# Jointly Minimizing the Expected Costs of Review for Responsiveness and Privilege in E-Discovery

DOUGLAS W. OARD, University of Maryland, USA
FABRIZIO SEBASTIANI, Consiglio Nazionale delle Ricerche, Italy
JYOTHI K. VINJUMUR, University of Maryland, USA

*Discovery* is an important aspect of the civil litigation process in the United States of America, in which all parties to a lawsuit are permitted to request relevant evidence from other parties. With the rapid growth of digital content, the emerging need for "e-discovery" has created a strong demand for techniques that can be used to review massive collections both for "responsiveness" (i.e., relevance) to the request and for "privilege" (i.e., presence of legally protected content that the party performing the review may have a right to withhold). In this process, the party performing the review may incur costs of two types, namely, *annotation costs* (deriving from the fact that human reviewers need to be paid for their work) and *misclassification costs* (deriving from the fact that failing to correctly determine the responsiveness or privilege of a document may adversely affect the interests of the parties in various ways). Relying exclusively on automatic classification would minimize annotation costs but could result in substantial misclassification costs, while relying exclusively on manual classification could generate the opposite consequences. This paper proposes a *risk minimization* framework (called MINECORE, for "minimizing the expected costs of review") that seeks to strike an optimal balance between these two extreme stands. In MINECORE (a) the documents are first automatically classified for both responsiveness and privilege, and then (b) some of the automatically classified documents are annotated by human reviewers for responsiveness (typically by junior reviewers) and/or, in cascade, for privilege (typically by senior reviewers), with the overall goal of minimizing the expected cost (i.e., the *risk*) of the entire process. Risk minimization is achieved by optimizing, for both responsiveness and privilege, the choice of which documents to manually review. We present a simulation study in which classes from a standard text classification test collection (RCV1-v2) are used as surrogates for responsiveness and privilege. The results indicate that MINECORE can yield substantially lower total cost than any of a set of strong baselines.

Additional Key Words and Phrases: E-discovery, Technology-Assisted Review, Utility Theory, Semi-automated Text Classification

**ACM Reference Format:**
Douglas W. Oard, Fabrizio Sebastiani, and Jyothi K. Vinjumur. 2018. Jointly Minimizing the Expected Costs of Review for Responsiveness and Privilege in E-Discovery . *ACM Transactions on Information Systems* 1, 1 (March 2018), 34 pages. https://doi.org/0000001.0000001

The method described in this paper is the subject of U.S. Provisional Patent Application number 62/518043, filed June 12, 2017. Authors are listed in alphabetical order.

Authors' addresses: Douglas W. Oard, iSchool and UMIACS, University of Maryland, College Park, MD, USA, oard@umd.edu; Fabrizio Sebastiani, Istituto di Scienza e Tecnologie dell'Informazione, Consiglio Nazionale delle Ricerche, 56124, Pisa, Italy, fabrizio.sebastiani@isti.cnr.it; Jyothi K. Vinjumur, iSchool and UMIACS, University of Maryland, College Park, MD, USA, jyothikv@umd.edu.


1046-8188/2018/3-ART $15.00
https://doi.org/0000001.0000001

## 1 INTRODUCTION

In civil litigation in the United States of America, a process referred to as *e-discovery* involves a review phase in which a set $\mathcal{D}$ of digital documents that may contain evidence that would be of interest in a specific lawsuit are reviewed to identify those which are "responsive" (i.e., relevant) to a request made by one of the parties. These documents must be "produced" (i.e., turned over) unless some "privilege" can be asserted (e.g., attorney-client privilege) [17]. Similar processes are used in other settings (e.g., regulatory investigations, criminal cases, and requests for government documents under transparency laws) and in other jurisdictions (e.g., e-disclosure in the United Kingdom).

The usual approach to e-discovery begins with one side in a case (the requesting party) submitting a "request for production" to the other side (the producing party). The producing party then conducts a search of their information systems to identify documents that are responsive to the request. This concept of *responsiveness*, the term we use in this paper, is essentially identical to relevance as understood in information retrieval, although the initial arbiter of responsiveness is the producing party, not the requesting party, because only the producing party has the right to inspect all of the documents.

Those documents that are determined to be responsive to the request are then reviewed, usually in a separate step by more highly trained (and thus more expensive) attorneys, to identify those which can properly be withheld on the basis of *privilege*, a broadly inclusive construct that subsumes several specific reasons for potentially withholding some specific content. One commonly claimed privilege is attorney-client privilege, which protects communication about litigation that occurs under rather broad (but not all-inclusive) circumstances between an attorney (or her representative) and her client (or her client's representative). Privilege is not absolute; a balancing test must be done by the courts when factors that mitigate against invoking privilege are claimed by the requesting party to exist. For this reason, the existence of documents on which privilege is asserted must be disclosed to the requesting party by entering them on a *privilege log*.

E-discovery requires that parties to a case balance three competing goals: (a) producing (to the requesting party) relevant documents, (b) withholding documents when privilege permits, and (c) performing all this efficiently. Two aspects of efficiency are important: (i) perhaps most obviously, the cost of discovery is necessarily limited by the value of the assets that are at stake in the litigation; (ii) also important, however, is that access to justice is advanced when litigants can be assured of expeditious resolution of their claims. We thus care about both cost and speed.

When most of the documents to be reviewed for responsiveness and privilege were in paper form, the review process was typically carried out manually. However, with the explosive growth in digital content, time and cost constraints now often preclude exhaustive manual review. This has led to the introduction of a number of techniques for *Technology-Assisted Review* (TAR), which may be loosely defined as a set of computerized techniques that support attorneys who need to perform an e-discovery review [3, 6, 27, 31]. An important class of TAR techniques is *predictive coding*, whereby one or more classifiers are trained (typically by means of supervised learning methods) using some manually annotated content. Once trained, these classifiers can automatically classify the remaining documents in $\mathcal{D}$ into documents to be produced and documents to be withheld.

No trained text classifier will be perfect, so attorneys will often perform some manual review of the classifier's results. If the manual review of a sample of the classifier's output reveals an unacceptably high error rate, then additional manual review would be needed. Additional training data might yield improved accuracy, but ultimately some limit will be reached beyond which an alternative strategy is needed. If the error rate that the automatic classifier ultimately achieves is still worse than what human reviewers can achieve, then additional manual review can further

decrease the overall error rate. This approach works because in e-discovery we are ultimately classifying some finite population of documents (i.e., the classifier operates not in an inductive but in a *transductive* setting – see e.g., [20]), and it is thus the accuracy of the classification decisions, and not of the classifier itself, that we care about.

Our focus in this paper is on this final step, in which it has been found that the error rate of the classifier we have generated with our best efforts is still too high, and thus some additional manual review will be needed. The key question, then, is which documents should be chosen for manual review. If the classifier were able to estimate which documents have a higher chance of being incorrectly classified, those documents would be good choices. And if some types of errors were more costly than others (e.g., if failing to withhold a privileged document were more costly than failing to produce a relevant document), then the ordering of documents for post-classification manual review should also take into account these differences in error costs. These two perspectives are complementary, and using them together yields a risk minimization framework. The goal of this paper is, in fact, the design of a risk minimization framework that accounts for the complex nature of review for responsiveness and privilege in e-discovery. To the best of our knowledge, this is the first published work which (a) addresses the use of TAR for performing review by responsiveness *and* review by privilege at the same time, (b) introduces the different costs involved in the e-discovery process (namely, the different costs of performing review by responsiveness and review by privilege, and the different costs that accrue from different ways of misclassifying a document) as explicit variables of the TAR model, and (c) (as we will see in the rest of the paper) uses *utility theory* to minimize the expected value of such costs.

The next two sections present the design of MINECORE (for "minimizing the expected costs of review"), our risk minimization method for ordering documents for post-classification manual review and for deciding when it would be prudent to end that manual review process. That description is followed by Section 4, which describes our experiment design and presents and discusses the results of those experiments. Section 5 discusses the prospects for adoption of the presented method for e-discovery, while Section 6 sets the same method in the context of related work. Section 7 concludes the paper, also discussing future work that could further extend the potential impact of these methods.

## 2 A SEMI-AUTOMATED PREDICTIVE CODING SYSTEM

We describe MINECORE, a semi-automated system whose goal is to identify, within a set of documents $\mathcal{D}$ (the "universe"), the documents that are at the same time (a) responsive to a certain topic, and (b) nonprivileged.[1] Documents that are both responsive and nonprivileged should be produced to the requesting party; documents that are responsive and privileged should be put on a *privilege log*; nonresponsive documents should be withheld. In an abstract way, our problem can thus be modelled as a *single-label classification* problem, i.e., as the problem of generating a classifier $h : \mathcal{D} \rightarrow C$, with $C = \{c_P, c_L, c_W\}$ the set of target classes, where

- $c_P$ is the class of the responsive nonprivileged documents, that should be Produced to the requesting party;
- $c_L$ is the class of the responsive privileged documents, that should be entered on the privilege Log;
- $c_W$ is the class of the nonresponsive documents, that should be Withheld by the producing party.

[1] Table 1 summarizes the mathematical notation that we are going to use in this section and in the rest of the paper.

Table 1. Notational conventions used in this paper.

| | |
|---|---|
| $d$ | A document |
| $\mathcal{D}$ | The "universe" (the set of all documents) |
| $c_r$ | The class of responsive documents |
| $c_p$ | The class of privileged documents |
| $c_P$ | The class of documents that should be Produced, with $c_P \equiv c_r \cap \overline{c}_p$ |
| $c_L$ | The class of documents that should be entered on the privilege Log, with $c_L \equiv c_r \cap c_p$ |
| $c_W$ | The class of the documents that should be Withheld, with $c_W \equiv \overline{c}_r$ |
| $\overline{c}$ | The complement in $\mathcal{D}$ of class $c$ |
| $C$ | The set $\{c_P, c_L, c_W\}$ |
| $y(d)$ | The true class of document $d$, with $y(d) \in C$ |
| $h$ | The final classifier $\mathcal{D} \rightarrow C$ |
| $\mathcal{D}_{ij}$ | The documents in $\mathcal{D}$ whose predicted class is $c_i$ and whose true class is $c_j$ |
| $D$ | The $3 \times 3$ contingency table |
| $D_{ij}$ | The number of documents in $\mathcal{D}_{ij}$ |
| $h_r$ | The text classifier that classifies for responsiveness |
| $h_p$ | The text classifier that classifies for privilege |
| $\Pr(c_r \mid d)$ | The ("posterior") probability that $d$ is responsive, as estimated by $h_r$ |
| $\Pr(c_p \mid d)$ | The ("posterior") probability that $d$ is privileged, as estimated by $h_p$ |
| $\Pr_\phi(c \mid d)$ | The ("posterior") probability that $d$ is in $c$ as computed in Phase $\phi$ |
| $h_\phi(d)$ | Class assigned to $d$ in Phase $\phi$ |
| $\Pr(c_P \mid d)$ | The probability that $d$ should be Produced |
| $\Pr(c_L \mid d)$ | The probability that $d$ should be entered on the privilege Log |
| $\Pr(c_W \mid d)$ | The probability that $d$ should be Withheld |
| $\lambda^m_{ij}$ | Unit cost of misclassifying an element of $c_j$ into $c_i$ (for $i, j \in \{P, L, W\}$) |
| $\Lambda^m$ | The $3 \times 3$ matrix of $\lambda^m_{ij}$ costs |
| $\lambda^a_r$ | Unit cost of annotating for responsiveness |
| $\lambda^a_p$ | Unit cost of annotating for privilege |
| $\Lambda^a$ | The vector $(\lambda^a_r, \lambda^a_p)$ consisting of two unit annotation costs |
| $K^m(d)$ | Misclassification cost incurred in classifying $d$ |
| $K^m(\mathcal{D})$ | Global misclassification cost incurred in classifying $\mathcal{D}$ |
| $K^a(d)$ | Annotation cost incurred in classifying $d$ |
| $K^a(\mathcal{D})$ | Global annotation cost incurred in classifying $\mathcal{D}$ |
| $R(d, c_i)$ | The risk associated with assigning $d$ to class $c_i$ |
| $R(\mathcal{D})$ | Global risk brought about by classifying $\mathcal{D}$ |
| $\mathcal{D}_r$ | Set of documents to be manually annotated for responsiveness |
| $\mathcal{D}_p$ | Set of documents to be manually annotated for privilege |
| $\tau_r = \lvert\mathcal{D}_r\rvert$ | Number of documents to be manually annotated for responsiveness |
| $\tau_p = \lvert\mathcal{D}_p\rvert$ | Number of documents to be manually annotated for privilege |
| $K^o(d)$ | Overall (annotation+misclassification) cost incurred for document $d$ |
| $K^o(\mathcal{D})$ | Overall (annotation+misclassification) cost incurred for set $\mathcal{D}$ |
| $E_y[\cdot]$ | Expected value over the $y$ random variable |
| $b$ | Batch size (in the ALvUS and ALvRS baselines) |

Our classification task[2] gives rise to the contingency table illustrated in Table 2(a), where $D_{ij}$ is the number of documents in $\mathcal{D}_{ij}$, i.e., the number of documents $d \in \mathcal{D}$ whose predicted class $h(d)$ is $c_i$ and whose true class (which we denote by $y(d)$) is $c_j$; it holds that $|\mathcal{D}| = \sum_{i,j \in \{P,L,W\}} D_{ij}$. The classes in $\{c_P, c_L, c_W\}$ can actually be defined in terms of the two "primitive" classes $c_r$ (the class of responsive documents) and $c_p$ (the class of privileged documents); i.e., we define $c_P \equiv c_r \cap \overline{c}_p$, $c_L \equiv c_r \cap c_p$, and $c_W \equiv \overline{c}_r$, where $\overline{c}_j$ denotes the complement in $\mathcal{D}$ of class $c_j$.

Our problem should actually be framed as a *cost-sensitive* single-label classification problem, since in e-discovery different classification errors bring about different costs. For instance, producing a document that should instead have been entered on the privilege log typically brings about a higher cost than producing a document that should instead have been withheld. As a consequence,

[2]The terms "reviewing", "classifying", "labeling" and "annotating", are often taken to be synonyms or near-synonyms when they refer to the attribution of a class label to a data item; the 1st of these terms is frequent in the legal world, the 2nd and 3rd in the machine learning and information retrieval communities, while the 4th is frequent, e.g., in natural language processing. In this paper we use the term "classifying" when the attribution of the class label is done by an automatic process, and "annotating" when this attribution is done by a human reviewer.

Table 2. Contingency table $D$ (a) and cost matrix $\Lambda^m$ (b) for our cost-sensitive, single-label classification problem.

| | | actual | | |
|---|---|---|---|---|
| | | $c_P$ | $c_L$ | $c_W$ |
| pred | $c_P$ | $D_{PP}$ | $D_{PL}$ | $D_{PW}$ |
| | $c_L$ | $D_{LP}$ | $D_{LL}$ | $D_{LW}$ |
| | $c_W$ | $D_{WP}$ | $D_{WL}$ | $D_{WW}$ |

(a)

| | | actual | | |
|---|---|---|---|---|
| | | $c_P$ | $c_L$ | $c_W$ |
| pred | $c_P$ | 0 | $\lambda^m_{PL}$ | $\lambda^m_{PW}$ |
| | $c_L$ | $\lambda^m_{LP}$ | 0 | $\lambda^m_{LW}$ |
| | $c_W$ | $\lambda^m_{WP}$ | $\lambda^m_{WL}$ | 0 |

(b)

we assume the existence of a cost matrix $\Lambda^m = \{\lambda^m_{ij}\}$ (for $i, j \in \{P, L, W\}$), illustrated in Table 2(b), where each entry $\lambda^m_{ij}$ (a *unit cost*) is a nonnegative value representing the cost incurred when misclassifying an element of $c_j$ into $c_i$ (the $m$ superscript stands for "misclassification"). Here, we consider all unit costs $\lambda^m_{ii}$ on the main diagonal to be 0 for all $i \in \{P, L, W\}$, since correct classification brings about no misclassification costs; all other unit costs are non null, and are input parameters to the cost-sensitive classification process and to its evaluation.

We may hypothesize two "extreme" solutions to our classification problem, a fully automated solution and a fully manual solution.

### 2.1 The fully automated solution

In the *fully automated solution* we train two automated classifiers $h_r$ (which classifies for responsiveness) and $h_p$ (which classifies for privilege), and we apply them to $\mathcal{D}$. The classifiers may be generated independently of each other, or in some sequence where some mutual dependency is exploited; the fully automated solution is essentially agnostic to how the two are trained. In particular, the documents to label for training purposes may be selected via random sampling, or keyword search, or "active learning" [33], or any combination of the above. In this work we make the simplifying assumption that training and running automated classifiers has zero cost (we defer the study of a model which relaxes this assumption to future work).

We may safely assume that $h_r$ and $h_p$ generate, for each document $d \in \mathcal{D}$, two posterior probabilities $\Pr(c_r|d)$ and $\Pr(c_p|d)$, which represent the classifiers' confidence in the fact that $d$ is responsive and that $d$ is privileged, respectively.[3] For $\Pr(c_r|d)$ a value of 1 represents total certainty that $d \in c_r$, a value of 0.5 represents total uncertainty, and a value of 0 represents total certainty that $d \in \overline{c}_r$; the same for $\Pr(c_p|d)$. Note that $\Pr(c_r|d)$ and $\Pr(c_p|d)$ are just subjective estimates generated by the classifiers, and are not probabilities in any "objective" sense (whatever this might mean).

[3]Ideally, these posterior probabilities should be "well calibrated", which is usually considered a synonym of "good-quality probabilities". Posterior probabilities $\Pr(c|d)$ are said to be *well calibrated* when, given a sample $S$ drawn from some population, $\lim_{|S|\to\infty} \frac{|\{d \in c | \Pr(c|d)=x\}|}{|\{d \in S | \Pr(c|d)=x\}|} = x$ [10]. Intuitively, this property implies that, as the size of the sample $S$ goes to infinity, e.g., 90% of the documents $d \in S$ that are assigned a well calibrated posterior probability $\Pr(c|d) = 0.9$ belong to class $c$. Some classifiers are known to return well calibrated probabilities (e.g., classifiers trained via logistic regression [40]). The posterior probabilities returned by some other classifiers are known instead to be not well calibrated (e.g., this is the case of the naïve Bayesian classifier [11]). Yet some other classifiers (e.g., those trained via SVMs) do not return posterior probabilities, but generic confidence scores. In these two last cases it is possible to map the obtained posterior probabilities / confidence scores into well calibrated posterior probabilities via some "calibration" method [28, 40]; see also Section 4.2 for more on this.

We also make the assumption that $c_r$ and $c_p$ are stochastically independent, an assumption which is largely verified in practical e-discovery scenarios. A consequence of this assumption is that

$$\begin{aligned} \Pr(c_P|d) &= \Pr(c_r|d)\Pr(\overline{c}_p|d) \\ \Pr(c_L|d) &= \Pr(c_r|d)\Pr(c_p|d) \\ \Pr(c_W|d) &= \Pr(\overline{c}_r|d) \end{aligned} \tag{1}$$

We take a *risk minimization* approach (as from normative – a.k.a. "prescriptive" – decision theory [1]) and classify each document $d$ in the class

$$h(d) = \arg\min_{c_i} R(d, c_i) \tag{2}$$

where $R(d, c_i)$ (the risk associated with assigning $d$ to class $c_i$) is defined as

$$R(d, c_i) = \sum_{j \in \{P,L,W\}} \lambda^m_{ij} \Pr(c_j|d) \tag{3}$$

As a result, the *global risk* brought about by this classification is

$$R(\mathcal{D}) = \sum_{d \in \mathcal{D}} R(d, h(d)) \tag{4}$$

In other words, to each document $d$ we assign the class that brings about the minimum expected misclassification cost (i.e., the minimum *misclassification risk*) for $d$, where expected misclassification cost is the sum of the misclassification costs of all possible events (i.e., classes to which $d$ might truly belong), each multiplied by the probability of occurrence of the event (which is estimated by the classifier).

The notion of risk arises naturally in a cost-sensitive classification context, since many courses of action (or "events", in the terminology of probability theory) we may opt for (e.g., deciding to enter a certain document on the privilege log) are taken under uncertainty (e.g., we do not know for certain if this document should be entered on the privilege log), and each course of action has its own cost (e.g., incurring a sanction for having entered on the privilege log a document that should instead have been produced). Minimizing risk involves, for example, avoiding courses of actions for which a combination of probability of occurrence and cost is high. Here, the notion of "risk" $R(d, c_i)$ is the converse of the notion of *utility*; one usually speaks of "risk" when each of the possible events has an associated *cost* (i.e., an undesired consequence), whereas one usually speaks of "utility" when each possible event has an associated *gain* (i.e., a desired consequence). Anyway, the two notions are interchangeable; we prefer speaking of "risk" here since the entire process involves costs, and not gains, for the producing party, and it is the expectation over these costs that we want to minimize.

As a function for measuring misclassification cost, it is quite natural to use

$$K^m(\mathcal{D}) = \sum_{i,j \in \{P,L,W\}} \lambda^m_{ij} D_{ij} \tag{5}$$

where the $m$ superscript stands for "misclassification". Note that $K^m(\mathcal{D})$ is linear, i.e., it can alternatively be written as $K^m(\mathcal{D}) = \sum_{d \in \mathcal{D}} K^m(d)$, where $K^m(d) = \lambda^m_{h(d)y(d)}$ is the cost of predicting document $d$ to be in class $h(d)$ when its true class is $y(d)$.

### 2.2 The fully manual solution

In the *fully manual solution* a reviewer (typically: a junior lawyer) annotates all documents in $\mathcal{D}$ for responsiveness. All the documents in $\mathcal{D}$ that the reviewer deems responsive are forwarded to another reviewer (typically: a senior lawyer) who annotates them for privilege, while all the others

are withheld. All the documents that this latter reviewer deems nonprivileged are produced to the requesting party, while all the documents that she deems privileged are entered on the privilege log.[4] In this work we make the simplifying assumption that our reviewers are perfectly reliable (i.e., they do not make annotation errors); we defer the study of a model which relaxes this assumption to future work.

Let the pair $\Lambda^a = (\lambda_r^a, \lambda_p^a)$ denote the costs of annotating a single document for responsiveness ($\lambda_r^a$) and for privilege ($\lambda_p^a$), where the $a$ superscript stands for "annotation". As a function for measuring annotation cost (which derives from the intervention of human reviewers) it is quite natural to use

$$K^a(\mathcal{D}) = \lambda_r^a \tau_r + \lambda_p^a \tau_p \tag{6}$$

where $\tau_r$ and $\tau_p$ are the numbers of documents manually annotated for responsiveness and for privilege, respectively.

Note that for the fully manual solution, $\tau_r$ is the number of documents in $\mathcal{D}$, and $\tau_p$ is the number of responsive documents in $\mathcal{D}$. Similarly to the cost matrix $\Lambda^m$, we assume the unit costs in $\Lambda^a$ to be input parameters, since they are not under the control of the experimenter.

### 2.3 Our hybrid solution

Both the fully automated solution and the fully manual solution have drawbacks. The fully automated solution has the advantage of zero annotation cost (given our simplifying assumption), but is disadvantageous because automated classifiers have a non-negligible misclassification cost. This will result, e.g., in withholding documents that should have been produced and producing documents that should have been withheld; and the cost generated by too many such misclassifications might be prohibitive. The fully manual solution has the advantage of zero misclassification cost (given our simplifying assumption) but (i) is expensive, since the costs involved in manual annotation are high, and (ii) is sometimes infeasible, since it might be impossible to manually annotate each document given the time constraints imposed by the legal process.

We try to strike a balance between the two, and devise a three-phase hybrid model where

(1) for each document in $\mathcal{D}$, two classifiers $h_r$ and $h_p$ first compute probabilities $\Pr(c_r|d)$ and $\Pr(c_p|d)$, respectively; following this, $d$ is assigned to a class $h(d) \in \{c_P, c_L, c_W\}$ using Equation 2;

(2) (junior) human reviewers then annotate for responsiveness a subset $\mathcal{D}_r$ of the documents in $\mathcal{D}$; for each $d \in \mathcal{D}_r$
   (a) 0 or 1 is assigned to $\Pr(c_r|d)$ if the reviewer has deemed $d$ responsive or nonresponsive, respectively;
   (b) $h(d) \in \{c_P, c_L, c_W\}$ is recomputed using Equation 2; this may cause $d$ to be reassigned to a class in $\{c_P, c_L, c_W\}$ different from the class currently assigned to it;

(3) (senior) human reviewers annotate for privilege a subset $\mathcal{D}_p$ of the documents in $\mathcal{D}$; for each $d \in \mathcal{D}_p$

[4]Note that in this solution the two reviewers work sequentially, rather than in parallel. This is justified by cost issues, i.e., (a) by the fact that it is a waste of resources to annotate by privilege a document that has already been ruled out on counts of responsiveness, and (b) by the fact that the reviewers who deal with responsiveness usually work at cheaper hourly rates than the reviewers who deal with privilege. This suggests to have a first pass carried out by the former before the latter intervene. We also assume, for ease of explanation, that there is only one reviewer for responsiveness and only one reviewer for privilege. In real applications there are often several reviewers of each type; however, what we describe straightforwardly applies to the case of more than one reviewer of each type.

(a) 0 or 1 is assigned to $\Pr(c_p|d)$ if the reviewer has deemed $d$ privileged or nonprivileged, respectively;
(b) $h(d) \in \{c_P, c_L, c_W\}$ is recomputed using Equation 2; again, this may cause $d$ to be reassigned a class in $\{c_P, c_L, c_W\}$ different from the class currently assigned to it.

There are no constraints on the relationship between $\mathcal{D}_r$ and $\mathcal{D}_p$. Therefore, a document $d$ may belong to just one of $\mathcal{D}_r$ and $\mathcal{D}_p$, or it may belong to both, or it may belong to neither. If $d$ belongs to neither, then the class initially assigned in Step 1 is never changed, and remains $d$'s definitively assigned class. If $d$ belongs to both, a new class may be reassigned to $d$ in Step 2 and yet another class may be reassigned to it in Step 3.

We will call this hybrid model MINECORE (for "minimizing the expected costs of review"). Note that the fully automated solution described in Section 2.1 coincides with Phase 1 of MINECORE; in other words, in MINECORE we employ human reviewers to revise, according to a risk minimization principle, some of the labels generated by the fully automated solution.

Of course, the right question here is how to strike an *optimal* balance, i.e., how to decide (i) which documents should be annotated (by the junior reviewers) in Phase 2, (ii) which should be annotated (by the senior reviewers) in Phase 3, and (iii) which others should instead be left unchecked. Our solution to striking such a balance makes use of

- the posterior probabilities $\Pr(c_r|d)$ and $\Pr(c_p|d)$ generated by the automated classifiers $h_r$ and $h_p$;
- the cost matrix $\Lambda^m$ and the pair $\Lambda^a$ of unit annotation costs.

From now on, by the term *cost structure* we indicate a pair $\Lambda = (\Lambda^m, \Lambda^a)$, with $\Lambda^m$ a cost matrix and $\Lambda^a$ a pair $(\lambda^a_r, \lambda^a_p)$ of unit annotation costs. The only constraints we impose on $\Lambda$ are that (i) all unit misclassification costs in $\Lambda^m$ and both unit annotation costs in $\Lambda^a$ must be nonnegative; (ii) all $\lambda^m_{ii} \in \Lambda^m$ must be 0; and (iii) it must hold that $\lambda^a_r \leq \lambda^a_p$. The rationale of constraint (iii) is discussed in Section 3.4.

The overall cost of any hybrid process can be quantified as

$$K^o(\mathcal{D}) = K^m(\mathcal{D}) + K^a(\mathcal{D}) \tag{7}$$

where the $o$ superscript stands for "overall", and where $K^m(\mathcal{D})$ and $K^a(\mathcal{D})$ are the costs defined in Equations 5 and 6. $K^o(\mathcal{D})$ is the evaluation function we adopt in this work for all systems we experimentally compare, and not just for MINECORE. Note that for the fully automated solution $K^o(\mathcal{D})$ coincides with $K^m(\mathcal{D})$, since for this solution we have assumed the annotation cost to be zero, and for the fully manual solution $K^o(\mathcal{D})$ coincides with $K^a(\mathcal{D})$, since for this solution we have assumed the misclassification cost to be zero.

Function $K^o(\mathcal{D})$ in Equation 7 is linear, since both $K^m(\mathcal{D})$ and $K^a(\mathcal{D})$ are linear; we can thus focus on the cost

$$K^o(d) = K^m(d) + K^a(d) \tag{8}$$

brought about by an individual document $d$. In MINECORE, when we need to decide if a document $d$ should be in $\mathcal{D}_r$ (i.e., should be annotated for responsiveness in Phase 2) and if a document $d$ should be in $\mathcal{D}_p$ (i.e., should be annotated for privilege in Phase 3), we do not know the true class $y(d)$ of $d$, so we cannot quantify $K^m(d)$ (hence $K^o(d)$) precisely. In developing our hybrid method we thus again use a risk minimization approach, where we try to minimize *an expectation* of the overall cost described in Equation 8; i.e., we want to minimize

$$E_y[K^o(d)] = E_y[K^m(d) + K^a(d)] \tag{9}$$

where $E[\cdot]$ stands for "expected value" and the $y$ index indicates that the expectation is taken over the $y(d)$ random variable (i.e., over the values that the true class of $d$ could take). Note that

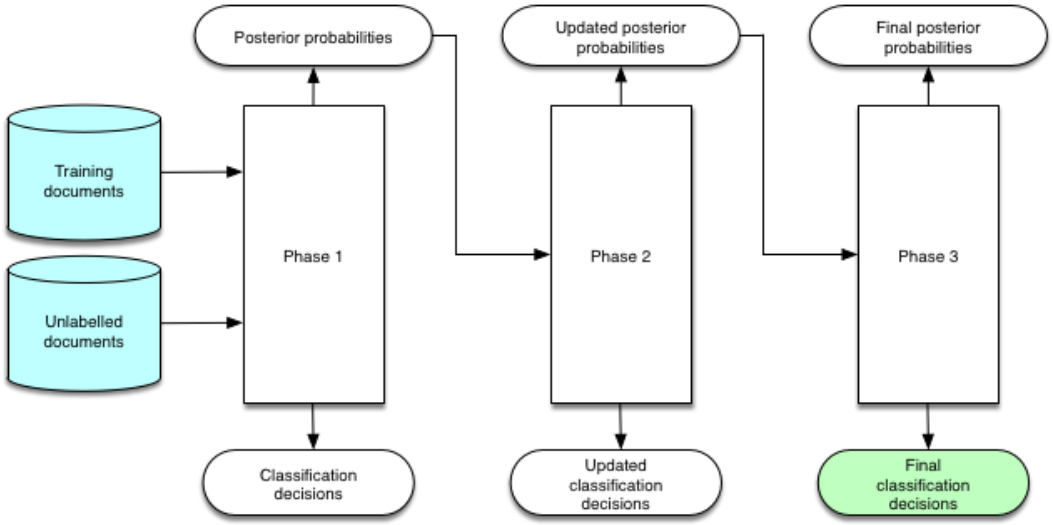


Fig. 1. The MINECORE flowchart.

minimizing $E_y[K^m(d) + K^a(d)]$ cannot be obtained by independently minimizing $E_y[K^m(d)]$ and $E_y[K^a(d)]$, since $K^m(d)$ and $K^a(d)$ are not independent. That is, we can easily minimize $K^m(d)$, by choosing to manually annotate $d$; however, if this is done for all $d \in \mathcal{D}$, $K^a(\mathcal{D})$ could be very high. Conversely, we can easily minimize $K^a(d)$, by choosing to automatically classify $d$; however, if this is done for all $d \in \mathcal{D}$, $K^m(\mathcal{D})$ could be very high. The next section thus documents our approach to *jointly* minimizing $E_y[K^m(d)]$ and $E_y[K^a(d)]$.

## 3 JOINTLY MINIMIZING EXPECTED ANNOTATION COSTS AND EXPECTED MISCLASSIFICATION COSTS

As hinted in Section 2.3, MINECORE essentially consists of an automatic classification phase (Phase 1), followed by two human annotation phases (Phase 2 and Phase 3) in which only the documents whose manual annotation is expected to reduce the overall cost are annotated.

For each phase $\phi$ and for each document $d$, two posterior probabilities $\Pr_\phi(c_r|d)$ and $\Pr_\phi(c_p|d)$ are generated. Based on these probabilities, a class $h_\phi(d)$ is assigned in Phase $\phi$ to each document $d$ as

$$\begin{aligned} h_\phi(d) &= \arg\min_{c_i} R_\phi(d, c_i) \\ &= \arg\min_{c_i} \sum_{j \in \{P,L,W\}} \lambda^m_{ij} \Pr_\phi(c_j|d) \end{aligned} \tag{10}$$

where $c_i$ ranges on $\{c_P, c_L, c_W\}$. Equation 10 is just Equation 2 where the phase $\phi$ in which the probabilities are computed and the class is assigned is made explicit.

The architecture of MINECORE is displayed in Figure 1.

### 3.1 Phase 1: Classification

In Phase 1 of MINECORE (see Figure 2 for a visual depiction) we train two automated classifiers, $h_r$ (which classifies according to responsiveness) and $h_p$ (which classifies according to privilege), from training data that we assume to be available, and we apply them to $\mathcal{D}$.

As in the fully automated solution described in Section 2.1, we assume that these two classifiers generate, for each document $d \in \mathcal{D}$, two posterior probabilities $\Pr_1(c_r|d)$ and $\Pr_1(c_p|d)$, which represent the classifiers' confidence in the fact that $d$ is responsive and that $d$ is privileged, respectively. Using these posterior probabilities, we assign a class $h_1(d) \in \{c_P, c_L, c_W\}$ to each document $d \in \mathcal{D}$ using Equation 10.

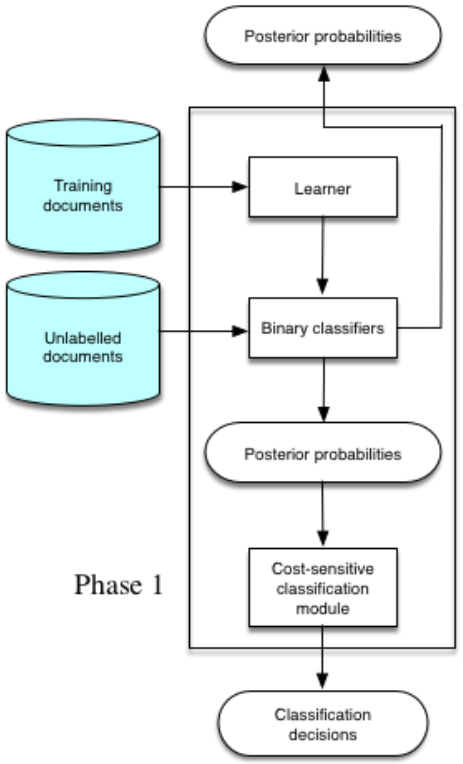


Fig. 2. Dataflow diagram for Phase 1.

### 3.2 Phase 2: Annotating for Responsiveness

In Phase 2 of MINECORE (see Figure 3 (a) for a visual depiction) the documents in $\mathcal{D}$ are ranked, and the reviewer (typically: a junior lawyer) annotates the top-ranked $\tau_r$ documents for responsiveness. Annotating $d$ has the effect of eliminating the uncertainty on the responsiveness of $d$. As a consequence, if $d$ is annotated as responsive we set $\Pr_2(c_r|d) = 1$, while if $d$ is annotated as nonresponsive we set $\Pr_2(c_r|d) = 0$; no annotation for privilege is performed in this phase, so $\Pr_1(c_p|d) = \Pr_2(c_p|d)$. At this point, by using Equation 10, $d$ is assigned a class $h_2(d) \in \{c_P, c_L, c_W\}$, which is possibly different from $h_1(d)$.

The documents $d$ from the $(\tau_r+1)$-th position onwards are not manually annotated; everything remains unchanged for these documents, i.e., $\Pr_2(c_r|d) = \Pr_1(c_r|d)$ and $\Pr_2(c_p|d) = \Pr_1(c_p|d)$, which implies that $h_2(d) = h_1(d)$.

In order to maximize the cost-effectiveness of this approach it is necessary to choose (i) an optimal ranking of the documents in $\mathcal{D}$ and (ii) an optimal threshold $\tau_r$ (which acts as the stopping condition for the annotation process).

Concerning point (i), similarly to the approach of [5] we adopt the principle that the documents in $\mathcal{D}$ are to be ranked in terms of the reduction in overall risk that annotating the document brings about; the documents whose manual annotation brings about the highest reduction are top-ranked. If by $K^m_\phi(d)$ we indicate the misclassification cost brought about by attributing class $h_\phi(d)$ to $d$, the difference in overall cost that annotating $d$ for responsiveness brings about can be written (using Equation 8) as

$$\begin{aligned}\Delta^{or}(d) &= K^o_2(d) - K^o_1(d) \\ &= K^m_2(d) + K^a_2(d) - K^m_1(d) - K^a_1(d) \\ &= K^m_2(d) + \lambda^a_r - K^m_1(d)\end{aligned} \tag{11}$$

However, as discussed in Section 2.3, at the time of ranking $\mathcal{D}$ the true class of $d$ (noted as $y(d)$) is not known, so $K^m_1(d)$ and $K^m_2(d)$ are also unknown. Therefore, at the time of ranking $\mathcal{D}$ what we can actually compute, instead of $\Delta^{or}(d)$, is an *expectation* of $\Delta^{or}(d)$ over the $y(d)$ random variable,

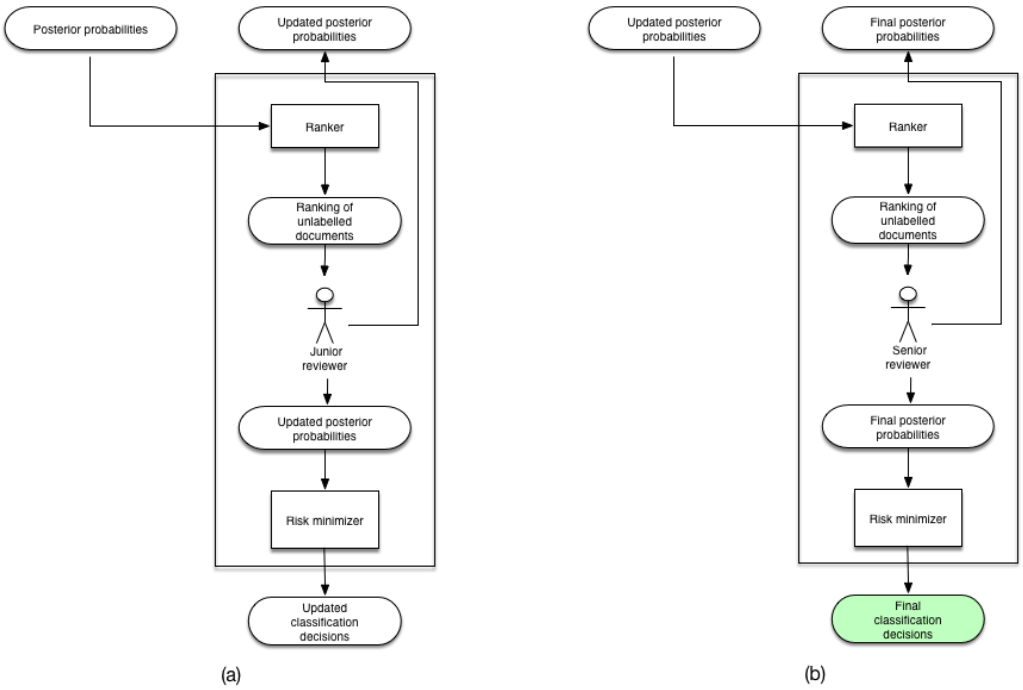


Fig. 3. Dataflow diagrams for Phase 2 (a) and Phase 3 (b).

i.e.,

$$\begin{aligned} E_y[\Delta^{or}(d)] &= E_y[K_2^m(d) + \lambda_r^a - K_1^m(d)] \\ &= E_y[K_2^m(d)] + \lambda_r^a - E_y[K_1^m(d)] \\ &= R_2(d, h_2(d)) + \lambda_r^a - R_1(d, h_1(d)) \end{aligned} \tag{12}$$

Actually, at the time of ranking $\mathcal{D}$ we also do not know the value of the $y_r(d)$ variable (a binary variable that indicates whether, if the reviewer had to annotate $d$, she would deem it responsive or not). This means that also the class $h_2(d)$ that would be assigned as a result of annotating $d$ is not known. $R_2(d, h_2(d))$ is thus not known either, which means that Equation 12 cannot be used directly as a criterion for ranking $\mathcal{D}$. At the time of ranking $\mathcal{D}$ we thus must compute an expectation of $E_y[\Delta^{or}(d)]$ over the $y_r(d)$ random variable, i.e.,

$$\begin{aligned} E_{y_r y}[\Delta^{or}(d)] &= E_{y_r}[R_2(d, h_2(d)) + \lambda_r^a - R_1(d, h_1(d))] \\ &= E_{y_r}[R_2(d, h_2(d))] + \lambda_r^a - R_1(d, h_1(d)) \end{aligned} \tag{13}$$

where we have shortened $E_{y_r}[E_y[\cdot]]$ as $E_{y_r y}[\cdot]$, and where the last simplification is justified by the fact that $R_1(d, h_1(d))$ does not depend on $y_r(d)$.

$E_{y_r}[R_2(d, h_2(d))]$ is computed by assigning probabilities to the events $c_r$ (i.e., "the reviewer annotates $d$ as responsive") and $\overline{c}_r$ ("the reviewer annotates $d$ as nonresponsive"). To do this, the best we can do is to "trust" our classifiers and assume that $d$ will be annotated as responsive with probability $\Pr_1(c_r|d)$ and nonresponsive with probability $\Pr_1(\overline{c}_r|d)$. Each of these probabilities is multiplied by the misclassification risk that the annotation would bring about, i.e.,

$$E_{y_r}[R_2(d, h_2(d))] = R_2(d, h_2(d)|c_r) \cdot \Pr_1(c_r|d) + R_2(d, h_2(d)|\overline{c}_r) \cdot \Pr_1(\overline{c}_r|d) \tag{14}$$

where by $R_2(d, h_2(d)|c_r)$ we indicate the misclassification risk that would result from assuming that $\Pr_2(c_r|d) = 1$ and $\Pr_2(c_p|d) = \Pr_1(c_p|d)$, and by $R_2(d, h_2(d)|\bar{c}_r)$ we indicate the misclassification risk that would result from assuming that $\Pr_2(c_r|d) = 0$ and $\Pr_2(c_p|d) = \Pr_1(c_p|d)$.

Equation 13 finally gives us a concrete method for ranking the automatically classified documents: for each $d \in \mathcal{D}$ compute $E_{y_r y}[\Delta^{or}(d)]$ (the expected increase in overall cost brought about by annotating $d$ for responsiveness), and rank the documents in $\mathcal{D}$ according to their $E_{y_r y}[\Delta^{or}(d)]$ score, top-ranking those with the *lowest* scores. This guarantees that the reviewer will first annotate the documents characterized by the highest expected *reduction* in cost that manually annotating them would bring about. In turn this guarantees that, whatever the amount $\tau_r$ of documents that the reviewers annotate, the expected cost-effectiveness of the annotation work will be maximized.

Equation 13 gives us also a concrete method for addressing point (ii) above, i.e., for setting the $\tau_r$ threshold. The overall cost $K^o(d)$ is expected to decrease as a result of annotating $d$ (i.e., $E_{y_r y}[\Delta^{or}(d)] < 0$) when the cost $\lambda_r^a$ of annotating $d$ is more than offset by the expected reduction $(R_1(d, h_1(d))) - E_{y_r}[R_2(d, h_2(d))]$ in misclassification cost that annotating $d$ brings about; conversely, if $E_{y_r y}[\Delta^{or}(d)] \geq 0$, then the expected reduction in misclassification cost is not worth the additional annotation effort. Therefore, the criterion we adopt is in order to decide when to stop annotating is:

> **Stopping condition (responsiveness)**. Let $d$ be the document at the $k$-th rank position. If $E_{y_r y}[\Delta^{or}(d)] < 0$, then annotate $d$ by responsiveness and move on to the document in the $(k+1)$-th rank position, else stop annotating.

The rationale for this criterion is that a reviewer will annotate a document only if this action is expected to diminish overall cost. Since the likelihood of diminishing overall cost decreases the more we go down the ranking, it follows that we should choose $\tau_r$ to be

$$\tau_r = |\{d|E_{y_r y}[\Delta^{or}(d)] < 0\}| \tag{15}$$

We now offer some concrete examples to show how Phase 2 works.

*Example 3.1. Table 3 shows some example documents processed in Phase 2.*

*The upper table shows the cost structure that we use in the lower table (the specific values were chosen for clarity of illustration, and are not assumed to be realistic). In the lower table, each example document d is represented as a triplet of rows. The 1st row of each triplet shows the values of* $h(d)$ *(Column 10) and* $R(d, h(d))$ *(Column 11) that result from the posterior probabilities* $\Pr_1(c_r|d)$ *and* $\Pr_1(c_p|d)$ *(Columns 2 and 3) returned by the automated classifiers of Phase 1. The 2nd and 3rd row of each triplet show instead the values of* $h(d)$ *and* $R(d, h(d))$ *that would result if the document were manually annotated as responsive (2nd row) or unresponsive (3rd row), which would cause* $\Pr_2(c_r|d)$ *to become 1 or 0, respectively (Column 2). Column 12 represents* $E_{y_r}[R(d)]$*, the expected cost of d after annotation for responsiveness, while Column 13 represents* $E_{y_r y}[\Delta^{or}(d)]$*, the expected reduction in the cost of d that annotating it by responsiveness would bring about. Column 14 indicates whether, as a result of the difference between the value in Column 13 and* $\lambda_r^a$*, it is decided to annotate d for responsiveness. The 3rd row of the header indicates the equations according to which the values in the respective columns are computed.*

*Let us look at some specific examples.*

*For* $d_1$*, the difference between the 1st and 2nd rows shows that a change in* $P(c_r|d)$ *can bring about a change between which of* $c_P$ *and* $c_L$ *is picked. This is interesting, since at first sight we might think that the decision whether to produce the document or file it into the privilege log should only depend on privilege-related considerations, since both actions concern documents whose responsiveness has been ascertained already; Equation 2 is the reason why this does not necessarily happen.*

*For* $d_2$*, annotation by responsiveness is expected to bring about a benefit in terms of misclassification risk, but not high enough to offset the cost of annotating the document.*

Table 3. Example documents as processed via MINECORE in Phase 2.

| $\lambda_r^a$ | $\lambda_p^a$ | $\lambda_{PP}^m$ | $\lambda_{PL}^m$ | $\lambda_{PW}^m$ | $\lambda_{LP}^m$ | $\lambda_{LL}^m$ | $\lambda_{LW}^m$ | $\lambda_{WP}^m$ | $\lambda_{WL}^m$ | $\lambda_{WW}^m$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 0 | 5 | 3 | 8 | 0 | 45 | 3 | 13 | 0 |

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\Pr(c_r\|d)$ | $\Pr(c_p\|d)$ | $\Pr(c_P\|d)$ | $\Pr(c_L\|d)$ | $\Pr(c_W\|d)$ | $R(d,c_P)$ | $R(d,c_L)$ | $R(d,c_W)$ | $h(d)$ | $R(d,h(d))$ | $E_{y_r}[R(d)]$ | $E_{y_r y}[\Delta^{or}(d)]$ | Annotate? |
| | | | Eq. 1 | Eq. 1 | Eq. 1 | Eq. 3 | Eq. 3 | Eq. 3 | Eq. 2 | Eq. 3 | Eq. 14 | Eq. 13 | |
| $d_1$ | 0.90 | 0.70 | 0.27 | 0.63 | 0.10 | 3.45 | 6.66 | 9.00 | $c_P$ | 3.45 | 2.16 | -1.29 | Yes |
| | 1.00 | 0.70 | 0.30 | 0.70 | 0.00 | 3.50 | 2.40 | 10.00 | $c_L$ | 2.40 | | | |
| | 0.00 | 0.70 | 0.00 | 0.00 | 1.00 | 3.00 | 45.00 | 0.00 | $c_W$ | 0.00 | | | |
| $d_2$ | 0.85 | 0.25 | 0.63 | 0.21 | 0.15 | 1.51 | 11.85 | 4.67 | $c_P$ | 1.51 | 1.06 | -0.45 | No |
| | 1.00 | 0.25 | 0.75 | 0.25 | 0.00 | 1.25 | 6.00 | 5.50 | $c_P$ | 1.25 | | | |
| | 0.00 | 0.25 | 0.00 | 0.00 | 1.00 | 3.00 | 45.00 | 0.00 | $c_W$ | 0.00 | | | |
| $d_3$ | 0.22 | 0.45 | 0.12 | 0.09 | 0.78 | 2.83 | 36.06 | 1.65 | $c_W$ | 1.65 | 0.49 | -1.15 | Yes |
| | 1.00 | 0.45 | 0.55 | 0.45 | 0.00 | 2.25 | 4.40 | 7.50 | $c_P$ | 2.25 | | | |
| | 0.00 | 0.45 | 0.00 | 0.00 | 1.00 | 3.00 | 45.00 | 0.00 | $c_W$ | 0.00 | | | |
| $d_4$ | 0.00 | 0.70 | 0.00 | 0.00 | 1.00 | 3.00 | 45.00 | 0.00 | $c_W$ | 0.00 | 0.00 | 0.00 | No |
| | 1.00 | 0.70 | 0.30 | 0.70 | 0.00 | 3.50 | 2.40 | 10.00 | $c_L$ | 2.40 | | | |
| | 0.00 | 0.70 | 0.00 | 0.00 | 1.00 | 3.00 | 45.00 | 0.00 | $c_W$ | 0.00 | | | |
| $d_5$ | 1.00 | 0.70 | 0.30 | 0.70 | 0.00 | 3.50 | 2.40 | 10.00 | $c_L$ | 2.40 | 2.40 | 0.00 | No |
| | 1.00 | 0.70 | 0.30 | 0.70 | 0.00 | 3.50 | 2.40 | 10.00 | $c_L$ | 2.40 | | | |
| | 0.00 | 0.70 | 0.00 | 0.00 | 1.00 | 3.00 | 45.00 | 0.00 | $c_W$ | 0.00 | | | |

*An interesting fact that example $d_3$ shows is that decreasing uncertainty does not always result in decreasing risk: the difference between the values of $R(d, h(d))$ of the 1st and 2nd rows shows that, if the reviewer annotated $d_3$ as responsive, $R(d, h(d))$ would actually increase. This may happen, for example, when we rule out the possibility that $d$ belongs to a class $c_i$ that is not "risky" (i.e., a class whose $\lambda_{ij}^m$'s are all low), thereby increasing the probability that $d$ belongs to other "riskier" classes.*

*Document $d_4$ represents an extreme case, since the classifier claims to be already certain that $d_4$ is nonresponsive. Therefore, having $d_4$ annotated by responsiveness is expected not to bring about any advantage, since the model assumes that the reviewer will certainly confirm $d_4$ to be nonresponsive. Similar comments may be made for $d_5$, which the model assumes to be certainly responsive.* □

### 3.3 Phase 3: Annotating for Privilege

At this point, in Phase 2 the human reviewer has manually annotated the $\tau_r$ documents characterized by the lowest value of $E_{y_r y}[\Delta^{or}(d)]$. Phase 3 can now start.

Phase 3 of MINECORE (see Figure 3 (b) for a visual depiction) does for privilege essentially what Phase 2 did for responsiveness; the steps we go through in this section mimic fairly closely those described in Section 3.2, and are thus described more concisely.

In Phase 3 the documents in $\mathcal{D}$ are again ranked, and the reviewer (typically: a senior lawyer) annotates the top-ranked $\tau_p$ documents for privilege.[5] If the reviewer annotates $d$ as privileged we set $\Pr_3(c_p|d) = 1$, while if the reviewer annotates $d$ as nonprivileged we set $\Pr_3(c_p|d) = 0$; no annotation for responsiveness is performed in this phase, so $\Pr_2(c_r|d) = \Pr_3(c_r|d)$. At this point, by using Equation 10, $d$ is assigned a class $h_3(d) \in \{c_P, c_L, c_W\}$, which is possibly different from $h_2(d)$. The documents $d$ from the $(\tau_p + 1)$-th position onwards are not manually annotated for privilege; for these documents, $\Pr_3(c_r|d) = \Pr_2(c_r|d)$ and $\Pr_3(c_p|d) = \Pr_2(c_p|d)$, which implies that $h_3(d) = h_2(d)$. Class $h_3(d) \in \{c_P, c_L, c_W\}$ is the final class assigned to $d$ by MINECORE, and the class that determines whether the document is produced to the requesting party ($h_3(d) = c_P$), entered on the privilege log ($h_3(d) = c_L$), or withheld ($h_3(d) = c_W$).

The difference $\Delta^{op}(d)$ in overall cost that annotating $d$ for privilege brings about is

$$\begin{aligned} \Delta^{op}(d) &= K_3^o(d) - K_2^o(d) \\ &= K_3^m(d) + K_3^a(d) - K_2^m(d) - K_2^a(d) \\ &= K_3^m(d) + \lambda_p^a - K_2^m(d) \end{aligned} \tag{16}$$

Similarly to Equation 11, and for the same reasons, Equation 16 cannot be used directly as a criterion for ranking $\mathcal{D}$. At the time of ranking $\mathcal{D}$ we thus compute the expected difference in cost

$$\begin{aligned} E_y[\Delta^{op}(d)] &= E_y[K_3^m(d) + \lambda_p^a - K_2^m(d)] \\ &= E_y[K_3^m(d)] + \lambda_p^a - E_y[K_2^m(d)] \\ &= R_3(d, h_3(d)) + \lambda_p^a - R_2(d, h_2(d)) \end{aligned} \tag{17}$$

Due to the fact that the value of $y_p(d)$ (a binary variable that indicates whether, if the reviewer had to annotate $d$, she would deem it privileged or not) is not known at the time of ranking, we must compute an expectation of $E_y[\Delta^{op}(d)]$ over the $y_p(d)$ random variable, i.e.,

$$\begin{aligned} E_{y_p y}[\Delta^{op}(d)] &= E_{y_p}[R_3(d, h_3(d)) + \lambda_p^a - R_2(d, h_2(d))] \\ &= E_{y_p}[R_3(d, h_3(d))] + \lambda_p^a - R_2(d, h_2(d)) \end{aligned} \tag{18}$$

where we have shortened $E_{y_p}[E_y[\cdot]]$ as $E_{y_p y}[\cdot]$. To compute $E_{y_p}[R_3(d, h_3(d))]$, we assume that $d$ will be annotated as privileged with probability $\Pr_1(c_p|d)$ and nonprivileged with probability $\Pr_1(\overline{c}_p|d)$, thus bringing about

$$E_{y_p}[R_3(d, h_3(d))] = R_3(d, h_3(d)|c_p) \cdot \Pr_1(c_p|d) + R_3(d, h_3(d)|\overline{c}_p) \cdot \Pr_1(\overline{c}_p|d) \tag{19}$$

Analogously to Equation 13, Equation 18 now gives us a concrete method for ranking the documents: rank the documents in $\mathcal{D}$ according to their $E_{y_p y}[\Delta^{op}(d)]$ score, top-ranking those with the lowest scores. The same equation also gives us a concrete method for setting the $\tau_p$ threshold: along the same lines discussed for Phase 2, the criterion we adopt in order to decide when to stop annotating is:

> **Stopping condition (privilege).** Let $d$ be the document at the $k$-th rank position. If $E_{y_p y}[\Delta^{op}(d)] < 0$, then manually annotate $d$ by privilege and move on to the document in the $(k + 1)$-th rank position, else stop annotating.

and we should choose $\tau_p$ to be

$$\tau_p = |\{d|E_{y_p y}[\Delta^{op}(d)] < 0\}| \tag{20}$$

[5] In an operational setting, the senior lawyer performing this final review might also correct any false positive annotations for responsiveness that they notice, but we do not presently model corrections to responsiveness that might be made during Phase 3.

Table 4. The same example documents from Table 3 as further processed via MINECORE in Phase 3; the structure of the tables is the same as those of Example 3.

| $\lambda_r^a$ | $\lambda_p^a$ | $\lambda_{PP}^m$ | $\lambda_{PL}^m$ | $\lambda_{PW}^m$ | $\lambda_{LP}^m$ | $\lambda_{LL}^m$ | $\lambda_{LW}^m$ | $\lambda_{WP}^m$ | $\lambda_{WL}^m$ | $\lambda_{WW}^m$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 0 | 5 | 3 | 8 | 0 | 45 | 3 | 13 | 0 |

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\Pr(c_r\|d)$ | $\Pr(c_p\|d)$ | $\Pr(c_P\|d)$ | $\Pr(c_L\|d)$ | $\Pr(c_W\|d)$ | $R(d,c_P)$ | $R(d,c_L)$ | $R(d,c_W)$ | $h(d)$ | $R(d,h(d))$ | $E_{y_p}[R(d)]$ | $E_{y_p y}[\Delta^{or}(d)]$ | Annotate? |
| | | | Eq. 1 | Eq. 1 | Eq. 1 | Eq. 3 | Eq. 3 | Eq. 3 | Eq. 2 | Eq. 3 | Eq. 19 | Eq. 18 | |
| $d_1$ | 1.00 | 0.70 | 0.30 | 0.70 | 0.00 | 3.45 | 3.50 | 2.40 | $c_L$ | 10.00 | 0.00 | -2.40 | Yes |
| | 1.00 | 1.00 | 0.00 | 1.00 | 0.00 | 5.00 | 0.00 | 13.00 | $c_L$ | 0.00 | | | |
| | 1.00 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 8.00 | 3.00 | $c_P$ | 0.00 | | | |
| $d_2$ | 0.85 | 0.25 | 0.63 | 0.21 | 0.15 | 1.513 | 11.85 | 4.67 | $c_P$ | 1.51 | 1.51 | 0.00 | No |
| | 0.85 | 1.00 | 0.00 | 0.85 | 0.15 | 4.70 | 6.75 | 11.05 | $c_P$ | 4.70 | | | |
| | 0.85 | 0.00 | 0.85 | 0.00 | 0.15 | 0.45 | 13.55 | 2.55 | $c_P$ | 0.45 | | | |
| $d_3$ | 1.00 | 0.45 | 0.55 | 0.45 | 0.00 | 2.25 | 4.40 | 7.50 | $c_P$ | 2.25 | 0.00 | -2.25 | Yes |
| | 1.00 | 1.00 | 0.00 | 1.00 | 0.00 | 5.00 | 0.00 | 13.00 | $c_L$ | 0.00 | | | |
| | 1.00 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 8.00 | 3.00 | $c_P$ | 0.00 | | | |
| $d_4$ | 0.00 | 0.70 | 0.00 | 0.00 | 1.00 | 3.00 | 45.00 | 0.00 | $c_W$ | 0.00 | 0.00 | 0.00 | No |
| | 0.00 | 1.00 | 0.00 | 0.00 | 1.00 | 3.00 | 45.00 | 0.00 | $c_W$ | 0.00 | | | |
| | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 3.00 | 45.00 | 0.00 | $c_W$ | 0.00 | | | |
| $d_5$ | 1.00 | 0.70 | 0.30 | 0.70 | 0.00 | 3.50 | 2.40 | 10.00 | $c_L$ | 2.40 | 0.00 | -2.40 | Yes |
| | 1.00 | 1.00 | 0.00 | 1.00 | 0.00 | 5.00 | 0.00 | 13.00 | $c_L$ | 0.00 | | | |
| | 1.00 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 8.00 | 3.00 | $c_P$ | 0.00 | | | |

*Example 3.2. Table 4 shows the same example documents from Table 3 as they are processed in Phase 3. Documents $d_1$ and $d_3$ were annotated for responsiveness in Phase 2; for them, $P(c_r|d)$ has thus been updated (we here assume that they have both been deemed responsive by the human reviewers, which means that their $P_2(c_r|d)$ value is 1), while for the other three documents it is the case that $P_1(c_r|d) = P_2(c_r|d)$.*

*Overall, out of the 5 example documents, two ($d_1$ and $d_3$) are annotated for both responsiveness and privilege, two ($d_2$ and $d_4$) are annotated for neither, and one ($d_5$) is annotated for privilege only.*

*One interesting case is represented by $d_4$. Since its $\Pr(c_r|d)$ value is 0, the system is already certain that its class is $c_W$, so the expected reduction in cost that would derive from annotating it by privilege (Column 13) is 0. So, the model sanctions that annotating $d_4$ by privilege would be completely useless. In other words, what is a standard practice of e-discovery (i.e., documents that are deemed nonresponsive are withheld without checking the existence of privilege) here "emerges" as a consequence of MINECORE.*

The overall algorithm that implements MINECORE is summarized as Algorithm 1.

### 3.4 A few observations

A first thing to observe is that, in MINECORE, a document can end up being manually annotated only for responsiveness, only for privilege, for both responsiveness and privilege, or for neither responsiveness nor privilege. Note that annotating a document $d$ for responsiveness has the effect

**ALGORITHM 1:** MINECORE, a hybrid model for jointly minimizing the expected costs of review for responsiveness and privilege.

**Input** : A training set $Tr_r$ of documents labeled for responsiveness;
A training set $Tr_p$ of documents labeled for privilege;
Documents $\mathcal{D}$ to be analysed for possible production to the requesting party;
Cost structure $\Lambda = (\Lambda_m, \Lambda_a)$.

**Output** : A partition of $\mathcal{D}$ into the following three sets:
– Set $\mathcal{D}_P$ of documents to be produced to the receiving party;
– Set $\mathcal{D}_L$ of documents to be put on the privilege log;
– Set $\mathcal{D}_W$ of documents to be withheld;
Annotation cost $K^a(\mathcal{D})$ incurred in the process.

`/* Phase 1 */`
Train classifiers $h_r$ and $h_p$ from $Tr_r$ and $Tr_p$, respectively;
**for** $d \in \mathcal{D}$ **do**
  Compute $\Pr_1(c_r|d)$ by means of $h_r$ and $\Pr_1(c_p|d)$ by means of $h_p$;
  Compute $h_1(d)$ via Equation 10;
**end**

`/* Phase 2 */`
**for** $d \in \mathcal{D}$ **do**
  $\Pr_2(c_r|d) \leftarrow \Pr_1(c_r|d)$; $\Pr_2(c_p|d) \leftarrow \Pr_1(c_p|d)$; Compute $E_{y_r y}[\Delta^{or}(d)]$ using Equation 13;
**end**
Generate a ranking $R^r_{\mathcal{D}}$ of $\mathcal{D}$ in increasing order of $E_{y_r y}[\Delta^{or}(d)]$;
`/*` $R^r_{\mathcal{D}}(k)$ `denotes the document at the` $k$`-th rank position in` $R^r_{\mathcal{D}}$ `*/`
$k \leftarrow 1$; $\tau_r \leftarrow 0$;
**while** $E_{y_r y}[\Delta^{or}(R^r_{\mathcal{D}}(k))] < 0$ **do**
  Have the reviewer annotate document $R^r_{\mathcal{D}}(k)$ for responsiveness;
  **if** $R^r_{\mathcal{D}}(k)$ *has been judged responsive by the reviewer* **then**
    $\Pr_2(c_r|R^r_{\mathcal{D}}(k)) \leftarrow 1$
  **else**
    $\Pr_2(c_r|R^r_{\mathcal{D}}(k)) \leftarrow 0$
  **end**
  $\tau_r \leftarrow \tau_r + 1$; $k \leftarrow k + 1$;
**end**
**for** $d \in \mathcal{D}$ **do**
  Compute $h_2(d)$ using Equation 10;
**end**

`/* Phase 3 */`
**for** $d \in \mathcal{D}$ **do**
  $\Pr_3(c_r|d) \leftarrow \Pr_2(c_r|d)$; $\Pr_3(c_p|d) \leftarrow \Pr_2(c_p|d)$; Compute $E_{y_p y}[\Delta^{op}(d)]$ using Equation 18;
**end**
Generate a ranking $R^p_{\mathcal{D}}$ of $\mathcal{D}$ in increasing order of $E_{y_p y}[\Delta^{op}(d)]$;
`/*` $R^p_{\mathcal{D}}(k)$ `denotes the document at the` $k$`-th rank position in` $R^p_{\mathcal{D}}$ `*/`
$k \leftarrow 1$; $\tau_p \leftarrow 0$;
**while** $E_{y_p y}[\Delta^{op}(R^p_{\mathcal{D}}(k))] < 0$ **do**
  Have the reviewer annotate document $R^p_{\mathcal{D}}(k)$ for privilege;
  **if** $R^p_{\mathcal{D}}(k)$ *has been judged privileged by the reviewer* **then**
    $\Pr_3(c_p|R^p_{\mathcal{D}}(k)) \leftarrow 1$
  **else**
    $\Pr_3(c_p|R^p_{\mathcal{D}}(k)) \leftarrow 0$
  **end**
  $\tau_p \leftarrow \tau_p + 1$; $k \leftarrow k + 1$;
**end**
**for** $d \in \mathcal{D}$ **do**
  Compute $h_3(d)$ using Equation 10;
**end**
$\mathcal{D}_P \leftarrow \{d|h_3(d) = c_P\}$; $\mathcal{D}_L \leftarrow \{d|h_3(d) = c_L\}$; $\mathcal{D}_W \leftarrow \{d|h_3(d) = c_W\}$;
Compute $K^a(\mathcal{D})$ using Equation 6.

of reducing the number of possible misclassification types for $d$. E.g., if $d$ is annotated as responsive, this is tantamount to turning (for $d$) the $3 \times 3$ matrixes of Table 2 into $2 \times 2$ matrixes, as a result of removing the $c_W$ row and the $c_W$ column; if it is instead annotated as nonresponsive, the $3 \times 3$ matrixes become $1 \times 1$ matrixes, where only the $c_W$ row and the $c_W$ column have survived. Likewise, if $d$ has been annotated as responsive, also annotating it by privilege has the effect of turning the $2 \times 2$ matrixes into $1 \times 1$ matrixes.

A second thing to observe is that Phases 2 and 3 are structurally identical, since Phase 2 does for responsiveness exactly what Phase 3 does for privilege. In particular, note that Phase 3 processes *all* documents $d \in \mathcal{D}$, and not just those that Phase 2 has decreed responsive or probably responsive (see the case of document $d_4$ in Example 4). One might thus wonder if we could switch the order of Phase 2 and Phase 3 without negatively impacting (or perhaps even positively impacting) $K^o(\mathcal{D})$. The answer is no, and the reason lies in the fact that, in typical e-discovery scenarios, $\lambda_p^a$ is higher or much higher than $\lambda_r^a$ (we indeed imposed the constraint that $\lambda_r^a < \lambda_p^a$ in Section 2.3). This has the consequence that it makes sense to employ the expensive (as characterised by $\lambda_p^a$) senior reviewers for annotating documents that the cheap (as characterised by $\lambda_r^a$) junior reviewers have already "pre-filtered".

A third important observation is about ranking. During Phase 2 MINECORE clearly separates the set (let us call it $\mathcal{D}_2^{man}$) of the $\tau_r$ documents that should be annotated from the set (let us call it $\mathcal{D}_2^{aut}$) of the $(|\mathcal{D}| - \tau_r)$ documents that should not be annotated (the same happens at the end of Phase 3). If the human reviewer annotates all and only the former, one might wonder why is ranking useful at all. While ranking is indeed unnecessary in theory, it is useful in practice, for two reasons:

- The choice of which documents to put in $\mathcal{D}_2^{man}$ and which to put in $\mathcal{D}_2^{aut}$ is far from perfect, since it relies on automatically generated posterior probabilities. As a result, the human reviewer might find out, at the very moment she is invited to stop annotating, that she was still finding many mislabeled documents, and she might thus want to annotate some more documents in order to be on the safe side;
- If, for some reason, the reviewer stops annotating before the stopping condition is reached, the fact that she has annotated by following the ranked list guarantees that the cost-effectiveness of her work has been maximized.

As a result, we indeed assume that rankings are generated (and followed by the human reviewers) in both Phase 2 and Phase 3.

## 4 EXPERIMENTS

In this section we describe a number of experiments that we have conducted to test the cost-effectiveness of MINECORE.[6]

### 4.1 Test Collection

One problem that hinders the evaluation of MINECORE is that in the world of e-discovery, at present, there is no publicly available collection of documents that are annotated by both responsiveness and privilege.[7] A way out of this could be to generate such an annotated collection ourselves: however, this would be a major feat in terms of annotation cost, since it takes real lawyers to do this annotation, and real lawyers (especially senior ones, whom we would need in order to annotate for

[6]The code that implements MINECORE is available at https://github.com/minecore2018/tois_code.git

[7]The TREC 2010 Legal Track included one (nontopical) "topic" annotated by privilege and several topics annotated for responsiveness, but the intersection between the former and each of the latter is minimal because the samples were drawn independently for each.

privilege) can be extremely expensive. We bypass this problem by running "simulated" experiments, on a collection unrelated to e-discovery in which documents can belong to more than one class, and by repeatedly picking two classes to play the role of $c_r$ and $c_p$, respectively.

As a test collection we have chosen RCV1-v2, a standard, publicly available benchmark for text classification first presented in [25] and consisting of 804,414 news stories produced by Reuters from 20 Aug 1996 to 19 Aug 1997.[8] RCV1-v2 ranks as one of the largest corpora currently used in text classification research; as pointed out in [12], it suffers from "drift", i.e., from substantial variability between the training set and the test set, which makes it a challenging test collection. RCV1-v2 is multi-label, i.e., a document may belong to several classes at the same time, which makes it suitable for our purposes. In [25] the collection is partitioned into a training set of 23,149 documents and a test set of 781,265 documents, the latter being split into four chunks of 199,328, 199,339, 199,576, 183,022 documents, respectively. In the experiments reported in this paper we have used the 23,149 training documents as the training set $Tr$, and the first chunk of 199,328 test documents as the test set $Te$.

In the "Topic" hierarchy of RCV1-v2 there are 103 classes, of which 101 have at least one positive training example. Since we experiment with pairs of classes (representing $c_r$ and $c_p$), we could in principle experiment with $101^2 = 10,201$ different pairs. Aside from representing a substantive computational load, this would also mean experimenting with classes whose prevalence is, given typical e-discovery scenarios, not realistic. We have therefore limited our experiments to pairs $(c_r, c_p)$ such that the prevalence of $c_r$ in the entire RCV1-v2 collection is in [0.03,0.07] and the prevalence of $c_p$ in the responsive documents is in [0.01,0.20]. This broad range is actually seen in e-discovery practice, with some classification tasks run in "needle in a haystack" conditions, and others run on collections that have been prescreened when they were acquired to have as high a responsiveness prevalence as can be achieved [27]. For each of the 24 responsiveness classes that meet the prevalence criterion we have randomly selected 5 privilege classes that meet the prevalence criterion. This gives rise to 120 class pairs, which is the set we use for the experiments described in this paper. We note that our process for selecting category pairs was entirely automatic, in contrast to the process used in the TREC 2002 Filtering Track, where an effort was made to select pairs that were related in ways that were expected to reflect some plausible information need at their intersection [29].

### 4.2 The learning algorithm

For all the experiments reported in this paper we have used Support Vector Machines (SVMs) as the learning device, since they have consistently delivered strong performance in text classification. We have used the well-known SVM-LIGHT implementation due to [18], for which we have used the default parameter values; in particular, we have used a linear kernel, due to its well-known good performance in text classification tasks [19]. Concerning the vector representations fed to the SVM learner, in order to enhance reproducibility we have used the ones made available as Online Appendix 13 of [25], which consist of vectors of unigrams obtained via standard tokenization, stopwording, stemming, and tf-idf (in the "ltc" variant) weighting.[9] We refer to [25] for more details on the preprocessing techniques that were used to generate them.

Classifiers generated via SVMs return confidence scores that are not posterior probabilities (see also Footnote 3); these scores must thus be converted into posterior probabilities, since MINECORE

[8] `http://trec.nist.gov/data/reuters/reuters.html`

[9] In actual practice in e-discovery additional features would be used, particularly for privilege classification, since the roles of the parties who sent and received document are also important. In our prior work on privilege classification [36] we have found that privilege classification accuracy is in line with what we would expect from topic classification when the feature set is well designed.

Table 5. Cost structures used in our experiments, as elicited from different experts. Each individual cost is expressed in US$.

| | $\lambda^a_r$ | $\lambda^a_p$ | $\lambda^m_{PL}$ | $\lambda^m_{PW}$ | $\lambda^m_{LP}$ | $\lambda^m_{LW}$ | $\lambda^m_{WP}$ | $\lambda^m_{WL}$ |
|---|---|---|---|---|---|---|---|---|
| CostStructure1 | 1.00 | 5.00 | 600.00 | 5.00 | 150.00 | 3.00 | 15.00 | 15.00 |
| CostStructure2 | 1.00 | 5.00 | 100.00 | 0.03 | 10.00 | 2.00 | 8.00 | 8.00 |
| CostStructure3 | 1.00 | 5.00 | 1000.00 | 0.10 | 1.00 | 1.00 | 1.00 | 1.00 |

essentially depends on the availability of such probabilities. Given that the returned scores are a monotonically increasing function of the classifier's confidence in the fact that the document belongs to the class, this conversion may be obtained by applying to the scores a logistic function, since such a function has a sigmoidal shape that monotonically maps $(\infty, +\infty)$ into $[0, 1]$. We obtain "well calibrated" posterior probabilities via so-called "Platt scaling", i.e., by using a "generalized" (i.e., parametric) logistic function and optimizing its parameters via $k$-fold cross-validation.[10]

### 4.3 Cost structures

In order to use realistic misclassification costs and annotation costs, we have chosen to elicit our cost structures from e-discovery experts. We have been able to obtain the help of three senior members of the e-discovery community (two lawyers and an technical expert in technology-assisted review), each of whom have extensive experience with actual e-discovery cases in their practice. We asked each of them to think of an actual case they may be familiar with, and to articulate the cost structure that characterises that case. Through this process we obtained 3 cost structures, which are detailed in Table 5. Note that the values indicated by the 3 experts are in some cases markedly different (e.g., there is a factor of 150 between the values of $\lambda^m_{LP}$ indicated by two of the experts); this need not be taken as evidence of disagreement among the experts for decisions on the same task, since different experts were free to choose different legal cases to have in mind when arriving at these estimates. Rather than trying to reconcile these cost structures in any way, we have thus run 3 experiments, one for each of the cost structures, on the assumption that MINECORE should be able to cater to different application needs.

### 4.4 Baselines

We are here proposing some baseline methods against which to compare MINECORE. Throughout this paper we use the same vector representations for the documents, the same supervised learning algorithm, and the same classifier outputs, for all the methods being compared. Each method (be it MINECORE or a baseline method) assigns, for each test document $d$, a class in $C = \{c_P, c_L, c_W\}$.

Our baseline methods are (aside from the fully automated and fully manual solutions) mixed-initiative, "human-in-the-loop" systems, i.e., their classification decisions are obtained via some combination of manual annotation work and automatic classification. Using the cost structures exemplified in Table 5 we can evaluate each system using the evaluation measure described in Equation 7; that is, for each system we compute the misclassification cost $K^m(\mathcal{D})$, the annotation cost $K^a(\mathcal{D})$, and the overall cost $K^o(\mathcal{D}) = K^m(\mathcal{D}) + K^a(\mathcal{D})$ they incur. The best system is the one with the lowest $K^o(\mathcal{D})$ cost.

[10] Although this calibration method is generally credited to Platt [28], the idea of using a logistic regression model for mapping the scores of a classifier into well calibrated probabilities was originally introduced in [24].

**Fully Automated (FA).** The first baseline we consider is the fully automated solution, as described in Section 2.1; for this method the annotation cost $K^a(\mathcal{D})$ is zero, so its cost $K^o(\mathcal{D})$ defaults to the misclassification cost $K^m(\mathcal{D})$, which is computed according to Equation 5.

**Fully Manual (FM).** The second baseline we consider is the fully manual solution, as described in Section 2.2; for this method the misclassification cost $K^m(\mathcal{D})$ is zero (since we assume perfect reviewers), so its cost $K^o(\mathcal{D})$ defaults to the annotation cost for the full collection $K^a(\mathcal{D})$, which is computed according to Equation 6.

**Uncertainty Ranking (UR).** In UR we first annotate for responsiveness the $\tau_r$ documents whose $\Pr(c_r|d)$ is closest to 0.5 (i.e., the ones whose responsiveness is most uncertain). A document is then deemed responsive if the reviewer has annotated it as such, or (for the documents which have not been manually annotated for responsiveness) if $\Pr(c_r|d) > 0.5$. We then annotate for privilege, among the documents that have been deemed responsive, the $\tau_p$ documents whose $\Pr(c_p|d)$ is closest to 0.5. A document is then deemed privileged if the reviewer has annotated it as such, or (for the documents which have not been manually annotated for privilege) if $\Pr(c_p|d) > 0.5$. This baseline is similar to MINECORE in that the class assigned to a test document may result from the reviewers' manual annotation work, or from the automated classifiers, or from a combination of them. However, neither annotation costs nor misclassification costs play a role in UR.

**Relevance Ranking (RR).** In RR we first annotate for responsiveness the $\tau_r$ documents with the *highest* $\Pr(c_r|d)$, and we then annotate for privilege, among the documents that the reviewers have deemed responsive in the previous phase, the $\tau_p$ documents with the *lowest* $\Pr(c_p|d)$. Unlike MINECORE and UR, RR assumes that only the documents that have been certified responsive and nonprivileged by the reviewers are going to be produced (documents certified responsive and privileged by the reviewers are entered on the privilege log, while all other documents are withheld); as a result, the two rankings (by $\Pr(c_r|d)$ and $\Pr(c_p|d)$) attempt to top-rank the documents that have the highest chances of meeting the requirements (responsiveness *and* nonprivilege) for disclosure.

**Active Learning via Uncertainty Sampling (ALvUS).** In the design of MINECORE our focus has been on cases in which, given the learning algorithm we have chosen, we have already built the best classifier we can, and in such cases we would not expect further gains from active learning. In our experiments, however, we have simply trained on a fixed set of 23,149 documents, and it is possible that active learning might indeed give further gains. This motivates our choice to include ALvUS and ALvRS (see below) as additional baselines. In ALvUS, an interactive process asks the reviewer to annotate for responsiveness the $b$ documents in $\mathcal{D}$ for which $\Pr(c_r|d)$ is closest to 0.5 (parameter $b$ is known as the *batch size*); at this point, this set $\mathcal{D}_r$ of $b$ documents is added to the training set, the posterior probabilities $\Pr(c_r|d)$ of the documents $d$ annotated as responsive (resp., nonresponsive) are set to 1 (resp., 0), $h_r$ is retrained, and $\mathcal{D}/\mathcal{D}_r$ is classified for responsiveness again; this process is repeated (using the newly computed $\Pr(c_r|d)$ values) until exactly $\tau_r$ documents have been annotated.[11] After this, an identical process is used for privilege, substituting $h_p$ and $\tau_p$ for $h_r$ and $\tau_r$ in the above. At the end, a document $d \in \mathcal{D}$ is assigned to $c_P$ iff $\Pr(c_r|d) > 0.5$ and $\Pr(c_p|d) \leq 0.5$; to $c_L$ iff $\Pr(c_r|d) > 0.5$ and $\Pr(c_p|d) > 0.5$; and to $c_W$ otherwise. ALvUS is similar to MINECORE and UR, in that the class assigned to a test document may result from the reviewers' manual annotation work, or from the automated classifiers, or from a combination of them. In the experiments reported in this paper we use $b = 1000$, which was found to work well by [6], since smaller values would be less computationally tractable. Note that the comparison between MINECORE and ALvUS is only partially fair, since ALvUS is much more expensive computationally,

[11] To be more precise, in the last iteration fewer than $b$ documents may be annotated, so as to make the total number of documents annotated equal to $\tau_r$. For example, if $\tau_r = 3267$ and $b = 1000$, 1000 documents will be annotated in each of the first three rounds, while in the final round only 267 documents will be annotated.

given that it requires $\lceil \tau_r/k \rceil + \lceil \tau_p/k \rceil$ retraining operations (unlike MINECORE, which requires none).[12]

**Active Learning via Relevance Sampling (ALvRS).** A variant of the previous baseline is obtained if the active learning process asks the reviewer to annotate for responsiveness the $b$ documents in $\mathcal{D}$ for which $\Pr(c_r|d)$ *is highest* (and the ones for which $\Pr(c_p|d)$ is *lowest* when the reviewer annotates for privilege). The rest of the process is as in ALvUS; in particular, here too we use $b = 1000$. At the end, a document $d \in \mathcal{D}$ is assigned to $c_P$ iff it has been manually annotated as responsive and nonprivileged; it is assigned to $c_L$ iff it has been manually annotated as responsive and privileged; it is assigned to $c_W$ otherwise. Unlike ALvUS, ALvRS thus assumes that, unless a document has been under the scrutiny of *both* the junior reviewer (for responsiveness) and the senior reviewer (for privilege), it is withheld.

Among e-discovery researchers and practitioners, ALvRS is known as "continuous active learning" (CAL) [6, 7, 9]; ALvRS was originally introduced in [24], where it was indeed called "Relevance Sampling".[13] The latter paper is also the work in which ALvUS was introduced first, under the name of "Uncertainty Sampling". Note also that both ALvUS and ALvRS as used here bear similarities with relevance feedback (which is indeed a form of active learning), since the classifiers they retrain only need to generalize to a finite set of examples (i.e., $\mathcal{D}$), and these examples are all available at training time. In other words, ALvUS and ALvRS here operate (like relevance feedback) in a *transductive* (a.k.a. "finite population") setting, unlike other instantiations of active learning (included the ones originally discussed in [24]) which are meant to operate in the full-blown inductive setting.

Note that for every baseline system other than FA and FM we compute the cost $K^o(\mathcal{D})$ that the baseline incurs when manually annotating exactly $\tau_r$ documents for responsiveness and, if possible,[14] $\tau_p$ documents for privilege, where $\tau_r$ and $\tau_p$ are the values used in the MINECORE system. This policy may be biased in favor of MINECORE, since $\tau_r$ and $\tau_p$ are optimal settings for MINECORE whereas other systems might have yielded lower overall costs with either more or less manual reviewing. However, none of the baseline systems we test have an a priori way of analytically setting the optimal number of documents to manually review. This means that our comparisons are, if not to post-hoc optimal systems, at least to reasonable systems.

### 4.5 Experimental protocol

The experimentation protocol we adopt is the following. As groundwork, we train our binary classifiers via the chosen binary learner using the 23,149 training documents, and apply them to the 199,328 test documents (the test set $Te$ thus plays the role of our universe $\mathcal{D}$). For each document $d \in Te$, the classifier for class $c$ generates a confidence score, from which we obtain a posterior probability $\Pr(c|d)$ via probability calibration.

At this point, we run each of the seven methods (MINECORE plus the six baseline methods) for each of the cost structures (see Table 5) we have elicited from the experts. In particular, for the risk minimization method, we first simulate the manual annotation process for responsiveness: for all $d \in \mathcal{D}$ such that $E_{y_r y}[\Delta^{or}(d)] < 0$ we set $\Pr_2(c_r|d)$ to 1 if $d$ is responsive and to 0 if $d$ is

[12] In order to increase accuracy even further, in an operational situation one could integrate MINECORE and active learning, by repeatedly (i) retraining $h_r$ (resp., $h_p$) after $b$ documents have been annotated, (ii) re-generating the $\Pr_2(c_r|d)$'s (resp., the $\Pr_3(c_r|d)$'s) via the newly trained classifier, and (iii) reranking the remaining documents.

[13] CAL, as described in [6, 7, 9], is actually a simpler variant of ALvRS since it deals with one classification task only (i.e., responsiveness), instead of the two cascaded tasks (i.e., responsiveness and privilege) that ALvRS deals with.

[14] In some cases a baseline system might deem responsive *fewer* than $\tau^p$ documents, which means that fewer than $\tau^p$ documents (i.e., all the ones deemed responsive) would be annotated for privilege; in this case the comparison between this baseline system and all other systems (including MINECORE) is still fair, though, since this system will incur a smaller annotation cost (for privilege) than MINECORE.

nonresponsive. We then do the same for privilege: for all $d \in \mathcal{D}$ such that $E_{y_p y}[\Delta^{op}(d)] < 0$ we set $\Pr_3(c_p|d)$ to 1 if $d$ is privileged and to 0 if $d$ is nonprivileged. We then compute the total cost of the process via Equation 7, which works out as

$$\begin{aligned} K^o(\mathcal{D}) &= K^a(\mathcal{D}) + K^m(\mathcal{D}) \\ &= \tau_r \lambda_r^a + \tau_p \lambda_p^a + \sum_{i,j \in \{P,L,W\}} \lambda_{ij}^m \cdot |\{d \in \mathcal{D} | h_3(d) = c_i \text{ and } y(d) = c_j\}| \end{aligned} \tag{21}$$

We simulate the manual annotation process in a similar way also for all the baseline methods.

### 4.6 Results

In this section we present the results of testing MINECORE against the 6 baseline methods presented in Section 4.4, on the 120 class pairs described at the end of Section 4.1; we have run each such experiment for each of the 3 cost structures discussed in Section 4.3.

In Table 6 we exemplify, on a sample cost structure (CostStructure1), what the results look like. The table reports, for each class pair, the class prevalences of $c_r$ and $c_p$, the values of $\tau_r$ and $\tau_p$ that MINECORE returns, the $K^o(\mathcal{D})$ value (expressed in thousands of US$) resulting from MINECORE, and, for each of the 6 baseline methods, the increase in $K^o(\mathcal{D})$ value with respect to MINECORE (a positive increase means that the baseline generates higher costs than MINECORE).

Table 6 shows that, for this cost structure, MINECORE is the least expensive of the seven methods for the 30 class pairs displayed; this actually happens for all 120 class pairs. An overall view of the relative merits of the 7 methods can be obtained by looking at the bottom row of the table, which reports median values computed across the 120 class pairs (throughout this paper we look at medians, rather than at averages, in order to reduce the impact of outliers). In terms of the median values, the 2nd best method is (surprisingly enough) the FA method, which is 29% more expensive than MINECORE. Other methods are even more expensive, up to 235% more than MINECORE; among these other methods one can note a slight advantage of the uncertainty-based methods (UR and ALvUS) over the relevance-based ones (RR and ALvRS), while there seems to be no substantial difference between the methods which are based on active learning (ALvUS and ALvRS) and the ones which are not (UR and RR).

The values of $\tau_r$ range in the [809,18998] interval, corresponding to [0.41%,9.53%] of the total set of 199,328 documents; those of $\tau_p$ range instead in the [389,7942] interval, corresponding to [0.20%,3.98%] of the total set. This shows two important facts. First, MINECORE sanctions that only a small minority of the documents (max 9.53% of the total lot for responsiveness, max 3.98% for privilege) should be manually reviewed; this is in line with what we would expect, given the cost structure. Second, MINECORE requires many fewer documents to be manually annotated for privilege than for responsiveness; this is a consequence (a) of the fact that many documents are ruled out from further consideration on responsiveness grounds alone, and are not further checked for privilege; and (b) of the fact that manually reviewing for privilege is more expensive, and thus more strongly discouraged by MINECORE, than manually reviewing for responsiveness.

Figure 4 illustrates the same results, with the class pairs sorted in order of decreasing overall cost for MINECORE. Three patterns are evident in that figure. First, the cost of the FM baseline is quite high, varying in a narrow range in a manner that strictly depends on the prevalence of the responsiveness class. Second, none of the baselines other than FM, while all systematically better than FM, are systematically better or systematically worse than all the other ones, which is shown by the fact that the relative plots keep intersecting each other. Third, MINECORE systematically outperforms all others, often by a substantial margin.

Table 6. Results obtained by using a sample cost structure (here: CostStructure1). Columns 2 and 3 indicate the identifiers of the RCV1-v2 classes that play the role of $c_r$ and $c_p$, respectively. MINECORE is here shortened as "RM" (for "Risk Minimization"). RM $K^o(\mathcal{D})$ denotes the cost incurred by MINECORE for a certain class pair; for readability we indicate costs in thousands of US$, rounding them to the closest unit (e.g., $23,456 would be indicated as 23). $\Delta$ denotes the percentage increase in cost that derives by adopting the method indicated instead of MINECORE (e.g., +30% means that the cost of the method is 30% higher than that of MINECORE). Due to pagination issues, only the first 30 class pairs are shown; a full table with data for all 120 class pairs × 3 cost structures is online at https://github.com/minecore2018/tois_code.git. The last row represents median values across the 120 class pairs.

| | $c_r$ | $c_p$ | $\Pr(c_r)$ | $\Pr(c_p\|c_r)$ | $\tau_p$ | $\tau_r$ | FA $\Delta$ | FM $\Delta$ | UR $\Delta$ | RR $\Delta$ | ALvUS $\Delta$ | ALvRS $\Delta$ | RM $K^o(\mathcal{D})$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | M12 | M14 | 3% | 1% | 3257 | 1100 | +13% | +865% | +22% | +45% | +28% | +41% | 23 |
| 2 | M12 | CCAT | 3% | 5% | 1738 | 1997 | +36% | +533% | +63% | +68% | +82% | +65% | 36 |
| 3 | M12 | M132 | 3% | 7% | 2889 | 1201 | +38% | +424% | +57% | +57% | +51% | +54% | 43 |
| 4 | M12 | E21 | 3% | 11% | 2048 | 2063 | +44% | +353% | +71% | +68% | +73% | +66% | 50 |
| 5 | M12 | M131 | 3% | 18% | 2726 | 1400 | +30% | +64% | +39% | +36% | +41% | +29% | 139 |
| 6 | M132 | GPOL | 3% | 1% | 2254 | 1227 | +25% | +859% | +39% | +59% | +44% | +54% | 24 |
| 7 | M132 | CCAT | 3% | 2% | 1794 | 2300 | +26% | +596% | +58% | +66% | +51% | +66% | 33 |
| 8 | M132 | M12 | 3% | 6% | 2360 | 1828 | +12% | +588% | +30% | +45% | +25% | +42% | 33 |
| 9 | M132 | M131 | 3% | 7% | 2506 | 1685 | +29% | +332% | +49% | +48% | +47% | +38% | 53 |
| 10 | M132 | GCAT | 3% | 15% | 2258 | 1152 | +25% | +592% | +40% | +48% | +47% | +46% | 33 |
| 11 | M131 | CCAT | 3% | 1% | 1141 | 2797 | +34% | +490% | +71% | +72% | +65% | +70% | 39 |
| 12 | M131 | M132 | 3% | 6% | 1709 | 1528 | +27% | +365% | +56% | +44% | +30% | +40% | 50 |
| 13 | M131 | E12 | 3% | 7% | 1309 | 2066 | +36% | +280% | +55% | +52% | +69% | +53% | 61 |
| 14 | M131 | ECAT | 3% | 9% | 822 | 3334 | +61% | +291% | +88% | +90% | +88% | +83% | 59 |
| 15 | M131 | M12 | 3% | 15% | 1465 | 1823 | +34% | +313% | +44% | +47% | +63% | +47% | 56 |
| 16 | E12 | M11 | 3% | 1% | 8371 | 437 | +32% | +458% | +7% | +14% | +10% | +12% | 42 |
| 17 | E12 | GDIP | 3% | 3% | 7135 | 1334 | +22% | +290% | +22% | +25% | +21% | +29% | 60 |
| 18 | E12 | E212 | 3% | 4% | 7135 | 1336 | +30% | +323% | +29% | +36% | +29% | +35% | 55 |
| 19 | E12 | M131 | 3% | 7% | 7639 | 1467 | +35% | +261% | +42% | +49% | +58% | +45% | 64 |
| 20 | E12 | E21 | 3% | 13% | 5589 | 1769 | +33% | +210% | +47% | +49% | +48% | +52% | 75 |
| 21 | C21 | C17 | 4% | 1% | 5862 | +18% | +18% | +254% | +14% | +19% | +9% | +13% | 66 |
| 22 | C21 | C15 | 4% | 3% | 4610 | 1651 | +11% | +211% | +16% | +19% | +13% | +15% | 75 |
| 23 | C21 | ECAT | 4% | 5% | 3084 | 2184 | +10% | +180% | +24% | +24% | +13% | +23% | 84 |
| 24 | C21 | C31 | 4% | 18% | 2037 | 2298 | +15% | +159% | +29% | +27% | +43% | +32% | 91 |
| 25 | C21 | M141 | 4% | 20% | 7052 | 389 | +15% | +162% | +10% | +12% | +9% | +10% | 90 |
| 26 | E212 | GPOL | 4% | 2% | 2527 | 3592 | +3% | +416% | +35% | +47% | +27% | +46% | 46 |
| 27 | E212 | E12 | 4% | 4% | 2357 | 1410 | +8% | +543% | +23% | +30% | +25% | +32% | 37 |
| 28 | E212 | M12 | 4% | 8% | 2312 | 1805 | +31% | +342% | +47% | +47% | +60% | +52% | 53 |
| 29 | E212 | MCAT | 4% | 9% | 2059 | 3171 | +23% | +297% | +51% | +53% | +59% | +50% | 59 |
| 30 | E212 | C17 | 4% | 19% | 1967 | 2574 | +11% | +327% | +34% | +35% | +48% | +37% | 55 |
| … | … | … | … | … | … | … | … | … | … | … | … | … | … |
| Median values across 120 class pairs | | | | | 4781 | 3610 | +29% | +221% | +46% | +53% | +46% | +52% | 77 |

Table 7 shows a comparison among the results obtained for the different cost structures on a representative class pair.[15] It is immediately obvious that the cost structure has a lot of influence (i) on how many documents get manually reviewed, both for responsiveness and for privilege, (ii) on the total costs incurred by the various methods, and (iii) on the difference in cost between these methods and MINECORE. The first first of those points has been previously noted in an more

[15]In this example responsiveness is simulated by RCV1-v2 class GPOL ("DomesticPolitics") while privilege is simulated by class CCAT ("Commercial/Industrial"); this class pair was chosen as representative since it is the one for which the median increase in overall cost (+47%) between MINECORE and a high-performing baseline (ALvUS) is obtained.

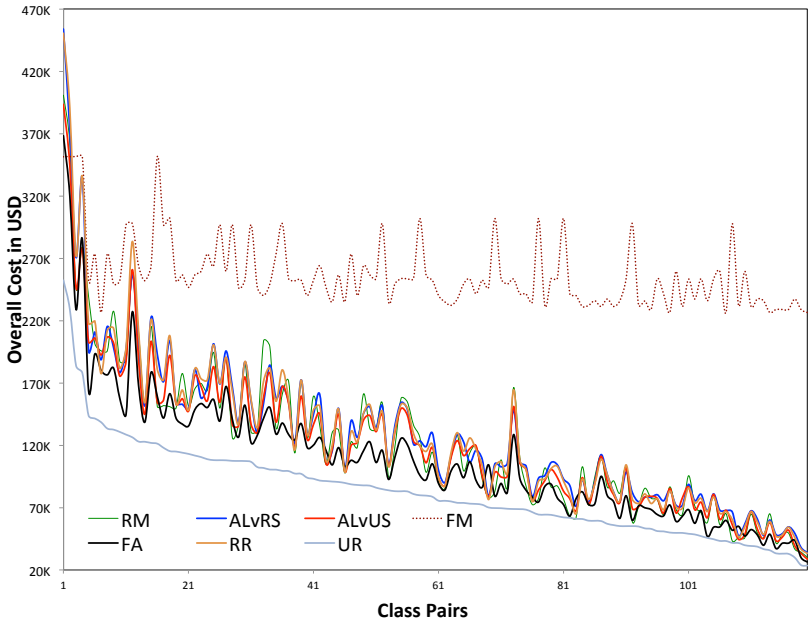


Fig. 4. Overall costs with CostStructure1 for the 7 methods across the 120 class pairs, with the $x$ axis sorted by decreasing cost for MINECORE (here shortened as "RM").

Table 7. Results obtained on a sample class pair (in this case: pair 97, with class GPOL as $c_r$ and class CCAT as $c_p$) using the different cost structures of Table 5. $K^o$ denotes the cost incurred by the method for the entire dataset $\mathcal{D}$ while $\Delta$ denotes the percentage increase in cost with respect to MINECORE (e.g., +30% means that the cost of the method is 30% higher than that of MINECORE). A greyed-out cell with a value in **boldface** indicates the best method. For readability we indicate costs in thousands of US$, rounding them to the closest unit; e.g., $272,456 would be indicated as 272. MINECORE is here shortened as "RM" (for "Risk Minimization").

| | $\tau_p$ | $\tau_r$ | FA | | FM | | UR | | RR | | ALvUS | | ALvRS | | RM |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | $K^o$ | $\Delta$ | $K^o$ | $\Delta$ | $K^o$ | $\Delta$ | $K^o$ | $\Delta$ | $K^o$ | $\Delta$ | $K^o$ | $\Delta$ | $K^o$ |
| CostStructure1 | 6169 | 6885 | 177 | +32% | 273 | +105% | 207 | +55% | 215 | +61% | 196 | +47% | 212 | +59% | **93** |
| CostStructure2 | 918 | 1189 | 57 | +3% | 273 | +397% | 63 | +14% | 64 | +16% | 57 | +3% | 63 | +14% | **55** |
| CostStructure3 | 0 | 0 | **15** | +0% | 273 | +1714% | **15** | +0% | **15** | +0% | **15** | +0% | **15** | +0% | **15** |

restricted setting, focused only on privilege [26]. In general CostStructure2 results in much smaller numbers of manually reviewed documents than CostStructure1; this is because (see Table 5) the misclassification costs are much smaller than in CostStructure1, which makes manual annotation less cost-effective. CostStructure3 is also an interesting limiting case, in that it results in $\tau_r = \tau_p = 0$; that is, MINECORE decrees that no document is worth manually annotating, and that the decisions of the automatic classifiers should be used, which means that in this case MINECORE coincides with FA. The reason for this behavior lies in the fact that the misclassification costs in $\Lambda_m$ are (relatively to the annotation costs in $\Lambda_a$) very low, too low to justify *any* amount of manual annotation. In general, if the costs in $\Lambda_m$ are low and the costs in $\Lambda_a$ are high, low values of $\tau_r$ and $\tau_p$ (sometimes as low as 0) will result, since manual annotation is discouraged. Conversely, if the costs in $\Lambda_m$

Table 8. Results obtained by using the different cost structures of Table 5; see Table 7 for the meaning of the various columns and notational conventions. The results in a given row are the median of the 120 results obtained with the tested 120 class pairs. A greyed-out cell with a value in **boldface** indicates the best method, while $^\dagger$ indicates a statistically significant ($p < 0.01$) increase in overall cost with respect to MINECORE (here shortened as "RM"), as determined by the Wilcoxon test discussed in this section.

| | FA | | FM | | UR | | RR | | ALvUS | | ALvRS | | RM |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $K^o(\mathcal{D})$ | $\Delta$ | $K^o(\mathcal{D})$ | $\Delta$ | $K^o(\mathcal{D})$ | $\Delta$ | $K^o(\mathcal{D})$ | $\Delta$ | $K^o(\mathcal{D})$ | $\Delta$ | $K^o(\mathcal{D})$ | $\Delta$ | $K^o(\mathcal{D})$ |
| CostStructure1 | 94 | +29%$^\dagger$ | 248 | +235%$^\dagger$ | 106 | +47%$^\dagger$ | 107 | +52%$^\dagger$ | 104 | +47%$^\dagger$ | 108 | +52%$^\dagger$ | **73** |
| CostStructure2 | 24 | +2%$^\dagger$ | 248 | +893%$^\dagger$ | 26 | +10%$^\dagger$ | 26 | +11%$^\dagger$ | 25 | +4%$^\dagger$ | 25 | +7%$^\dagger$ | **24** |
| CostStructure3 | **10** | +0% | 248 | +2416% | **10** | +0% | **10** | +0% | **10** | +0% | **10** | +0% | **10** |

are high and the costs in $\Lambda_a$ are low, high values of $\tau_r$ and $\tau_p$ (sometimes as high as $|\mathcal{D}|$) will result, and MINECORE will suggest manual annotation for all documents in $\mathcal{D}$. In general, the higher (resp., lower) the ratio between the costs in $\Lambda_m$ and those in $\Lambda_a$, the closer to FM (resp., FA) MINECORE is going to be performance-wise. MINECORE is especially advantageous with respect to both baselines when the cost structure justifies the notion that some (but not all) of the documents in $\mathcal{D}$ are worth annotating manually.

Figure 5 extends the comparison shown in Table 7 to the full set of class pairs. As can be seen, all of the baselines generally incur substantially higher costs than MINECORE with CostStructure1; this difference is instead far smaller for CostStructure2 (as noted above, there is no difference between MINECORE and the other baselines – except FM – for CostStructure3).

Finally, Table 8 shows the median (across the 120 class pairs) overall cost obtained by each method with each cost structure. For CostStructure2, MINECORE does better by this median measure than all of the baseline methods, albeit by smaller margins than are achieved for CostStructure1. For both of those two cost structures, the costs generated by each baseline method is statistically significantly higher according to a Wilcoxon signed rank test for paired samples over the 120 class pairs, at $p < 0.01$. Concerning CostStructure3, similarly to what happened for the pair showcased in Table 7, MINECORE evaluates both $\tau_r$ and $\tau_p$ to 0 for all class pairs, making MINECORE and all the other methods (aside from FM) coincide with FA.

Incidentally, one cannot help noticing how the FM fully manual baseline is, by a very wide margin and according to all three cost structures, the worst of all systems. This is a further confirmation of a fact first noted in [15], which reasserts that technology-assisted review is nowadays unavoidable in e-discovery.

### 4.7 Efficiency issues

We now discuss issues of computational cost. In Table 9 we report, for each cost structure, the average computation times (in seconds) required by each method, where averages are computed across all the class pairs. The figures do not include the times needed to index the documents, train the original classifiers, and apply the classifiers to all the test documents, which are the same for all methods and all cost structures (this is the reason why times for the FA method are 0). By "computation time" we thus mean

(1) for the UR and RR, baselines: the time needed to generate the two rankings;
(2) for MINECORE (RM): the time needed to calibrate the probabilities + the time needed to generate the two rankings.
(3) for the ALvUS and ALvRS baselines: the time needed to repeatedly (a) generate the two rankings and (b) retrain the classifiers;

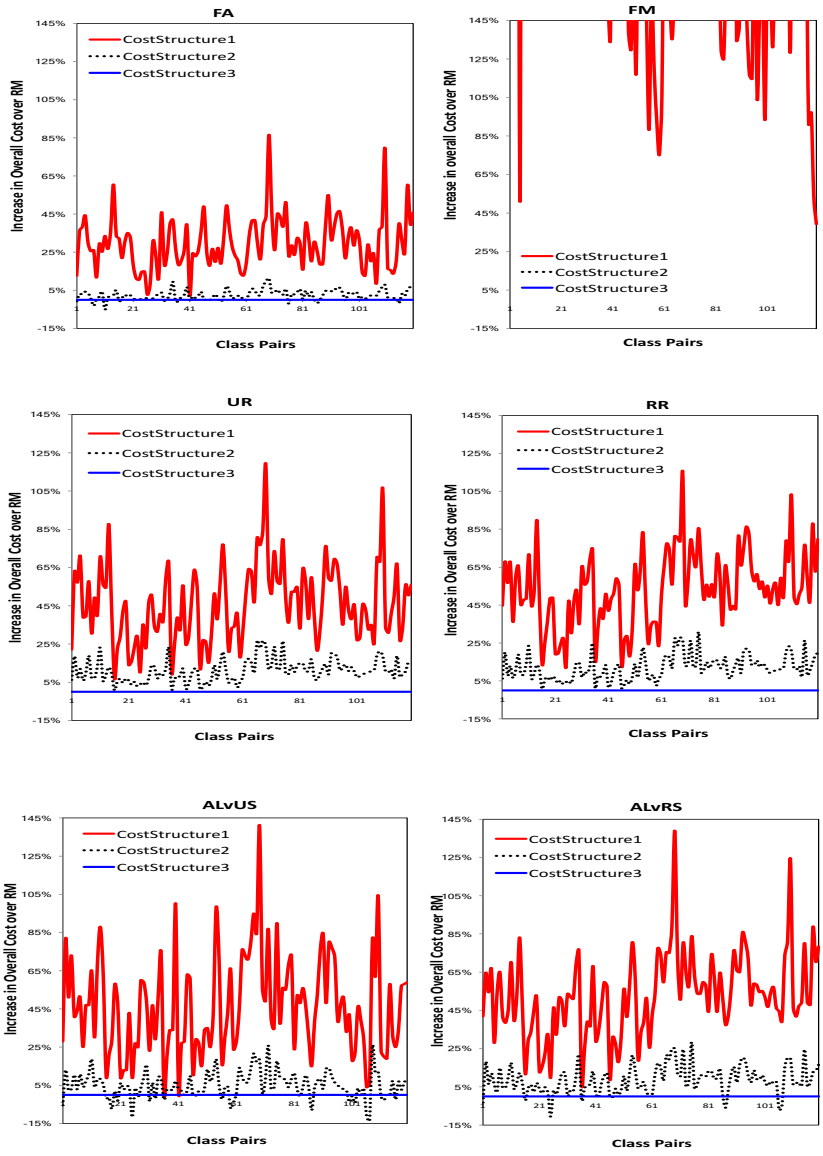


Fig. 5. Percentage increase (with respect to MINECORE) in the overall cost $K^o(\mathcal{D})$ resulting from the 6 baseline methods for each of the 120 class pairs according to the 3 different cost structures. Pairs are listed on the $x$ axis by decreasing cost brought about by MINECORE. For better comparison all figures are displayed across the range [-15%,+145%] on the $y$ axis; in the FM figure (top right) this makes the CostStructure2 and CostStructure3 curves, and most of the CostStructure1 curve, fall way off the ceiling.

The probability calibration phase is taken into consideration for MINECORE but not for the baselines since it is strictly needed for MINECORE but not for the baselines (in the baselines, while we indeed

Table 9. Average computation times (in seconds) per class pair for each cost structure and each method.

| | FA | FM | UR | RR | ALvUS | ALvRS | RM |
|---|---|---|---|---|---|---|---|
| CostStructure1 | 0 | 0 | 11 | 10 | 774 | 750 | 19 |
| CostStructure2 | 0 | 0 | 11 | 10 | 432 | 420 | 19 |
| CostStructure3 | 0 | 0 | 11 | 10 | 11 | 10 | 19 |

use the calibrated probabilities, we might as well have used the original uncalibrated scores, since for the baselines the generated rankings are the same in the two cases; this is not the case for MINECORE).

Table 9 shows that the computational cost of MINECORE is roughly double than that of UR and RR. The reason is that, for each document in $\mathcal{D}$, MINECORE needs to compute $E_y[\Delta^{or}(d)]$ (via Equation 13) and $E_y[\Delta^{op}(d)]$ (via Equation 18) *and* generate the two rankings, while UR and RR only need to generate the two rankings. All in all, the increased computational cost is tolerable (note that in a real e-discovery scenario we would deal with just *one* class pair), given the sizeable reduction in total US$ cost that the use of MINECORE brings about. Concerning ALvUS and ALvRS, instead, we note that their computational cost is dramatically higher than that of MINECORE, while (as seen in Section 4.6) bringing about qualitatively inferior results.

Table 9 also shows that the computational cost of UR, RR, and MINECORE are independent of the cost structure; the reason is that the amount of computation that each of them performs does not depend on the actual values of the unit costs.

While the values in Table 9 are averages (across class pairs), we should note that for each method in {UR, RR, MINECORE} the time is essentially the same for each class pair and each cost structure. This is due to the fact that these methods consist of ranking $\mathcal{D}$ documents twice, irrespectively of the class pair and cost structure involved. Instead, times can vary across class pairs and cost structures quite a lot in ALvUS and ALvRS because different class pairs and cost structures in general give rise to different values of $\tau_r$ or $\tau_p$, which in turn gives rise, for ALvUS and ALvRS, to different numbers of retraining operations. In ALvUS and ALvRS a sizeable part of the computation is due to the retraining-and-reranking operations (one every $b$ manually annotated documents – see Section 4.4), which are interleaved with the reviewers' work.

A further advantage of MINECORE over ALvUS and ALvRS is that all of its processing is instead carried out offline, i.e., before the interaction with the reviewers start; this means that this interaction can occur smoothly, not hampered by intermediate processing phases during which the reviewers are effectively stalled.

All our experiments were run on a machine equipped with a 4-core processor Intel(R) Core(TM) i5-4670 CPU with 16GB of RAM under Red Hat Enterprise Linux 6. (One core only, with no multiple threading, was used though.)

## 5 DISCUSSION

### 5.1 Estimating costs in operational conditions

When using the system in operational conditions, rather than in lab experiments like the ones above, $K^m(\mathcal{D})$ (and, as a consequence, $K^o(\mathcal{D})$) cannot be computed precisely since, even after the entire process has ended, we do not know the true classes of some of the unlabeled documents (the only documents whose true class we know are the ones that have been manually annotated for

both responsiveness and privilege). The best we can do is thus to compute estimates $\hat{C}^m(\mathcal{D})$ and $\hat{C}^o(\mathcal{D})$.[16]

One possible method for computing these estimates is the following. In order to compute $\hat{C}^m(\mathcal{D})$ and $\hat{C}^o(\mathcal{D})$, we need estimates (denoted as $\hat{D}_{ij}$) of the contingency cell values $D_{ij}$ (for $i, j \in \{P, L, W\}$) that result from the entire process. We start by computing estimates of the $D_{ij}$ values that result from Phase 1. In order to compute them, we perform a $k$-fold cross-validation on the training set $Tr$, so that each element of $Tr$ is classified by an automatic classifier, and can thus be assigned to one of the cells $Tr_{ij}$ of the contingency table. We then make the assumption that the training set and the test set are independent and identically distributed.[17] This allows us to compute a maximum-likelihood estimate of $D_{ij}$ as $\hat{D}_{ij} = Tr_{ij} \cdot |Te|/|Tr|$.[18] By applying Equation 5 we obtain

$$\hat{C}^o_1(\mathcal{D}) = \hat{C}^m_1(\mathcal{D}) = \sum_{i,j \in \{P,L,W\}} \lambda^m_{ij} \hat{D}_{ij}$$

which represents the estimate of $K^o(\mathcal{D})$ at the end of Phase 1.

Once annotation by responsiveness has started, every time a document $d$ is manually annotated we would like to update the current estimate of $K^o(\mathcal{D})$ by bringing to bear the reduction in cost that annotating $d$ has brought about. However, we do not know the true class of $d$, so we do not know exactly which values $\hat{D}_{ij}$ we should update. We must thus switch from costs $\hat{C}^o(\mathcal{D})$ to *expected* costs (i.e., risks) $E[\hat{C}^o(\mathcal{D})]$. We first initialize $E[\hat{C}^o(\mathcal{D})] \leftarrow \hat{C}^o_1(\mathcal{D})$, after which we perform each update as

$$E[\hat{C}^o(\mathcal{D})] \leftarrow E[\hat{C}^o(\mathcal{D})] + R_2(d, h_2(d)) + \lambda^a_r - R_1(d, h_1(d)) \tag{22}$$

where $R_\phi(d, h_\phi(d))$ is as in Equation 10. In other words, we add to the current estimate the expected difference in risk that annotating the document has brought about (note that the expression that gets added to $E[\hat{C}^o(\mathcal{D})]$ is exactly $E_y[\Delta^{or}(d)]$ as in Equation 12). Equivalently, once annotation by privilege has started, we perform each update as

$$E[\hat{C}^o(\mathcal{D})] \leftarrow E[\hat{C}^o(\mathcal{D})] + R_3(d, h_3(d)) + \lambda^a_p - R_2(d, h_2(d)) \tag{23}$$

At the end of Phase 3, $E[\hat{C}^o(\mathcal{D})]$ is our final estimate of the overall costs that MINECORE has brought about, while $\hat{C}^o_1(\mathcal{D}) - E[\hat{C}^o(\mathcal{D})]$ is our estimate of the reduction in overall costs that manual annotation has brought about.

### 5.2 Generalizing MINECORE to multi-stage review

MINECORE is easily extended to supporting multi-stage review for application scenarios different from those of e-discovery. While we have described a process that supports just two stages of annotation (for the two classes $c_r$ and $c_p$, respectively) and three alternative actions to perform $(P, L, W)$, the framework can be easily extended to supporting $n$ stages of annotation (for classes $c_1, ..., c_n$) and $k$ alternative actions $A_1, ..., A_k$, where (as in Equation 1) each action is identified by

[16] Consistent with most mathematical literature, we use the caret symbol (ˆ) to indicate estimation.

[17] This assumption is reasonable only if the training set is has been obtained via a random sampling of the set of documents that need to be classified. If the training set has instead been obtained, say, via active learning, this assumption is not satisfied, because active learning chooses the documents to be manually annotated according to a policy that is anything but random; see [14] for alternatives to $k$-fold cross-validation suitable for active learning.

[18] As in many other contexts, the assumption that the training set and the test set are independent and identically distributed may not be satisfied in practice; if it is not, in our case this leads to imprecise estimates of the contingency cell counts. While this may be suboptimal, there is practically nothing we can do about it, since we do not know the real values of these counts; in other words, $k$-fold cross validation is our "best possible shot" at estimating them in the absence of foreknowledge. In a controlled experiment we could exactly measure *how* suboptimal these estimates are by computing costs using the true values of the contingency cells.

a Boolean combination of $c_1, ..., c_n$ and the $k$ Boolean combinations form a partition of the event space.

In this more general case the contingency table $D$ and the cost matrix $\Lambda^m$ of Table 2 become $k \times k$ matrixes, and a cost structure $\Lambda = (\Lambda^m, \Lambda^a)$ consists of the $k \times k$ cost matrix $\Lambda^m$ plus a vector $\Lambda^a = (\lambda_1^a, ..., \lambda_n^a)$ of $n$ annotation costs, one for each class in $\{c_1, ..., c_n\}$. Reformulating Equation 3 (defining the risk associated with choosing action $A_j$ for $d$), Equations 5 and 6 (defininining misclassification cost and annotation cost), and Equation 10 (defininining cost-sensitive classification), to account for these changes, is straightforward.

In this more general case Phase 1 consists of generating $n$ classifiers $h_1, ..., h_n$, and of using them to generate calibrated posterior probabilities $\Pr(c_1|d), ..., \Pr(c_n|d)$ for each unlabelled document $d$. Following this the process consists of additional $n$ phases (instead of the two we have needed for the purposes of e-discovery); remembering what we observed in Section 3.4 concerning how to best sequence these phases, we here assume that $\lambda_1^a \leq ... \leq \lambda_n^a$. Phase $(i + 1)$ (for $i \in \{1, ..., n\}$) consists in determining (via ranking) which of the unlabelled documents should be manually reviewed for $c_i$ and which should not, analogously to what we have done in Phases 2 and 3 in Section 3. No equation among the ones described in Section 3 needs rewriting, since the above-mentioned reformulation of Equations 3, 5, 6, 10 is sufficient to reframe the entire framework in this more general way.

### 5.3 MINECORE and the legal profession

MINECORE has several attributes that are novel from the perspective of the way the legal profession presently performs the e-discovery review process. Adoption of the risk minimization method we have presented will thus turn on the ability of the profession to address the several issues that we identify in this section.

Perhaps most basically, we have assumed that lawyers will be able to conceptualize unit annotation costs and unit misclassification costs in comparable units. Although this has proven to be a useful formalism, one important insight from the literature on behavioral economics is that people often find it difficult to quantify uncertain costs using the same units in which they would express costs that would certainly be incurred. Moreover, the behavioral economics literature contains numerous examples of studies in which the models that we might infer from the choices that people make are inherently inconsistent when viewed rationally [21]. We have assumed for the purposes of our work that some model of costs and risks exists and can be formalized, but in practice the process of designing such models may not be as simple as asking an attorney to assign values to the elements in one of our cost structures. Research on alternative approaches for model elicitation is beyond the scope of our present work, but if these methods are to be adopted there will be a need for serious work on that question.

A second important observation is that we have assumed that both costs and risks accumulate linearly. This is surely a reasonable approximation for annotation costs: training costs and fatigue effects may introduce some nonlinearities, but expecting actual annotation costs to be asymptotically linear does seem reasonable. The same may not be true, however, for misclassification costs. As one example of the potential complexity, with some exceptions single errors can and will be forgiven, since the standard applied in the law is reasonableness, not perfection, but an excessive error rate might be taken as evidence of inattention and thus stiffly penalized. It remains to be seen whether lawyers can agree on linear models for both costs and risks; if not, MINECORE may need to be extended to accommodate the specific types of nonlinearities that lawyers would wish to model.

In an adversarial legal system, such as the one in the United States of America, lawyers must nonetheless agree on some things. In current practice, for example, lawyers negotiate on questions such as what technical approaches (e.g., manual query formulation, simple passive machine learning,

or active learning – see Section 6) will be used. Our risk minimization framework will give lawyers more to discuss, since adopting our approach would mean that they would ultimately need to agree on both the cost structure and the way in which error probabilities are estimated.

Finally, lawyers may even need to change their views on what it means for a decision to be "right". In MINECORE we define a decision to be right if and only if it minimizes the overall risk. Because the cost structure may be highly skewed, there could well be cases in which risk minimization would rationally select a decision that is less likely to be correct if the cost of making such an error is low.[19] Essentially, by quantifying what it means to be "wrong" we enter a world in which it can be right to be wrong. That alone may be enough to keep the discussion between the text classification and legal worlds going for some time.

## 6 RELATED WORK

**Predictive coding in technology-assisted review.** The state-of-the-art in the application of predictive coding to technology-assisted review in e-discovery is reviewed in [27], and has been the subject of many recent studies [2, 3, 7, 8, 15, 30, 31, 38]. All of the work cited here (and the vast majority of published work on predictive coding in TAR) address review for responsiveness, and not review for privilege; the latter has been addressed only sporadically (and tentatively) up to now [13, 36, 37].

Cormack and Grossman classify the predictive coding protocols used in TAR into three classes [6]. In each of these classes (i) a "seed set" of documents (usually identified via keyword search) is manually annotated for use as initial training data; (ii) this initial training set is expanded into a more refined training set by selecting new documents and asking the reviewers to manually annotate them; and (iii) the classifier trained on this expanded training set ranks the remaining documents in $\mathcal{D}$ in terms of probability of responsiveness, so that human reviewers may annotate them starting from the top of the list and identify as many responsive documents as possible. The classes identified by Cormack and Grossman differ in terms of how the new documents of Step (ii) are selected: (a) random selection in *simple passive learning* (SPL), (b) selection by uncertainty (as in our ALvUS baseline) in *simple active learning* (SAL), and (c) selection by relevance (as in our ALvRS baseline) in *continuous active learning* (CAL). We want to stress that our work is not concerned with how Steps (i) and (ii) are accomplished, and instead redefines Step (iii), by (a) bringing privilege (alongside responsiveness) into play, (b) bringing annotation costs and misclassification costs into play as explicit variables of the model, and (c) assuming that also documents which have not been manually annotated (by responsiveness, or by privilege, or both) can be produced to the requesting party, provided that the estimated risk of doing so is low enough.

**Multi-stage (text) classification.** Other systems for two-stage (or multi-stage) classification have been proposed, either for textual documents or for other items, but are substantially different from MINECORE. In some cases, the rationale of performing classification in more than one stage if to have cheap early-stage classifiers act as coarse filters, and then more expensive and more efficient classifiers take the final decision on the documents that have passed the previous filters [32]; here the classes involved in the different stages are the same, unlike in MINECORE where the

[19] As an example, assume CostStructure1, and assume we know for certain that document $d$ is responsive (e.g., because it has been manually annotated as such); we thus need to decide whether $d$ should be produced or logged. According to CostStructure1 (see Table 5), producing when we should instead log is 4 times as expensive as logging when we should instead produce (since $\lambda^m_{PL}/\lambda^m_{LP} = 4$). If $\Pr(c_p|d) = .30$, then probabilistic considerations alone would tell us that we should produce $d$ (since $\Pr(c_p|d) = .30 < \Pr(\overline{c}_p|d) = .70$); however, when we bring cost considerations in, we will rationally decide to log $d$, since the risk involved in logging $d$ is $\lambda^m_{LP}\Pr(\overline{c}_p|d) = 150.00$ US$$\times .70 = 105.00$ US$ while the risk involved in producing $d$ is $\lambda^m_{PL}\Pr(c_p|d) = 600.00$ US$ $\times .30 = 180.00$ US$. Given CostStructure1, only when $\Pr(c_p|d) < .20$ we will rationally opt for producing $d$.

two stages deal with two different classes (responsiveness and privilege). Yet a different example is hierarchical classification (see e.g., [23, 39]), where a decision is taken whether or not to assign a fine-grained class (e.g., 'Baseball') only after a coarse-grained class (e.g., 'Sports') has been assigned. MINECORE is different from all the systems above, e.g., because it is a mixed-initiative (human and machine) system while they are not; because in MINECORE the 2nd stage (privilege) is carried out independently of the outcome of the 1st (responsiveness), unlike in the systems above; because MINECORE uses cost-sensitivity while the systems above do not; and because in the systems above there is no combination of the decisions taken in the different stages, while there is in MINECORE.

**Evaluating technology-assisted review in e-discovery.** A number of papers in the field of predictive coding for TAR do not use, as evaluation measures, cost-sensitive measures such as the one in Equation 7, but exclusively use "effort curves" to plot recall as a function of the number of training documents (see e.g., [6, 31]). While the number of training documents used indeed brings cost into the picture (since annotating them has a cost), effort curves reflect the costs of just a single review stage.

**Cost-sensitive active learning**. Some aspects of MINECORE are reminiscent of past efforts in cost-sensitive active learning. The work closest in spirit to ours is [22], where the cost of manually annotating a document is (as in MINECORE) an explicit variable in a model that ranks items for presentation to a human reviewer. However, the goal of [22] is not prioritizing the documents whose annotation would bring about the highest reduction in overall cost, but annotating the documents that would prove most valuable when used as training examples for retraining the classifier. In other words, the task we deal with is not generating new training data that allow us to train a more accurate classifier, but reviewing a set of documents at the minimum possible overall cost; this difference in goals shapes the difference between that technique and MINECORE. Other works in cost-sensitive active learning (e.g., [34, 35]) are even more different from ours since they focus on modelling the fact that different types of items may involve different annotation costs, an issue that we do not address in MINECORE.

**Minimizing costs in classification endeavors**. Our focus is complementary to that of [2], which addresses the problem of minimizing total annotation costs for a fully automated classifier, including both annotation for training and for evaluation. Rather, we focus here on the costs of correcting the results of automated classifiers – a process that the authors of [2] do not model. Unlike them, for the purposes of our work we treat training costs as fixed.

**Utility theory for technology-assisted review**. This work applies some of the principles described in [5], which presents a utility-theoretic model for ranking automatically classified documents in order to optimize the work of human reviewers who annotate some of them. One major difference is that [5] is more theoretical in spirit, while the present work can be seen as an application to an e-discovery context of some of the principles presented there. Another major difference is that [5] does not consider annotation cost, and focuses on misclassification cost; as a result, the amount of documents that the reviewer annotates is a free variable of their model, and the evaluation is carried out for different values of this variable. In this work, instead, we also consider annotation cost, and we derive the optimal amount of documents that the reviewer should annotate as a function of unit misclassification costs and annotation costs. Yet another difference is that in [5] the cost matrix emerges from the evaluation function (e.g., $F_1$), which is given as an input to the problem, while in MINECORE it is the evaluation function (Equation 7) which emerges from the cost structure, which is given as an input to the problem. Finally, we should note that, while [5] discusses two different models (the "static" and the "dynamic" model), we here discuss a single "static" model; this derives from the fact that the evaluation function we use is (unlike the $F_1$ measure used in [5]) linear in its free variables, and linearity makes the static and the dynamic models coincide.

## 7 CONCLUSION

We have developed MINECORE, a framework for jointly minimizing the expected total cost of review for responsiveness and privilege. This framework, which is based on utility theory and relies on multi-stage cost-sensitive ranking by uncertainty, accounts for the fact that misclassification costs are not defined individually at the level of the individual aspect (e.g., responsiveness only), but rather at the global, two-stage level (i.e., responsiveness and privilege), so the two issues are best addressed jointly.

Differently from other competing models (e.g., CAL), MINECORE assumes that a document might be produced to the requesting party even if it has not been manually certified to be responsive and nonprivileged. A "minimum risk principle" is adopted when deciding which course of action ("Produce", "Log", "Withhold") should be chosen for a document, so that the action which is expected to bring about the smallest cost is chosen. Human annotation effort is directed towards globally reducing this expected cost for the entire universe of documents to be searched, and documents are manually reviewed only insofar as the cost of reviewing them is expected to be offset by the reduction in cost that reviewing them is expected to bring about. Indeed, MINECORE is characterized by the analytical derivation of an optimality criterion, in the form of two thresholds $\tau_r$ and $\tau_p$ that indicate when the reviewers should stop annotating. In other models (say, in active learning models), the stopping criteria used are mainly heuristic (see [31, Section 2c] for a discussion of this point). What enables us to analytically derive these optimal thresholds is the fact that we explicitly model both annotation costs and misclassification costs, which means that an optimal threshold may be defined as the one that best trades off between the two.

Our conclusions are supported by substantial experimentation, wherein 7 different methods (MINECORE plus 6 baselines) were tested on a collection of nearly 200,000 documents, using 120 pairs of classes (playing the role of the responsive and the privileged classes, respectively) and three different cost structures.

There are several ways in which MINECORE could be extended. One might consist of conducting experiments with types of classifiers (e.g., a transductive SVM [20]) that are (see also Section 1) better suited to the finite nature of the universe of documents that a specific e-discovery endeavor needs to address. This move might bring about better posterior probabilities $\Pr_1(c_r|d)$ and $\Pr_1(c_p|d)$, which would likely result in higher cost-effectiveness. A second way forward might consist of switching to a nonlinear cost model, since attorneys and the courts demand not perfection but rather reasonableness; models that forgive a few errors but impose steep penalties for systematic mistake patterns might better represent actual practice in e-discovery. A third extension of this work might consist of relaxing two simplifying assumptions we have built into MINECORE, i.e. (a) that human reviewers are infallible (i.e., they do not bring about any misclassification costs), and (b) that the costs of setting up automated classifiers can be ignored. MINECORE is, all in all, a reasonable first approximation, since assumption (a) biases the evaluation in favor of manual endeavors, while assumption (b) generates an opposite bias in favor of automatic tools; however, a solution in which both simplifications are removed might provide a more accurate picture of the benefits of our risk minimization model.

Finally, we should emphasize that e-discovery is not unique in using multiple stages of review to balance multiple goals. Similar situations arise in other settings, such as when fostering government transparency [4] or as when archivists seek to open previously restricted collections for unrestricted use by researchers. Indeed, as our society generates ever increasing quantities of digital content in which the banal is intermixed with the crucial, which in turn is intermixed with the sensitive, techniques such as those explored in this work will assume increasing importance.

## ACKNOWLEDGEMENTS

We thank Jason Baron, Maura Grossman and David Lewis, for discussions that helped us to develop representative cost structures for the e-discovery review task. We also thank Giacomo Berardi for useful discussions and for sharing code with us. This work has been supported in part by NSF grants 1065250 and 1618695.

## REFERENCES

[1] Paul Anand. 1993. *Foundations of Rational Choice under Risk.* Oxford University Press, Oxford, UK.

[2] Mossaab Bagdouri, David D. Lewis, Douglas W. Oard, and William Webber. 2013. Towards minimizing the annotation cost of certified text classification. In *Proceedings of the 22nd ACM Conference on Information and Knowledge Management (CIKM 2013)*. San Francisco, US, 989–998. DOI: http://dx.doi.org/10.1145/2505515.2505708

[3] Jason R. Baron, Michael D. Berman, and Ralph C. Losey (Eds.). 2016. *Perspectives on Predictive Coding and Other Advanced Search and Review Technologies for the Legal Practitioner.* ABA Book Publishing, Washington, US.

[4] Giacomo Berardi, Andrea Esuli, Craig Macdonald, Iadh Ounis, and Fabrizio Sebastiani. 2015b. Semi-Automated Text Classification for Sensitivity Identification. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management (CIKM 2015)*. Melbourne, AU, 1711–1714. DOI: http://dx.doi.org/10.1145/2806416.2806597

[5] Giacomo Berardi, Andrea Esuli, and Fabrizio Sebastiani. 2015a. Utility-Theoretic Ranking for Semi-Automated Text Classification. *ACM Transactions on Knowledge Discovery from Data* 10, 1 (2015), Article 6. DOI: http://dx.doi.org/10.1145/2742548

[6] Gordon V. Cormack and Maura R. Grossman. 2014. Evaluation of machine-learning protocols for technology-assisted review in electronic discovery. In *Proceedings of the 37th ACM Conference on Research and Development in Information Retrieval (SIGIR 2014)*. Gold Coast, AU, 153–162. DOI: http://dx.doi.org/10.1145/2600428.2609601

[7] Gordon V. Cormack and Maura R. Grossman. 2015a. Autonomy and Reliability of Continuous Active Learning for Technology-Assisted Review. (2015). CoRR abs/1504.06868.

[8] Gordon V. Cormack and Maura R. Grossman. 2015b. Multi-Faceted Recall of Continuous Active Learning for Technology-Assisted Review. In *Proceedings of the 38th ACM Conference on Research and Development in Information Retrieval (SIGIR 2015)*. Santiago, CL, 763–766. DOI: http://dx.doi.org/10.1145/2766462.2767771

[9] Gordon V. Cormack and Mona Mojdeh. 2009. Machine learning for information retrieval: TREC 2009 Web, Relevance Feedback and Legal Tracks. In *Proceedings of the 18th Text Retrieval Conference (TREC 2009)*. Gaithersburg, US.

[10] Morris H. DeGroot and Stephen E. Fienberg. 1983. The comparison and evaluation of forecasters. *The Statistician* 32, 1/2 (1983), 12–22. DOI: http://dx.doi.org/10.2307/2987588

[11] Pedro M. Domingos and Michael J. Pazzani. 1997. On the Optimality of the Simple Bayesian Classifier under Zero-One Loss. *Machine Learning* 29, 2-3 (1997), 103–130.

[12] George Forman. 2006. Tackling concept drift by temporal inductive transfer. In *Proceedings of the 29th ACM International Conference on Research and Development in Information Retrieval (SIGIR 2006)*. Seattle, US, 252–259. DOI: http://dx.doi.org/10.1145/1148170.1148216

[13] Manfred Gabriel, Chris Paskach, and David Sharpe. 2013. The Challenge and Promise of Predictive Coding for Privilege. In *Proceedings of the ICAIL 2013 Workshop on Standards for Using Predictive Coding (DESI V)*. Roma, IT.

[14] Aubrey Gress and Ian Davidson. 2015. Accurate Estimation of Generalization Performance for Active Learning. In *Proceedings of the 15th IEEE International Conference on Data Mining (ICDM 2015)*. Atlantic City, US, 131–140. DOI: http://dx.doi.org/10.1109/icdm.2015.137

[15] Maura R. Grossman and Gordon V. Cormack. 2011. Technology-assisted review in e-discovery can be more effective and more efficient than exhaustive manual review. *Richmond Journal of Law and Technology* 17, 3 (2011), Article 5.

[16] Maura R. Grossman and Gordon V. Cormack. 2013. The Grossman-Cormack glossary of technology-assisted review, with foreword by John M. Facciola, U.S. Magistrate Judge. *Federal Courts Law Review* 7, 1 (2013), 1–34.

[17] Sherry B. Harris and Paul H. McVoy (Eds.). 2014. *The Sedona Conference Glossary: E-Discovery and Digital Information Management* (4th ed.). The Sedona Conference, Phoenix, US. Available at http://bit.ly/2Bhz0TB.

[18] Thorsten Joachims. 1999. Making large-scale SVM Learning Practical. In *Advances in Kernel Methods – Support Vector Learning*, Bernhard Schölkopf, Christopher J. Burges, and Alexander J. Smola (Eds.). The MIT Press, Cambridge, US, Chapter 11, 169–184.

[19] Thorsten Joachims. 2000. Estimating the Generalization Performance of a SVM Efficiently. In *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*. Stanford, US, 431–438.

[20] Thorsten Joachims. 2006. Transductive Support Vector Machines. In *Semi-Supervised Learning*, Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien (Eds.). The MIT Press, Cambridge, US, 105–117.

[21] Daniel Kahneman and Amos Tversky. 1979. Prospect Theory: An Analysis of Decision Under Risk. *Econometrica* 47, 2

(1979), 263–291.

[22] Ashish Kapoor, Eric Horvitz, and Sumit Basu. 2007. Selective Supervision: Guiding Supervised Learning with Decision-Theoretic Active Learning. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence (IJCAI 2007)*. San Francisco, US, 877–882.

[23] Daphne Koller and Mehran Sahami. 1997. Hierarchically Classifying Documents Using Very Few Words. In *Proceedings of the 14th International Conference on Machine Learning (ICML 1997)*. Nashville, US, 170–178.

[24] David D. Lewis and William A. Gale. 1994. A sequential algorithm for training text classifiers. In *Proceedings of the 17th ACM International Conference on Research and Development in Information Retrieval (SIGIR 1994)*. Dublin, IE, 3–12. DOI: http://dx.doi.org/10.1007/978-1-4471-2099-5_1

[25] David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. 2004. RCV1: A New Benchmark Collection for Text Categorization Research. *Journal of Machine Learning Research* 5 (2004), 361–397.

[26] Douglas Oard, Jyothi Vinjumur, and Fabrizio Sebastiani. 2017. When is it Rational to Review for Privilege?. In *Presented at the ICAIL 2017 Workshop on Using Advanced Data Analysis in eDiscovery & Related Disciplines to Identify and Protect Sensitive Information in Large Collections (DESI VII)*. London, UK.

[27] Douglas W. Oard and William Webber. 2013. Information Retrieval for E-discovery. *Foundations and Trends in Information Retrieval* 7, 2/3 (2013), 99–237. DOI: http://dx.doi.org/10.1561/1500000025

[28] John C. Platt. 2000. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In *Advances in Large Margin Classifiers*, Alexander Smola, Peter Bartlett, Bernard Schölkopf, and Dale Schuurmans (Eds.). The MIT Press, Cambridge, MA, 61–74.

[29] Stephen E. Robertson and Ian Soboroff. 2002. The TREC 2002 Filtering Track Report. In *Proceedings of the 11th Text REtrieval Conference (TREC 2002)*. Gaithersburg, US.

[30] Herbert L. Roitblat, Anne Kershaw, and Patrick Oot. 2010. Document Categorization in Legal Electronic Discovery: Computer Classification vs. Manual Review. *Journal of the American Society for Information Science and Technologies* 61, 1 (2010), 70–80. DOI: http://dx.doi.org/10.1002/asi.21233

[31] Tanay K. Saha, Mohammad Al Hasan, Chandler Burgess, M. Ahsan Habib, and Jeff Johnson. 2015. Batch-mode active learning for technology-assisted review. In *Proceedings of the 3rd IEEE International Conference on Big Data (Big Data 2015)*. Santa Clara, US, 1134–1143. DOI: http://dx.doi.org/10.1109/bigdata.2015.7363867

[32] Ted E. Senator. 2005. Multi-Stage Classification. In *Proceedings of the 5th IEEE International Conference on Data Mining (ICDM 2005)*. Houston, US, 386–393. DOI: http://dx.doi.org/10.1109/ICDM.2005.102

[33] Burr Settles. 2012. *Active learning*. Morgan & Claypool Publishers, San Rafael, US. DOI: http://dx.doi.org/10.2200/s00429ed1v01y201207aim018

[34] Burr Settles, Mark Craven, and Lewis Friedland. 2008. Active Learning with Real Annotation Costs. In *Proceedings of the NIPS Workshop on Cost-Sensitive Learning*. Vancouver, CA.

[35] Katrin Tomanek and Udo Hahn. 2010. A Comparison of Models for Cost-Sensitive Active Learning. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010)*. Beijing, CN, 1247–1255.

[36] Jyothi K. Vinjumur. 2015. Evaluating expertise and sample bias effects for privilege classification in e-discovery. In *Proceedings of the 15th International Conference on Artificial Intelligence and Law (ICAIL 2015)*. San Diego, US, 119–127. DOI: http://dx.doi.org/10.1145/2746090.2746101

[37] Jyothi K. Vinjumur, Douglas W. Oard, and Amittai Axelrod. 2016. An AID for Avoiding Inadvertent Disclosure: Supporting Interactive Review for Privilege in E-Discovery. In *Proceedings of the 1st ACM SIGIR Conference on Human Information Interaction and Retrieval (CHIIR 2016)*. Chapel Hill, US, 53–62. DOI: http://dx.doi.org/10.1145/2854946.2854964

[38] Jyothi K. Vinjumur, Douglas W. Oard, and Jiaul H. Paik. 2014. Assessing the reliability and reusability of an e-discovery privilege test collection. In *Proceedings of the 37th ACM Conference on Research and Development in Information Retrieval (SIGIR 2014)*. Gold Coast, AU, 1047–1050. DOI: http://dx.doi.org/10.1145/2600428.2609506

[39] Erik D. Wiener, Jan O. Pedersen, and Andreas S. Weigend. 1995. A neural network approach to topic spotting. In *Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval (SDAIR 1995)*. Las Vegas, US, 317–332.

[40] Bianca Zadrozny and Charles Elkan. 2002. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the 8th ACM International Conference on Knowledge Discovery and Data Mining (KDD 2002)*. Edmonton, CA, 694–699. DOI: http://dx.doi.org/10.1145/775107.775151

Received December 2017; revised June 2018; accepted August 2018