. A more interesting and complex situation is when, in weighing evidence, the model's decision simply comes out on the side of the default class. In this case, an immediate reaction may be to apply the explanation procedure to generate explanations of why the case was classified as being the default (i.e., if these words were removed, the class would change to positive). However, when the case truly is of the "uninteresting" class, the explanations returned would likely be fairly meaningless, for example, "if you remove all of the content words on the page except the 'offending words' (e.g., the words with positive weights), the classifier would classify the page as an offensive page." However, applying the procedure may be very helpful for explaining false negatives, because it would show the words that the model feels trump the positive-class-indicative words on the page (e.g., if you remove the *medical* terminology on the page, the classifier would *then* rate the page as being adult). This again could provide a solid foundation for the process of improving the classifiers.

Within our safe advertising application, an explanation for all 46 false negatives is found, indicating that indeed adult words are present, but that these are outweighed by the non-adult, negative words. Example explanations of such false negatives are given in Explanation 3. For some words like *blog* it

[15] Also, recall that these experiments were conducted mainly in Matlab on a desktop PC. Further speed improvements could easily be obtained with faster software implementations or with the high-performance computing systems typically used by organizations that build text classifiers from massive data. Importantly, once again, the complexity is independent of the size of the vocabulary. Furthermore, unordered-set search is highly parallelizable.

Explanation 3: Explanations of web pages misclassified as non-adult (false negatives), which indicate which words the model feels trump the positive-class-indicative words.

Explaining document 10 (class 1) with 31 features and class -1 (score -0.126867)...

- Iteration 4 (from score -0.126867 to 0.00460739): If words (policy gear found blog) are removed then class changes from -1 to 1 (0 sec)

Explaining document 13 (class 1) with 50 features and class -1 (score -0.123585)...

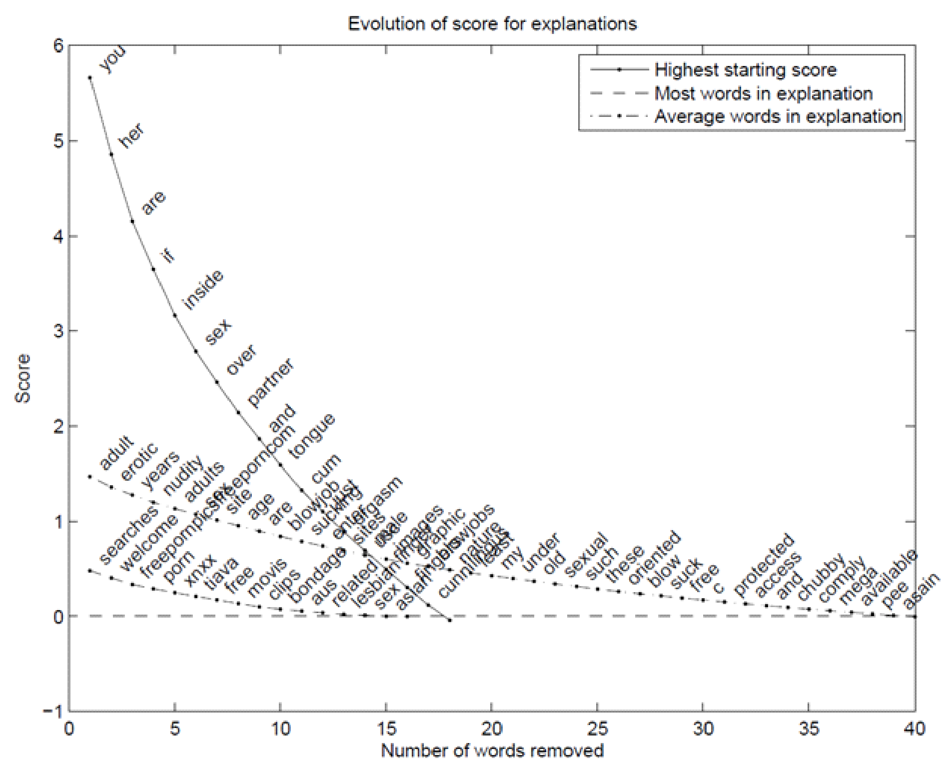
- Iteration 4 (from score -0.123585 to 0.000689515): If words (sorry miscellaneous found about) are removed then class changes from -1 to 1 (0 sec)

Explaining document 11 (class 1) with 198 features and class -1 (score -0.142504)...

- Iteration 2 (from score -0.142504 to 0.00313354): If words (watch bikini) are removed then class changes from -1 to 1 (1 sec)

Explaining document 31 (class 1) with 22 features and class -1 (score -0.0507037)...

- Iteration 4 (from score -0.0507037 to 0.00396628): If words (search ***[16] bonus big) are removed then class changes from -1 to 1 (0 sec)



Score evolution when removing words from the three selected documents: the one with the highest starting score, the one with the most words in an explanation, and a document with an average number of words in an explanation. The class changes to non-adult when the score falls below zero.

Figure 6. Score Evolution

[16]The word is removed as it is not appropriate for publication.

seems logical to have received a large non-adult/negative weight. The word *bikini* seemingly ought to receive a non-adult weight as well, as swimsuit sites are generally not considered to be adult content by raters. However, some pages mix nudes with celebrities in bikinis (for example). If not enough of these are in the training set, it potentially would cause bikini to lead to a false negative. Many other words, however, can be found in the explanations that do seem to be adult-related, and as such should receive a positive weight. All the words are great candidates for human feedback to indicate which of these words actually are adult-related and potentially to update the model's weights (a mechanism known as active feature labeling; Sindhwani and Melville 2008) or to review the labeling quality of the web pages with the word. The words occurring most in these explanations of false negatives (when considering only the first explanation) are *found*, *blog*, and *policy*. The seemingly adult-related words are not found when examining the words with most negative weights, again supporting the need to look at explanations separately, on an instance level.

Hyper-Explanation 2: Too Much Evidence of Non-Default Class Present. No explanation is provided because, although a non-default class is predicted, there are many words in support of this class and one would need to remove almost all of them before the class changes. The situations when this occurs fall along a spectrum between two fundamentally different reasons:

1. There are very many words each providing *weak* evidence in support of the class. Thus, the explanation exceeds the bound given to the algorithm, or the algorithm does not return a result in a timely fashion. Figure 6 shows the words of the explanations for three documents and how the scores change as the words are removed. The middle line, for the explanation with the most words, shows that if the number of allowed words is below 40, no explanation is found. This lack of explanation can be explained by this hyper-explanation, as too many adult-related words are present for a short explanation to be found.

2. There are very many words each providing *strong* evidence. In this case, the procedure may not be able to get the score below the threshold with a small explanation, because there is just so much evidence for the class. The full upper line with the highest starting score in Figure 6 shows such an example: when allowing fewer than 15 words in an explanation, the score remains above the threshold and no explanation can be given.

This lack of base-level explanation can be mitigated (partially) by presenting "the best" partial explanation as the search advances.

Hyper-Explanations for Nonintuitive Explanations

Explanations are always correct in the technical sense: removing the words by definition changes the class. However, it is possible that the explanation clashes with the user's intuition, creating a perceived anomaly that should be explained. Several reasons exist for this:

- **The data instance is misclassified**. Explanations of some of the web pages that are misclassified by the SVM model are listed in Explanation 4. For these pages the predicted class is adult, while the human-provided class label is non-adult (false positives). These three explanations indicate strongly that the web pages actually contain adult content and the human-provided label seems wrong. On the other hand, in other cases, explanations indicate that their web pages seem to be non-adult and hence are probably misclassified. Examples are given in Explanation 5.[17] Such explanations provide very useful support for interactive model development, as the technical/business team can fix training data or incorporate background knowledge to counter the misclassification.

- **The data instance is correctly classified, but the explanation just does not make sense to the business users/developers**. This case is particularly problematic for any automated explanation procedure, since providing explanations that "make sense" requires somehow codifying in an operationally useful way the background knowledge of the domain, as well as common sense, which is (far) beyond current capabilities and beyond the scope of this paper. Nevertheless, we still can provide a quite useful hyper-explanation in the specific and common setting where the document classification model had been built from a training set of labeled instances (as in our case study), as discussed next.

Hyper-Explanation 3: Show Similar Training Instance. For a case with a counter-intuitive explanation, we can show "similar" *training* instances with the same class. The similarity metric in principle should roughly match that used by the induction technique that produced the classifier. Such a nearest-neighbor approach can aid understanding in two ways.

(1) If the training classifications of the similar examples do make sense, then the user can understand why the focal example was classified as it was.

[17]Our models are limited by the data set obtained for the case study. By our understanding, models built for this application from orders-of-magnitude larger data sets are considerably more accurate; nonetheless, they still make both false-positive and false-negative errors, and the general principles illustrated here apply.

Explanation 4: Explanations of web pages misclassified as adult (false positives), which indicate that the model is right and the class should have been adult (class 1).

Explaining document 1 (class -1) with 180 features and class 1 (score 1.50123)...

- Iteration 35 (from score 1.50123 to -0.00308141): If words (you years web warning usc these sites site sexual sex section porn over offended nudity nude models material male links if hosting hardcore gay free explicit exit enter contains comic club are age adults adult) are removed then class changes from 1 to -1 (53 sec)

Explaining document 2 (class -1) with 106 features and class 1 (score 0.811327)...

- Iteration 24 (from score 0.811327 to -0.00127533): If words (you web warning under und these site porn over offended nude nature material links illegal if here exit enter blonde are age adults adult) are removed then class changes from 1 to -1 (15 sec)

Explaining document 3 (class -1) with 281 features and class 1 (score 0.644614)...

- Iteration 15 (from score 0.644614 to -0.00131314): If words (you sex prostitution over massage inside hundreds here girls click breasts bar) are removed then class changes from 1 to -1 (29 sec)

Explanation 5: Explanations of truly misclassified web pages (false positives).

Explaining document 8 (class -1) with 57 features and class 1 (score 0.467374)...

- Iteration 7 (from score 0.467374 to -0.0021664): If words (welcome searches jpg investments index fund domain) are removed then class changes from 1 to -1 (3 sec)

Explaining document 16 (class -1) with 101 features and class 1 (score 0.409314)...

- Iteration 8 (from score 0.409314 to -0.000867436): If words (welcome und sites searches domain de b airline) are removed then class changes from 1 to -1 (5 sec)

Explaining document 32 (class -1) with 66 features and class 1 (score 0.124456)...

- Iteration 2 (from score 0.124456 to -0.00837441): If words (searches airline) are removed then class changes from 1 to -1 (0 sec)

(2) If the training classifications do not make sense (e.g., they are wrong), then this hyper-explanation provides precise guidance to the data science team for improving the training,[18] and thereby the model.

Consider document 8. Explanation 5 suggests strongly that it contains non-adult content, even though the model classifies it as adult. The web page most similar to document 8 is also classified as adult and has 44 (out of 57) words which are the same, which are listed in Explanation 6. This is a web page with a variety of topics, and probably a listing of links to other websites, and it requires expert investigation prior to use for training (and evaluating) models for safe advertising. It could be that labelers have not properly examined the entire web site; it may be that there indeed is adult content in images that our text-based analysis does not consider; it may be that these sites simply are misclassified; or it may be that in order to classify such pages correctly, the data science team needs to construct a specifically tailored feature to deal with the ambiguity.

Discussion and Limitations

Managers and developers need to be able to interact to assess whether a classification system is behaving appropriately, and to improve it if necessary. However, recent developments in machine learning and data mining arguably have moved us further away from the needed transparency in the process of building models for business applications. There has been increasing research emphasis on techniques that produce complex models, such as boosting, nonlinear SVMs, feature

[18]Data cleaning is a very important aspect of the data mining process that has received relatively little treatment in the research literature. One of the main data cleaning activities in classifier induction is "fixing" labels on mislabeled training data.

Explanation 6: Hyper-explanation 3 showing the words of the web page most similar to document 8. This most similar web page is classified as adult, providing a hyper-explanation of why document 8 is also classified (incorrectly) as adult.

and, articles, at, buy, capital, check, china, commitment, dat, file, files, for, free, fund, funds, high, hot, in, index, instructionalwwwehowcom, international, internet, investing, investment, investments, jpg, listings, mutual, out, performance, project, related, results, return, searches, social, sponsored, temporary, tiff, to, trading, vietnam, web, welcome.

hashing, etc. Instance-level explanation methods such as SEDC can have a substantial impact on improving the process of building document classification models.

Specifically, systems like SEDC may become a critical component of the iterative process for improving document classification models. As the case study and the newsgroup study showed, SEDC can identify data quality issues and model deficiencies. Addressing these deficiencies can lead to improved models directly. Alternatively, it can lead to improved data quality, which ultimately should lead to better model performance and decision making.

This paper has not provided a rigorous study of the insight provided by the explanations. The case studies show that the method is capable of providing improved understanding of the inner workings of the classifier, and better understanding of the domain of application. It would be fascinating future work to examine changes in decision makers' judgment after having been presented with instance-level explanations.

An unexpected result of the case study was the need for various sorts of hyper-explanations. Several of these are the result of the document classification models being statistical models learned from data, and thus are subject to the main challenges of machine learning: over-fitting, under-fitting, and errors in the data. When classification errors are introduced due to these pathologies, even instance-level explanations may be inadequate (e.g., missing) or unintuitive. Hyper-explanations are needed for deep understanding, for example, showing training cases that likely led to the current model behavior.

This paper focused specifically on document classification. We conjecture that these techniques also will be quite useful in other high-dimensional classification problems, which are becoming increasingly important to modern business. For example, it may not be obvious, but classifying web users based on the web pages they visit (Provost et al. 2009; Raeder et al. 2012) could be cast in the same framework as document classification. Each user can be represented by a set of web page URLs from an extremely large set (billions). Users are classified by models over this vocabulary. Understanding their classifications is directly analogous to the problem addressed in this paper. Similarly, the problem of classifying bank customers for targeted marketing based on the parties with which they transact (Martens and Provost 2011) also can be formulated similarly. The "documents" are the customers and the "words" are the payment receivers. In both of these additional domains, being able to understand the individual classifications would have the same benefits elucidated by the extended gap model presented in the section "Explanations and Classification Models." However, the technique would not necessarily apply to every high-dimensional classification problem. It is necessary that the individual dimensions (and small subsets thereof) can be interpretable. So, in the aforementioned web-user classification example, if the URLs were irreversibly hashed for privacy reasons prior to forming the classification model, then the techniques introduced in this paper would not provide useful explanations.

Conclusion

The business problem this paper addresses is to enhance the understanding of a document classification model such that (1) the manager using it understands how decisions are being made, (2) the customers affected by the decisions can be advised why a certain action regarding them is taken, and (3) the data science/development team can improve the model iteratively. Further, (4) document classification explanations can provide better understanding of the business domain. The seven-gap extension to Kayande et al.'s three-gap framework formalizes these different roles, and shows how explanations can reduce the corresponding gaps between the user's mental model(s) and the decision system in both directions, and also can reduce the gap between the decision system and reality, as the developers use the explanations to help improve the model.

We found that global explanations in the form of a decision tree or a list of the most indicative words do not provide a satisfactory solution. Moreover, previously proposed explanation methods on the data-instance level are not able to deal with the huge dimensionality of document classification problems. With the technical constraints of high-dimensional data in mind, we addressed this business problem by creating explanations as "necessary" sets of words. Each explanation is a minimal set such that after its removal, the current classification would no longer be made. The search algorithm (SEDC) presented for finding such explanations is optimal for linear binary classification models, and heuristic for nonlinear models.

The results show that the explanations are quite concise and comprehensible, comprising a few to a few dozen words (a very small portion of the overall vocabulary). The words in the explanations vary greatly across the explanations, even with words in different languages, which supports the claim that existing global explanations are inadequate for such document classification domains.

We hope that this new sort of instance-level explanation for document classification will provide an immediately useful method across a wide variety of business (and scientific, medical, and legal) applications where document classifications are critical. We also hope we have made the case that thinking about explanations in this way opens up a large number of new research problems and opportunities for improving the state of the art in building and using data-driven document classification systems.

Acknowledgments

We thank the anonymous reviewers and editors for very constructive comments, which substantially improved the paper. We extend our gratitude to AdSafe Media and Josh Attenberg for many discussions into the problem of safe advertising. This particular data set was not necessarily used in the development of any production model used for safe advertising. Foster Provost also thanks NEC for a Faculty Fellowship.

References

Agrawal, R., and Srikant, R. 1994. "Fast Algorithms for Mining Association Rules," in *Proceedings of the 20th International Conference on Very Large Data Bases*, J. B. Bocca, M. Jarke, and C. Zaniolo (eds.), September 12-15, Santiago de Chile, Chile, pp. 487-499.

Arnold, V., Clark, N., Collier, P. A., Leech, S. A., and Sutton, S. G. 2006. "The Differential Use and Effect of Knowledge-Based System Explanations in Novice and Expert Judgement Decisions," *MIS Quarterly* (30:1), pp. 79-97.

Arnott, D. 2006. "Cognitive Biases and Decision Support Systems Development: A Design Science Approach," *Information Systems Journal* (16:1), pp. 55-78.

Attenberg, J., Ipeirotis, P., and Provost, F. 2011. "Beat the Machine: Challenging Workers to Find the Unknown Unknowns," in *Proceedings of the 3rd Human Computation Workshop*, August 8, San Francisco, pp. 1-6.

Attenberg, J., and Provost, F. 2010. "Why Label When You Can Search? Alternatives to Active Learning for Applying Human Resources to Build Classification Models under Extreme Class Imbalance," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York: ACM, pp. 423-432.

Attenberg, J., Weinberger, K. Q., Smola, A., Dasgupta, A., and Zinkevich, M. 2009. "Collaborative Email-Spam Filtering with the Hashing-Trick," in *Proceedings of the 6th Conference on Email and Anti-Spam*, July 16-17, Mountain View, CA.

Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., and Müller, K.-R. 2010. "How to Explain Individual Classification Decisions," *Journal of Machine Learning Research* (11), pp. 1803-1831.

Baesens, B., Van Gestel, T., Viaene, S., Stepanova, M., Suykens, J., and Vanthienen, J. 2003. "Benchmarking State-of-the-Art Classification Algorithms for Credit Scoring," *Journal of the Operational Research Society* (54:6), pp. 627-635.

Banker, R., and Kauffman, R. 2004. "The Evolution of Research on Information Systems: A Fiftieth-Year Survey of the Literature in *Management Science*," *Management Science* (50:3), pp. 281-298.

Barki, H., and Hartwick, J. 2001. "Interpersonal Conflict and its Management in Information System Development," *MIS Quarterly* (25:2), pp. 195-228.

Bishop, C. M. 1996. *Neural Networks for Pattern Recognition*, Oxford, UK: Oxford University Press.

Buchanan, B. G., and Shortliffe, E. H. 1984. *Rule Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*, Reading, MA: Addison Wesley.

Chang, C-C., and Lin, C-J. 2001. "LIBSVM: A Library for Support Vector Machines," Department of Computer Science and Information Systems, National Taiwan University.

Craven, M. W., and Shavlik, J. W. 1997. "Extracting Tree-Structured Representations of Trained Networks," *Advances in Neural Information Processing Systems: Proceedings of the 1996 Conference*, M. C. Mozer, J. I. Jordan, and T. Petsche (eds.), Cambridge, MA: Massachusetts Institute of Technology, pp. 24-30.

DeSanctis, G. 1983. "Expectancy Theory as Explanation of Voluntary Use of a Decision Support System," *Psychological Reports* (52), pp. 247-260.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. 1990. "Indexing by Latent Semantic Analysis," *Journal of the American society for Information Sciences* (41:6), pp. 391-407.

eMarketer. 2010. "Brand Safety Concerns Hurt Display Ad Growth," April 27 (http://www.emarketer.com/Article/ Brand-Safety-Concerns-Hurt-Display-Ad-Growth/1007661).

Fan, R-E., Chang, K-W., Hseih, C-J., Wang, W-R., and Lin, C-J. 2008. "LIBLINEAR: A Library for Large Linear Classification," *Journal of Machine Learning Research* (9), pp. 1871-1874.

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

Fayyad, U. M., Piatetsky-Shapiro, G., and Smyth, P. 1996. "From Data Mining to Knowledge Discovery: An Overview," *AI Magazine*, pp. 37-54.

Gönül, M. S., Önkal, D., and Lawrence, M. 2006. "The Effects of Structural Characteristics of Explanations on Use of a DSS," *Decision Support Systems* (42:3), pp. 1481-1493.

Gregor, S., and Benbasat, I. 1999. "Explanations from Intelligent Systems: Theoretical Foundations and Implications for Practice," *MIS Quarterly* (23:4), pp. 497-530.

Han, E-H., Karypis, G., and Kumar, V. 2001. "Text Categorization Using Weight Adjusted k-Nearest Neighbor Classification," in *Proceedings of the 5th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, London: Springer-Verlag, pp. 53-65.

Hastie, T., Tibshirani, R., and Friedman, J. 2001. *The Elements of Statistical Learning, Data Mining, Inference, and Prediction*," New York: Springer.

Hettich, S., and Bay, S. D. 1996. "The UCI KDD Archive" (http://kdd.ics.uci.edu).

Hevner, A. R., March, S. T., Park, J., and Ram. S. 2004. "Design Science in Information Systems Research," *MIS Quarterly* (28:1), pp. 75-106.

Hotho, A., Nü rnberger, A., and Paas, G. 2005. "A Brief Survey of Text Mining," *LDV Forum* (20:1), pp. 19-62.

Joachims, T. 1998. "Text Categorization with Support Vector Machines: Learning with Many Relevant Features," in *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, Berlin: Springer, pp. 137-142.

Kayande, U., De Bruyn, A., Lilien, G. L., Rangaswamy, A., and van Bruggen, G. H. 2009. "How Incorporating Feedback Mechanisms in a DSS Affects DSS Evaluations," *Information Systems Research* (20:4), pp. 527-546.

Lessmann, S., Baesens, B., Mues, C., and Pietsch, S. 2008. "Benchmarking Classification Models for Software Defect Prediction: A Proposed Framework and Novel Findings," *IEEE Transactions on Software Engineering*, (34:4), pp. 485-496.

Lilien, G. L., Rangaswamy, A., van Bruggen, G. H., and Starke, K. 2004. "DSS Effectiveness in Marketing Resource Allocation Decisions: Reality vs. Perception," *Information Systems Research* (15:3), pp. 216-235.

Limayem, M., and DeSanctis, G. 2000. "Providing Decisional Guidance for Multicriteria Decision Making in Groups," *Information Systems Research* (11:4), pp. 386-401.

Martens, D., Baesens, B., Van Gestel, T., and Vanthienen, J. 2007. "Comprehensible Credit Scoring Models Using Rule Extraction from Support Vector Machines," *European Journal of Operational Research* (183:3), pp. 1466-1476.

Martens, D., and Provost, F. 2011. "Pseudo-Social Network Targeting from Consumer Transaction Data," working paper CeDER-11-05, Stern School of Business, New York University.

Norvig, P. 2011. "On Chomsky and the Two Cultures of Statistical Learning" (http://norvig.com/chomsky.html).

Pang, B., and Lee, L. 2008. "Opinion Mining and Sentiment Analysis," *Foundations and Trends in Information Retrieval* (2:1/2), pp. 1-135.

Pant, G., and Srinivasan, P. 2005. "Learning to Crawl: Comparing Classification Schemes," *ACM Transactions on Information Systems* (23:4), pp. 430-462.

Platt, J. 1999. "Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods," *Advances in Large Margin Classifiers* (10:3), pp. 61-74.

Provost, F., Dalessandro, B., Hook, R., Zhang, X., and Murray, A. 2009. "Audience Selection for On-Line Brand Advertising: Privacy-Friendly Social Network Targeting," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York: ACM, pp. 707-716.

Qi, X., and Davison, B. D. 2009. "Web Page Classification: Features and Algorithms," *ACM Computing Surveys* (41:2), pp. 1-31.

Quinlan, J. R. 1993. *C4.5: Programs for Machine Learning*, San Francisco: Morgan Kaufmann Publishers Inc.

Raeder, T., Dalessandro, B., Stitelman, O., Perlich, C., and Provost, F. 2012. "Design Principles of Massive, Robust Prediction Systems," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 12-16, Beijing, China.

Robnik-Sikonja, M., and Kononenko, I. 2008. "Explaining Classifications for Individual Instances," *IEEE Transactions on Knowledge and Data Engineering* (20:5), pp. 589-600.

Sambamurthy, V., and Poole, M. S. 1992. "The Effects of Variations in Capabilities of GDSS Designs on Management of Cognitive Conflict in Groups," *Information Systems Research* (3:3), pp. 224-251.

Schapire, R. E., and Singer, Y. 2000. "Boostexter: A Boosting-Based System for Text Categorization," *Machine Learning* (39:2/3), pp. 135-168.

Sheng, V. S., Provost, F., and Ipeirotis, P. 2008. "Get Another Label? Improving Data Quality and Data Mining Using Multiple, Noisy Labelers," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 24-27, Las Vegas, NV.

Shmueli, G., and Koppous, O. R. 2011. "Predictive Analytics in Information Systems Research," *MIS Quarterly* (35:3), pp. 553-572.

Sidorova, A., Evangelopoulos, N., Valacich, J. S., and Ramakrishnan, T. 2008. "Uncovering the Intellectual Core of the Information Systems Discipline," *MIS Quarterly* (32:3), pp. 467-482.

Silver, M. S. 1991. "Decisional Guidance for Computer-Based Decision Support," *MIS Quarterly* (15:1), pp. 105-122.

Sindhwani, V., and Melville, P. 2008. "Document-Word Co-Regularization for Semi-Supervised Sentiment Analysis," in *Proceedings of the 8th IEEE International Conference on Data Mining*, December 15-19, Pisa, Italy, pp. 1025-1030.

Štrumbelj, E., and Kononenko, I. 2010. "An Efficient Explanation of Individual Classifications Using Game Theory," *Journal of Machine Learning Research* (11), pp. 1-18.

Štrumbelj, E., Kononenko, I., and Robnik-Sikonja, M. 2009. "Explaining Instance Classifications with Interactions of Subsets of Feature Values," *Data & Knowledge Engineering* (68:10), pp. 886-904.

Todd, P. A., and Benbasat, I. 1999. "Evaluating the Impact of DSS, Cognitive Effort, and Incentives on Strategy Selection," *Information Systems Research* (10:4), pp. 356-374.

Tseng, Y.-H., Lin, C.-J., and Lin, Y. 2007. "Text Mining Techniques for Patent Analysis," *Information Processing Management* (43:5), pp. 1216-1247.

Umanath, N. S., and Vessey, I. 1994. "Multiattribute Data Presentation and Human Judgment: A Cognitive Fit," *Decision Sciences* (25:5/6), pp. 795-824.

Vapnik, V. N. 1995. *The Nature of Statistical Learning Theory*, New York: Springer-Verlag.

Wallace, B., Trikalinos, T., Lau, J., Brodley, C., and Schmid, C. 2010. "Semi-Automated Screening of Biomedical Citations for Systematic Reviews," *BMC Bioinformatics* (11:55) (http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2824679/).

Webb, G. I. 1995. "OPUS: An Efficient Admissible Algorithm for Unordered Search," *Journal of Artificial Intelligence Research* (3), pp. 431-465.

Ye, L. R., and Johnson, P. E. 1995. "The Impact of Explanation Facilities on User Acceptance of Expert Systems Advice," *MIS Quarterly* (19:2), pp. 157-172.

About the Authors

David Martens is an assistant professor at the faculty of Applied Economics at University of Antwerp, Belgium, where he heads the Applied Data Mining research group. His research has been published in high impact international journals and is mainly focused on learning from social network data and the development of comprehensible data mining techniques. In 2008, he was a finalist for the prestigious ACM SIGKDD Doctoral Dissertation Award. Applications of his work can be found in the banking, telecommunication, and marketing domains.

Foster Provost is Professor and NEC Faculty Fellow at New York University's Stern School of Business, where he studies and teaches data science. He has served as editor-in-chief of *Machine Learning* and a program chair for the ACM SIGKDD conference. His work has won best paper awards, IBM Faculty Awards, a President's Award from NYNEX (now Verizon) Science and Technology, and the 2009 INFORMS Design Science Award.

THIS PAGE DELIBERATELY BLANK