Mach Learn (2018) 107:797–824
https://doi.org/10.1007/s10994-017-5671-3

Learning with rationales for document classification

Manali Sharma[1] · **Mustafa Bilgic**[1]

Received: 5 July 2015 / Accepted: 6 September 2017 / Published online: 19 December 2017

Abstract We present a simple and yet effective approach for document classification to incorporate rationales elicited from annotators into the training of any off-the-shelf classifier. We empirically show on several document classification datasets that our classifier-agnostic approach, which makes no assumptions about the underlying classifier, can effectively incorporate rationales into the training of multinomial naïve Bayes, logistic regression, and support vector machines. In addition to being classifier-agnostic, we show that our method has comparable performance to previous classifier-specific approaches developed for incorporating rationales and feature annotations. Additionally, we propose and evaluate an active learning method tailored specifically for the learning with rationales framework.

Keywords Document classification · Learning with rationales · Active learning

1 Introduction

Annotating documents for supervised learning is a tedious, laborious, and time consuming task for humans. Given huge amounts of unlabeled documents, it is impractical for annotators to go over each document and provide a label. To reduce the annotation time and effort, various approaches such as semi-supervised learning (Chapelle et al. 2006) that utilizes both labeled and unlabeled data, and active learning (Settles 2012) that carefully chooses instances for annotation have been developed. To further minimize the human effort, recent work looked at eliciting domain knowledge, such as rationales and feature annotations, from the annotators instead of just the labels of documents.

Editor: Hal Daume III.

✉ Mustafa Bilgic
mbilgic@iit.edu

Manali Sharma
msharm11@hawk.iit.edu

[1] Illinois Institute of Technology, 10 W 31st Street, Chicago, IL 60616, USA

Humans can classify instances based on their prior knowledge about feature-class correlations. In order for the classifier to learn similar feature-class correlations from the data, it needs to see many labeled instances. For example, consider the task of sentiment analysis for movie reviews where a classifier is tasked with classifying the reviews as overall positive or overall negative. When the classifier is presented with a negative review that reads "I saw this movie with my friends over the weekend. The movie was terrible.", the classifier does not know which terms in this review are responsible for classifying it as a negative review. Unless the classifier has observed many more negative reviews that have the word "terrible" in them, it would not know that "terrible" is a negative sentiment word, and unless it has seen many positive and negative reviews that have the words "friend" and "weekend" in them, it would not know that these words are potentially neutral sentiment words. In domains where labeled data is scarce, teasing out this kind of information is like searching for a needle in haystack. In learning with rationales framework, in addition to a label, the annotator provides a rationale, pointing out the phrases that are responsible for the assigned label, enabling the classifier to quickly identify the important feature-class correlations and speed up the learning.

A bottleneck in effective utilization of rationales elicited from annotators is that the traditional supervised learning approaches cannot readily handle the elicited rich feedback. To address this issue, many methods have been developed that are classifier-specific. Examples include knowledge-based neural networks (Towell and Shavlik 1994; Girosi and Chan 1995; Towell et al. 1990), knowledge-based support vector machines (Fung et al. 2002), pooling multinomial naïve Bayes (Melville and Sindhwani 2009), incorporating feature annotation into locally-weighted logistic regression (Das et al. 2013), incorporating constraints into the training of naïve Bayes (Stumpf et al. 2007), and converting rationales and feature annotations into constraints for support vector machines (Small et al. 2011; Zaidan et al. 2007). Being classifier-specific limits their applicability when one does not know which classifier is best suited for his/her domain and hence would like to test several classifiers, necessitating a simple and generic approach that can be utilized by several off-the-shelf classifiers.

In this article we present a simple and yet effective approach that can incorporate the elicited rationales into the training of any off-the-shelf classifier. This article builds upon our earlier work (Sharma et al. 2015). Our main contributions are:

- We present a simple and intuitive approach for incorporating rationales into the training of any off-the-shelf classifier for document classification.
- We empirically evaluate our method on several document classification datasets and show that our method can effectively incorporate rationales into the training of naïve Bayes, logistic regression, and support vector machines using binary and tf-idf representations of the documents.
- We present results showing how much a document annotated with a label *and* a rationale is worth compared to the document annotated with just the label, allowing one to judge whether the extra time spent on providing rationales is worth the extra effort.
- We evaluate our method on user-annotated dataset provided by Zaidan et al. (2008) and show that our approach performs well with user-annotated rationales, which could be noisy.
- We compare our method to Zaidan et al. (2007), which was specifically designed for incorporating rationales into the training of support vector machines, and show that our method has comparable performance.
- We compare our method to Melville and Sindhwani (2009), which was specifically designed for incorporating feature annotations into the training of multinomial naïve Bayes, and show that our method has comparable performance.

– We compare our method to Das et al. (2013), which was specifically designed for incorporating feature labels into the training of locally-weighted logistic regression, and show that our method has comparable performance.
– We propose and evaluate a novel active learning approach specifically tailored for utilizing the rationales provided by the labeler.

The rest of the article is organized as follows. In Sect. 2, we provide a brief background on eliciting rationales in the context of active learning. In Sect. 3, we describe our approach for incorporating rationales into the training of classifiers, compare the improvements provided by incorporating rationales into learning to traditional learning that does not use rationales, and evaluate our approach on a dataset with user-annotated rationales. In Sect. 4, we compare our method to three baselines, Melville and Sindhwani (2009), Zaidan et al. (2007), and Das et al. (2013). In Sect. 5, we present an active learning method using the learning with rationales framework and present relevant results. Finally, we discuss future work in Sect. 6, discuss related work in Sect. 7, and conclude in Sect. 8.

2 Background

Let $\mathcal{D}$ be a set of document-label pairs $\langle x, y\rangle$, where the label (value of y) is known for only a small subset $\mathcal{L} \subset \mathcal{D}$ of documents: $\mathcal{L} = \{\langle x, y\rangle\}$ and the rest, $\mathcal{U} = \mathcal{D} \setminus \mathcal{L}$, consists of the unlabeled documents: $\mathcal{U} = \{\langle x, ?\rangle\}$. We assume that each document x^i is represented as a vector of features (most commonly as a bag-of-words model with a dictionary of predefined set of phrases, which can be unigrams, bigrams, etc.): $x^i \triangleq \{f_1^i, f_2^i, \cdots, f_n^i\}$. Each feature f_j^i represents the binary presence (or absence), frequency, or tf-idf representation of the word/phrase j in document x^i. Each label $y \in \mathcal{Y}$ is a discrete-valued variable: $\mathcal{Y} \triangleq \{y_1, y_2, \cdots, y_l\}$.

Typical greedy active learning algorithms iteratively select an informative document $\langle x^*, ?\rangle \in \mathcal{U}$ according to utility-based heuristics, query a labeler for its label y^*, and incorporate the new document $\langle x^*, y^*\rangle$ into the training set, $\mathcal{L}$. This process continues until a stopping criterion is met, usually until a given budget, B, is exhausted.

In the learning with rationales framework, in addition to querying for label y^* of document x^*, the active learner asks the labeler to provide a rationale, $R(x^*)$, for the chosen label. The rationale in its most general form consists of a subset of the terms that are present in document x^*: $R(x^*) = \{f_k^* : k \in x^*\}$. Note that there might be cases where the labeler cannot pinpoint any phrase as a rationale, in which case $R(x^*)$ is allowed to be empty (ϕ). The labeled set now contains the document-label-rationale triplets $\langle x^*, y^*, R(x^*)\rangle$, instead of the document-label pairs $\langle x^*, y^*\rangle$. Algorithm 1 formally describes the active learning process that elicits rationales from the labeler.

Algorithm 1 Active Learning with Rationales

Input: $\mathcal{U}$ - unlabeled documents, $\mathcal{L}$ - labeled documents, θ - underlying classification model, B - budget
repeat
  $x^* = \underset{x \in \mathcal{U}}{\operatorname{argmax}}\, utility(x|\theta)$
  request label and rationale for this label
  $\mathcal{L} \leftarrow \mathcal{L} \cup \{\langle x^*, y^*, R(x^*)\rangle\}$
  $\mathcal{U} \leftarrow \mathcal{U} \setminus \{\langle x^*\rangle\}$
  Train θ on $\mathcal{L}$
until Budget B is exhausted; e.g., $|\mathcal{L}| = B$

The goal of eliciting rationales is to improve the learning efficiency by incorporating domain knowledge. However, it is not trivial to integrate domain knowledge into the state-of-the-art classifiers, such as logistic regression and support vector machines, because the traditional classifiers are able to handle only $\langle x, y\rangle$ pairs and they cannot readily handle $\langle x, y, R(x)\rangle$ triplets. In order to incorporate the additional rationales or feature annotations into learning, a few classifier-specific approaches have been developed, that *modify* the way a classifier is trained. For example, Zaidan et al. (2007) and Raghavan and Allan (2007) introduced constraints for support vector machines to incorporate rationales. Melville and Sindhwani (2009) incorporated feature annotation into multinomial naïve Bayes by training two multinomial naïve Bayes models, one on labeled instances and the other on labeled features, and used linear pooling to combine the two models. Das et al. (2013) utilized locally-weighted logistic regression to incorporate feature labels into logistic regression by locally fitting a logistic function on instances around a small neighborhood of test instances and taking into account the labeled features. We next describe our approach that can readily incorporate rationales into any classifier by modifying the training data, without requiring changes to the training algorithm of a classifier.

3 Learning with rationales

In this section, we first provide the formulation of our approach to incorporate rationales into learning and then present results comparing *learning with rationales* (LwR) to *learning without rationales* (Lw/oR) on four document classification datasets. We evaluate our approach using multinomial naïve Bayes, logistic regression, and support vector machines classifiers.

3.1 Training a classifier using labels and rationales

Like most previous work, we assume that the rationales, i.e. the phrases, returned by the labeler already exist in the dictionary of the vectorizer. Hence, the rationales correspond to features in our vector representation. It is possible that the labeler returns a phrase that is currently not in the dictionary; for example, the labeler might return a phrase that consists of three words whereas the representation has single words and bi-grams only. In that case, the representation can be enriched by creating and adding a new feature that represents the phrase returned by the labeler.

Our simple approach works as follows: we modify the features of the annotated document $\langle x^i, y^i, R(x^i)\rangle$ to emphasize the rationale(s) and de-emphasize the remaining phrases in that document. We simply multiply the features corresponding to phrase(s) that are returned as rationale(s) by weight r and we multiply the remaining features in the document by weight o, where $r > o$, and r and o are hyper-parameters. The modified document becomes:

$$x'^i = \left\langle r \times f^i_j, \forall f^i_j \in R(x^i); o \times f^i_j, \forall f^i_j \notin R(x^i) \right\rangle \tag{1}$$

Note that the rationales are tied to documents for which they were provided as rationales. One phrase might be a rationale for the label of one document and yet it might not be a rationale for the label of another document. Hence, the feature weightings are done at the document level, rather than globally. To illustrate this concept, we provide an example dataset below with three documents. In these documents, the words that are returned as rationales are underlined.

Table 1 The Lw/oR binary representation (top) and its LwR transformation (bottom) for Documents 1, 2, and 3. Stop words are removed. LwR multiplies the rationales by weight r and other features by weight o

	great	movie	plot	performance	actor	terrible	avoid	outdoor	cinema	atmosphere	terrific
Lw/oR Representation (binary)											
Document 1	1	1									
Document 2	1		1	1	1	1	1				
Document 3	1	1						1	1	1	1
LwR Transformation of the binary Lw/oR representation											
Document 1	r	o									
Document 2	o		o	o	o	r	r				
Document 3	o	o						o	o	o	r

Document 1: This is a great movie.

Document 2: The plot was great, but the performance of the actors was terrible. Avoid it.

Document 3: I've seen this at an outdoor cinema; great atmosphere. The movie was terrific.

As these examples illustrate, the word "great" appears in all three documents, but it is marked as a rationale only for *Document 1*. Hence, we do not weight the rationales globally; rather, we modify only the labeled document using its particular rationale. Table 1 illustrates the Lw/oR and LwR representations for these documents.

Our approach modifies the training data, in which the rationale features are weighted higher than the other features, and hence our approach can incorporate rationales into the training of any off-the-shelf classifier, without requiring changes to the training algorithm of a classifier. In our approach, the training algorithm of a classifier uses the modified training data to estimate the parameters of the model. This approach is simple, intuitive, and classifier-agnostic. As we will show later, it is quite effective empirically as well. To gain a theoretical understanding of this approach, consider the work on regularization: the aim is to build a sparse/simple model that can capture the most important features of the training data and thus have large weights for important features and small/zero weights for irrelevant features. For example, consider the gradient of weight w_j for feature f_j for logistic regression with l_2 regularization (assuming y is binary with 0/1):

$$\nabla w_j = C \times \sum_{x^l \in \mathcal{L}} f_j^l \times (y^l - P(y = 1|x^l)) - w_j \tag{2}$$

where C is the complexity parameter that balances between fit to the data and the model complexity. With our rationales framework, the gradient for w_j will be:

$$\nabla w_j = C \times \left(\sum_{x^l \in \mathcal{L}: f_j^l \in R(x^l)} r \times f_j^l \times (y^l - P(y^l = 1|x^l)) + \sum_{x^l \in \mathcal{L}: f_j^l \notin R(x^l)} o \times f_j^l \times (y^l - P(y^l = 1|x^l)) \right) - w_j \quad (3)$$

In Eq. 3, feature f_j contributes more to the gradient of weight w_j when a document in which it is marked as a rationale is misclassified. When f_j appears in another document x^k, but is not a rationale, its contribution to the gradient is muted by o. Hence, when $r > o$, this framework implicitly provides more granular (per instance-feature combination) regularization by placing a higher importance on the contribution of the rationales versus non-rationales in each document.[1]

Note that in our framework, the rationales are tied to their own documents; that is, we do not weight rationales and non-rationales globally. In addition to providing more granular regularization, this approach has the benefit of allowing different rationales to contribute differently to the objective function of the trained classifier. For example, consider the case where the number of documents in which word f_j (e.g., "excellent") is marked as a rationale is much more than the number of documents in which another word f_k (e.g., "good") is marked as a rationale. In this case, the first summation term in Eq. 3 will range over more documents for the gradient of w_j compared to the gradient of w_k, giving more importance to w_j than to w_k. In the traditional feature annotation work, this can be achieved only if the labeler can rank the features; but then, it is often very difficult, if not impossible, for the labelers to determine how much more important one feature is compared to another.

3.2 Experiments comparing LwR to Lw/oR

In this section, we first describe the settings, datasets, and classifiers used for our experiments and how we simulated a human labeler to provide rationales. Then, we present results comparing the learning curves achieved with *learning without rationales* (Lw/oR) and *learning with rationales* (LwR).

3.2.1 Methodology

For this study, we used four document classification datasets. IMDB dataset consists of movie reviews (Maas et al. 2011). Nova is a text classification dataset used in active learning challenge (Guyon 2011). SRAA[2] dataset consists of documents that discuss either auto or aviation. WvsH[3] is a 20 Newsgroups dataset using the Windows vs. hardware categories. We provide the description of these datasets in Table 2. IMDB and WvsH had separate train and test datasets. For NOVA and SRAA datasets, we randomly selected two-thirds of the documents as the training dataset and the remaining one-third of the documents were used as the test dataset. We treated the training datasets as unlabeled set, $\mathcal{U}$, in Algorithm 1.

[1] The justification for our approach is similar for support vector machines. The idea is also similar for multinomial naïve Bayes with Dirichlet priors α_j. For a fixed Dirichlet prior with $\langle \alpha_1, \alpha_2, \cdots, \alpha_n \rangle$ setting, when $o < 1$ for a feature f_j, its counts are smoothed more.

[2] http://people.cs.umass.edu/mccallum/data.html.

[3] http://qwone.com/jason/20Newsgroups/.

Table 2 Description of the datasets: the domain, number of instances in training and test datasets, and size of vocabulary

Dataset	Task	Train	Test	Vocabulary
IMDB	Sentiment analysis of movie reviews	25,000	25,000	27,272
NOVA	Email classification (politics versus religion)	12,977	6,498	16,969
SRAA	Aviation versus auto document classification	48,812	24,406	31,883
WvsH	20Newsgroups (Windows vs. hardware)	1,176	783	4,026

We used the bag-of-words representation of documents with a dictionary of predefined vocabulary of phrases, consisting of only unigrams. To test whether our approach works across representations, we experimented with both binary and tf-idf representations for these text datasets. We evaluated our method using multinomial naïve Bayes, logistic regression, and support vector machines, as these are strong classifiers for text classification. We used the scikit-learn (Pedregosa et al. 2011) implementation of these classifiers with their default parameter settings for the experiments in this section.

To compare various strategies, we used learning curves. The initially labeled dataset was bootstrapped using 10 documents by picking 5 random documents from each class. A budget, B, of 200 documents was used in our experiments, because most of the learning curves flattened out after about 200 documents. We evaluated all the strategies using AUC (Area Under an ROC Curve) measure. The code to repeat our experiments is available on Github (http://www.cs.iit.edu/~ml/code/).

While incorporating the rationales into learning, we set the weights for rationales and the remaining features of a document as 1 and 0.01 respectively (i.e., $r = 1$ and $o = 0.01$). That is, we did not overemphasize the features corresponding to rationales but rather de-emphasized the remaining features in the document. These weights worked reasonably well for all four datasets, across all three classifiers, and using both binary and tf-idf data representations.

Obviously, these are not necessarily the best weight settings one can achieve; the optimal settings for r and o depend on many factors, such as the extent of the knowledge of the labeler (i.e., how many words a labeler can recognize), how noisy the labeler is, and how much labeled data there is in the training set. A more practical approach is to tune these parameters (e.g., using cross-validation) at each step of the learning curve. For simplicity, in this section, we present results using fixed weights for r and o as 1 and 0.01 respectively. Later, in Sect. 4, we present results by tuning the weights r and o using cross-validation on labeled data.

3.2.2 Simulating the human expert

Like most literature on feature labeling, we constructed an artificial labeler to simulate a human labeler, to allow for large-scale experimentation on several datasets and parameter configurations. Every time a document is annotated, we asked the artificial labeler to mark a word as a rationale for the chosen label. We allowed the labeler to return any one, and not necessarily the top one, of the positive words as a rationale for a positive document and any one of the negative words as a rationale for a negative document. If the labeler did not recognize any of the words as positive (negative) in a positive (negative) document, we let the labeler return null (ϕ) as the rationale.

‘great’, ‘excellent’, ‘wonderful’, ‘perfect’, ‘best’, ‘amazing’, ‘beautiful’, ‘love’, ‘favorite’, ‘loved’, ‘superb’, ‘brilliant’, ‘highly’, ‘fantastic’, ‘today’, ‘performance’, ‘beautifully’, ‘also’, ‘always’, ‘both’, ‘heart’, ‘performances’, ‘touching’, ‘wonderfully’, ‘enjoyed’, ‘well’
‘worst’, ‘bad’ , ‘waste’, ‘awful’, ‘terrible’, ‘stupid’, ‘worse’, ‘boring’, ‘horrible’, ‘poor’, ‘nothing’, ‘crap’, ‘minutes’, ‘supposed’, ‘poorly’, ‘no’, ‘lame’, ‘ridiculous’, ‘plot’, ‘script’, ‘avoid’, ‘dull’, ‘mess’

Fig. 1 Words selected as rationales for positive movie reviews (top) and negative movie reviews (bottom) for IMDB dataset

To make this as practical as possible in a real-world setting, we constructed the artificial labeler to recognize only the most apparent words in the documents. For generating rationales, we chose only the positive (negative) features that had the highest χ^2 (Chi-squared) statistic in at least 5% of the positive (negative) documents. This resulted in an overly-conservative labeler that recognized only a tiny subset of the words as rationales. For example, the artificial labeler knew about only 49 words out of 27272 words for IMDB, 111 words out of 16969 words for NOVA, 67 words out of 31883 words for SRAA, and 95 words out of 4026 words for WvsH dataset.

To determine whether the rationales selected by this artificial labeler are meaningful, we printed the actual words returned as rationales for IMDB dataset in Fig. 1, and verified that a majority of these words are human-recognizable words that could be naturally provided as rationales for classification. For example, the positive terms for the IMDB dataset included “great”, “excellent”, and “wonderful” and the negative terms included “worst”, “bad”, and “waste”. As Fig. 1 shows, the rationales returned by the artificial labeler are unigrams.

3.2.3 Results

Figure 2 presents the learning curves comparing LwR to Lw/oR on four document classification datasets with binary and tf-idf representations and using multinomial naïve Bayes, logistic regression, and support vector machines. We made sure that both Lw/oR and LwR work with the same set of documents, and the only difference between them is that in Lw/oR, the labeler provides only a label, whereas in LwR, the labeler provides both a label and a rationale. Hence, the difference between the learning curves of Lw/oR and LwR stems not from choosing different documents but rather from incorporating rationales into learning. Figure 2 shows that even though the artificial labeler knew about only a tiny subset of the vocabulary, and returned any *one* word, rather than the top word or all the words, as a rationale, LwR drastically outperformed Lw/oR across all datasets, classifiers, and representations. These results show that our method for incorporating rationales into the learning process is quite effective.

LwR provides improvements over Lw/oR, especially at the beginning of learning, when the labeled data is limited. LwR improves learning by enabling the classifier to quickly identify important feature-class correlations using the rationales provided by labeler. When the labeled data is large, Lw/oR can surpass LwR when $r >> o$. Ideally, one should have $r >> o$ when the labeled data is small and r should be closer to o when the labeled data is large. A more practical approach would be to tune these parameters (e.g., using cross-validation, as we later present in Sect. 4.2.2) at each iteration of learning. We empirically

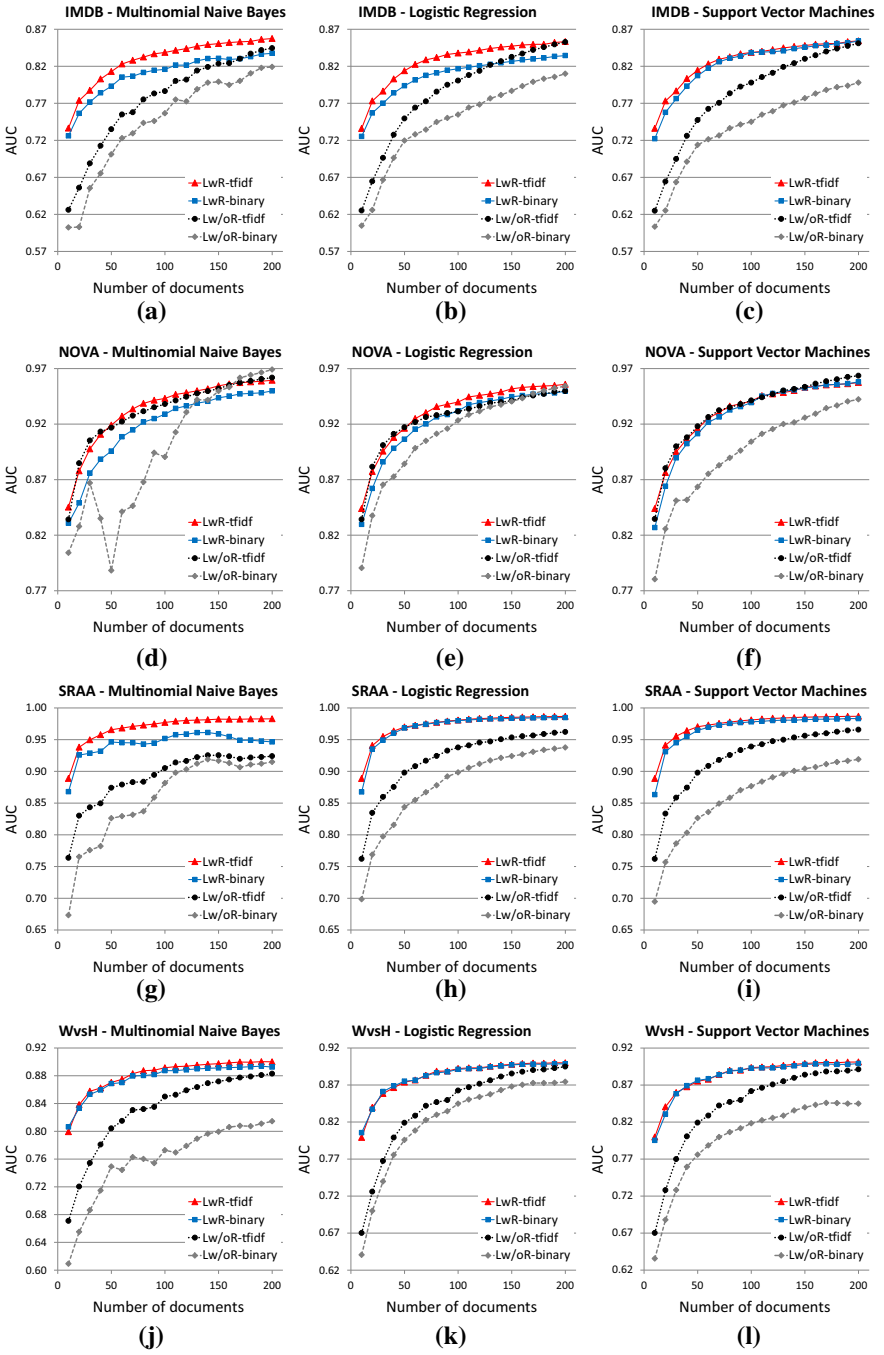

Fig. 2 Comparison between LwR and Lw/oR using multinomial naïve Bayes, logistic regression, and support vector machines on four datasets: IMDB (**a**–**c**), NOVA (**d**–**f**), SRAA (**g**–**i**), and WvsH (**j**–**l**). LwR provides drastic improvements over Lw/oR for all datasets with binary and tf-idf representations and using all three classifiers

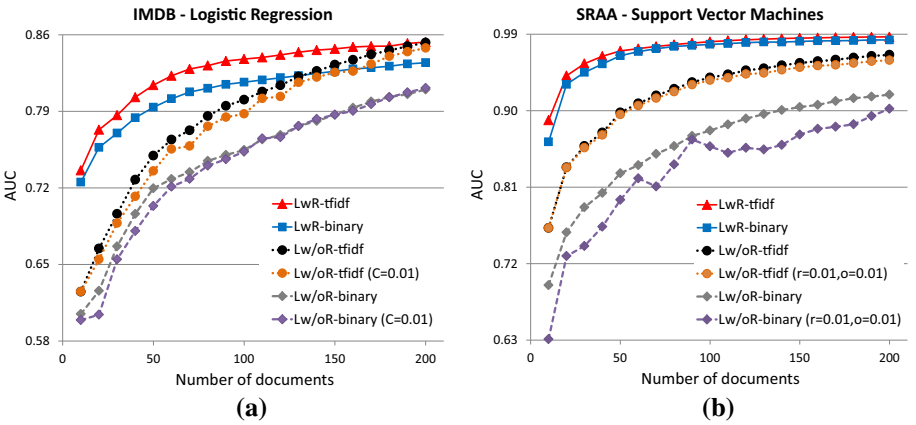

Fig. 3 **a** Results showing the effect of setting $C = 0.01$ for Lw/oR using binary and tf-idf representations. **b** Results showing the effect of multiplying the weights for all features by 0.01, i.e. setting $r = 0.01$ and $o = 0.01$. Using a higher regularization, $C = 0.01$, for Lw/oR or indiscriminately multiplying the weights of all features by 0.01 does not provide improvement over Lw/oR

found that most settings where $r > o$ in LwR approach performed better than Lw/oR. In this section, for simplicity, we set $r = 1$ and $o = 0.01$.

As discussed in Sect. 3.2.1, we used the default complexity parameters for logistic regression and support vector machines and used Laplace smoothing for multinomial naïve Bayes. Since most features are expected to be non-rationales, in Eq. 3, most features will appear in the second summation term, with $o = 0.01$. We tested whether the improvements that LwR provides over Lw/oR are simply due to implicit higher regularization for most of the features with $o = 0.01$, and hence experimented with Eq. 2 (which is Lw/oR) using $C = 0.01$. We observed that setting $C = 0.01$ and indiscriminately regularizing all the terms did not improve Lw/oR on most datasets and classifiers using both binary and tf-idf representations, providing experimental evidence that the improvements provided by LwR are not due to just higher regularization, but they are due to a more fine-grained regularization, as explained in Sect. 3.1. We present one such result for IMDB dataset using logistic regression in Fig. 3a.

Similarly, since most features in LwR representation had a weight of 0.01, and only a handful of features had a weight of 1, we repeated all the experiments using $r = 0.01$ and $o = 0.01$ to test whether indiscriminately decreasing the weights for all the terms in all the documents provides any improvement in Lw/oR. One would not expect that decreasing the weights for all the terms in all the documents would provide any improvement in learning, however, the LwR representation with $r = 1$ and $o = 0.01$ is quite similar to the representation where $r = 0.01$ and $o = 0.01$, because all the words, except the rationale word, in a document have a weight of 0.01. As expected, we found that for all datasets and classifiers and using both binary and tf-idf representations, indiscriminately multiplying all the terms by 0.01, i.e. setting $r = 0.01$ and $o = 0.01$, did not improve Lw/oR, providing further experimental evidence that the improvements provided by LwR over Lw/oR are not just due to placing smaller weights on all the terms. We present one such result for SRAA dataset using support vector machines in Fig. 3b.

Even though LwR improves performance drastically over Lw/oR, providing both a label and a rationale is expected to take more time of the labeler than simply providing a label. The question then is how to best utilize the labeler's time and effort: is it better to ask for only the labels of documents or should we elicit rationales along with the labels? To test how

Table 3 Comparison of number of documents needed to be annotated to achieve various target AUC performances using Lw/oR and LwR with multinomial naïve Bayes using binary representation. 'N/A' represents that a target AUC cannot be achieved by the learning strategy

Dataset	Target AUC	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
IMDB	Lw/oR-binary	23	63	79	102	152	339	N/A	N/A
	LwR-binary	2	5	11	22	62	257	N/A	N/A
	Ratio	11.5	12.6	7.2	4.6	2.5	1.3	N/A	N/A
NOVA	Lw/oR-binary	2	5	98	134	160	201	304	584
	LwR-binary	2	2	5	6	11	24	51	N/A
	Ratio	1	2.5	19.6	22.3	14.5	8.4	5.9	N/A
SRAA	Lw/oR-binary	6	9	25	76	100	188	294	723
	LwR-binary	2	2	3	5	7	9	20	N/A
	Ratio	3	4.5	8.3	15.2	14.3	20.9	14.7	N/A
WvsH	Lw/oR-binary	6	17	28	38	139	693	N/A	N/A
	LwR-binary	2	3	4	6	12	32	200	N/A
	Ratio	3	5.7	7	6.3	11.6	21.7	N/A	N/A

much a document annotated with a label and a rationale is worth, we computed how many documents a labeler would need to inspect to achieve a target AUC performance, using Lw/oR and LwR. Tables 3 and 4 present the number of documents required to achieve a target AUC using Lw/oR and LwR for multimonial naïve Bayes using binary and tf-idf representations.

Tables 3 and 4 show that LwR drastically accelerates learning compared to Lw/oR, and it requires relatively very few annotated documents for LwR to achieve the same target AUC as Lw/oR. For example, in order to achieve a target AUC of 0.95 for SRAA dataset (using tf-idf representation with MNB classifier), Lw/oR required labeling 656 documents, whereas LwR required annotating a mere 29 documents. That is, if the labeler is spending a minute per document to simply provide a label, then it is better to provide a label *and* a rationale as long as providing both a label and a rationale does not take more than $656/29 \approx 22$ minutes of labeler's time. The results for logistic regression and support vector machines using both binary and tf-idf representations are similar, and hence they are omitted to avoid redundancy.

Zaidan et al. (2007) conducted user studies and showed that providing 5 to 11 rationales and a class label per document takes roughly twice the time of providing only the label for documents. In our experiments, the labeler was asked to provide *any* one rationale instead of all the rationales. Hence, even though we do not know for sure whether labelers would take more/less time in providing one rationale as opposed to all the rationales, Tables 3 and 4 show that documents annotated with rationales are often worth at least as two and sometimes more than even 20 documents that are simply annotated with labels.

3.2.4 Results with user-annotated rationales

We evaluated our approach on user-annotated IMDB dataset provided by Zaidan et al. (2008). The dataset consists of 1800 IMDB movie reviews for which a user provided rationales for labeled documents. The main difference between the simulated expert and the user-annotated dataset is that the simulated expert selected only one word as a rationale, whereas the human highlighted many words, and sometimes even phrases, as rationales. Simulated rationales

Table 4 Comparison of number of documents needed to be annotated to achieve various target AUC performances using Lw/oR and LwR with multinomial naïve Bayes using tf-idf representation. 'N/A' represents that a target AUC cannot be achieved by the learning strategy

Dataset	Target AUC	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
IMDB	Lw/oR-tfidf	7	14	37	65	106	233	841	N/A
	LwR-tfidf	2	4	10	16	37	164	N/A	N/A
	Ratio	3.5	3.5	3.7	4.1	2.9	1.4	N/A	N/A
NOVA	Lw/oR-tfidf	2	2	3	3	5	12	28	126
	LwR-tfidf	2	2	2	3	4	11	31	110
	Ratio	1	1	1.5	1	1.2	1.1	0.9	1.1
SRAA	Lw/oR-tfidf	2	4	7	12	21	58	109	656
	LwR-tfidf	2	2	3	4	6	8	13	29
	Ratio	1	2	2.3	3	3.5	7.3	8.4	22.6
WvsH	Lw/oR-tfidf	5	9	17	33	57	127	380	N/A
	LwR-tfidf	2	3	4	6	12	33	188	N/A
	Ratio	2.5	3	4.3	5.5	4.8	3.8	2	N/A

can also be noisy; in our study, the simulated labeler returns *any* one word as a rationale, but in real life, it might not be the rationale.

We performed 5-fold cross validation and repeated each experiment 5 times for each fold and present average results. We used tf-idf representation of the dataset. Figure 4 presents the results on user-annotated IMDB dataset comparing LwR to Lw/oR using multinomial naïve Bayes, logistic regression, and support vector machines. We found that LwR performed better than Lw/oR using the default weight settings ($r = 1$ and $o = 0.01$). However, user-annotated rationales can be really noisy, where users do not necessarily pinpoint just the important words, but rather highlight phrases (or even sentences) that span several words. When the expert is noisy, the trust in the expert should be reflected in the weights r and o. If the user is trustworthy and precise in pin-pointing the rationales, then r should be much greater than o, but if the user is noisy, then r should be relatively closer to o.

To test the effect of weights r and o on noisy rationales, we experimented with various settings for r and o between 0.001 and 1000. For user-annotated IMDB dataset, we found that weight settings where r was closer to o worked better than weight settings where r was much greater than o. In general, the default setting of r=1 and o=0.01 worked well for the simulated labeler case and the setting $r = 1$ and $o = 0.1$ worked well for the user-annotated case.

4 Comparison with baselines

In this section, we empirically compare our approach to incorporate rationales with other classifier-specific approaches from the literature. Our experiments were based on three classifiers: multinomial naïve Bayes, logistic regression, and support vector machines. Hence, we looked for classifier-specific approaches in the literature that focused on these three classifiers.

When the underlying classifier is support vector machines, the closest work to ours is that of Zaidan et al. (2007), in which they incorporated rationales into the training of support vector machines, so we chose this as a baseline for our approach using support vector machines. When the underlying classifier is multinomial naïve Bayes, we are not aware of any approach

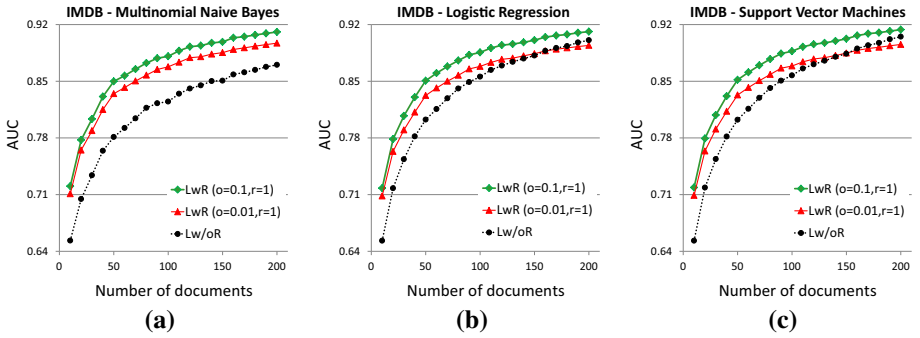

Fig. 4 Comparison of LwR to Lw/oR on user-annotated IMDB dataset with tf-idf representation using **a** multinomial naïve Bayes, **b** logistic regression, and **c** support vector machines. LwR with default weight setting of $r = 1$ and $o = 0.01$ provides improvements over Lw/oR using all three classifiers. Since user-annotated rationales can be rather noisy, LwR with weights $r = 1$ and $o = 0.1$ performs better than LwR with weights $r = 1$ and $o = 0.01$

specifically developed to incorporate rationales into learning. The closest work to learning with rationales is feature annotation (e.g., Melville and Sindhwani 2009; Raghavan and Allan 2007; Stumpf et al. 2009), in which labelers annotate features independent of the documents. Even though learning with rationales is not the same as feature annotation, learning with rationales can be treated as feature annotation if the underlying rationales correspond to features. Melville and Sindhwani (2009) presented pooling multinomials to incorporate feature annotations into the training of multinomial naïve Bayes, hence we chose this as a baseline for our approach using multinomial naïve Bayes. We are not aware of any approach specifically developed to incorporate rationales into the training of logistic regression classifier, and the closest work is that of Das et al. (2013), which was specifically designed to incorporate feature annotation into the training of locally-weighted logistic regression, and hence we chose it as a baseline for our approach using logistic regression.

4.1 Description of the baselines

4.1.1 Description of Zaidan et al. (2007)

Zaidan et al. (2007) presented a method to incorporate rationales into the training of support vector machines. They asked labelers to highlight the most important words and phrases as rationales to justify why a movie review is labeled as positive or negative. For each document, x^i, annotated with a label and one or more rationales, one or more contrast examples, v^{ij} (where j is the number of rationales for document x^i), is created that resembles x^i, but lacks the evidence (rationale) that the annotator found significant, and new examples $x^{ij} \stackrel{\text{def}}{=} \frac{x^i - v^{ij}}{\mu}$ along with their class labels, $\langle x^{ij}, y^i \rangle$, are added to the training set, where μ controls the desired margin between the original and contrast examples. The soft-margin SVM chooses w and ξ_i to minimize:

$$\min_w \frac{1}{2} \|w\|^2 + C \left(\sum_{i=1}^{n} \xi_i \right) \tag{4}$$

subject to the constraints:

$$(\forall i) w \cdot x^i \cdot y^i \geq 1 - \xi_i \tag{5}$$

$$(\forall i) \xi_i \geq 0 \tag{6}$$

where x^i is a training document, $y^i \in \{-1, +1\}$ is the class, and ξ_i is the slack variable. The parameter $C > 0$ controls the relative importance of minimizing w and cost of the slack. In their approach, they add the contrast constraints:

$$(\forall i, j) w \cdot (x^i - v^{ij}) \cdot y^i \geq \mu(1 - \xi_{ij}) \tag{7}$$

where $\xi_{ij} > 0$ is the associated slack variable. The contrast constraints have their own margin, μ, and the slack variables have their own cost, so their objective function for support vector machines becomes:

$$\min_w \frac{1}{2} \|w\|^2 + C \left(\sum_{i=1}^{n} \xi_i \right) + C_{contrast} \left(\sum_{i,j} \xi_{ij} \right) \tag{8}$$

In Zaidan et al. (2007), for each document, one contrast example, v^{ij}, and several pseudoexamples, x^{ij}, for the rationales are created. Hence, according to Eq. 8, the hyperplane is determined by whether the contrast examples or the pseudoexamples add to the loss function or participate in the optimization as a support vector. Analytically, our approach is equivalent to Zaidan et al. (2007) when all of the following three conditions hold: (i) $C = C_{contrast}$, (ii) $o = 1$ and $r = \frac{1}{\mu}$, and (iii) in our approach, if a document x^i becomes a support vector, then in Zaidan et al. (2007) approach, both the contrast example, v^{ij}, and pseudoexamples, x^{ij}, for the document x^i also become support vectors.

4.1.2 Description of Melville and Sindhwani (2009)

Melville and Sindhwani (2009) presented an approach to incorporate feature labels and instance labels into the training of a multinomial naïve Bayes classifier. They build two multinomial naïve Bayes models: one trained on labeled instances and the other trained on labeled features. The two models are then combined using linear pooling (Melville et al. 2009) to aggregate the conditional probabilities, $P(f_j|y_k)$ using:

$$P(f_j|y_k) = \beta P_e(f_j|y_k) + (1 - \beta) P_f(f_j|y_k) \tag{9}$$

where y_k is the class, $P_e(f_j|y_k)$ and $P_f(f_j|y_k)$ represent the probabilities assigned by the model trained on labeled instances and the model trained on labeled features respectively, and β is the weight for combining these two conditional probability distributions.

In order to build a model trained on labeled features, Melville et al. (2009) assumed that a positive term, $f+$, is more likely to appear in a positive document than in a negative document and a negative term, $f-$, is more likely to appear in a negative document than in a positive document. To build a model trained on labeled features, they specified a parameter for polarity level, γ, to measure the likeliness of positive (negative) term to occur in a positive (negative) document compared to a negative (positive) document. Equation 10 computes the conditional probabilities of the unknown terms, f_u, given class labels, '+' and '−'.

$$\begin{aligned} P(f_u|+) &= \frac{n(1 - 1/\gamma)}{(p+n)(m-p-n)}, \quad \text{and} \\ P(f_u|-) &= \frac{n(1 - 1/\gamma)}{(p+n)(m-p-n)} \end{aligned} \tag{10}$$

where $P(f_u|+)$ and $P(f_u|-)$ are the conditional probabilities of the unknown terms given class, m is the number of terms in the dictionary, p is the number of positive terms labeled by the labeler, and n is the number of negative terms labeled by the labeler.

The main difference between our approach and Melville and Sindhwani (2009) is that in our approach, rationales are tied to the documents in which they appear as rationales, whereas in Melville and Sindhwani (2009), the feature labels are weighted globally, and all positive words are equally positive, and all negative words are equally negative. Our approach provides more granular (per instance-feature combination) regularization as described in Sect. 3.1. Hence, there is no parameter setting where our approach is equivalent to Melville and Sindhwani (2009), however, as we show in Sect. 4.2, empirically, our approach performs quite similar to Melville and Sindhwani (2009).

4.1.3 Description of Das et al. (2013)

Das et al. (2013) proposed an approach for incorporating feature labels into the training of a locally-weighted logistic regression classifier (Cleveland and Devlin 1988). In feature annotation, each feature (for example, the term) is labeled by the human. For example, for a binary sentiment classification task, the terms are labeled as positive or negative. Locally-weighted logistic regression fits one logistic function per test instance, where the objective function for the logistic regression model is modified so that the training instances that are closer to the test instance are given higher weights compared to the training instances that are farther away from the test instance. When computing similarity between the test instances and training instances, in addition to regular document similarity, Das et al. (2013) takes labeled features into account: when a test document shares labeled features with a training document, it computes similarity between the test document and the training document based on the labeled features and the label of the training instance.

Logistic regression maximizes the conditional log likelihood of data as:

$$l_w(\theta) = \sum_{i=1}^{N} \log\left(P_\theta(y^i|x^i)\right) \tag{11}$$

Locally-Weighted Logistic Regression (LWLR) fits a logistic function around a small neighborhood of test instance, x^t, where the training instances, x^i, that are closer to x^t are given higher weights compared to the training instances that are farther away from x^t. LWLR maximizes the conditional log likelihood of data as:

$$l_w(\theta) = \sum_{i=1}^{m} w(x^t, x^i) \log\left(P_\theta(y^i|x^i)\right) \tag{12}$$

where, the weight $w(x^t, x^i)$ is a kernel function:

$$w(x^t, x^i) = \exp\left(-\frac{f(x^t, x^i)^2}{k^2}\right) \tag{13}$$

where $f(x^t, x^i)$ is a distance function and k is the kernel width.

Das et al. (2013) used LWLR for its ability to weight training instances differently, rather than for its ability to learn a non-linear decision boundary. LWLR assigns higher weights to documents that are more similar to x^t, and lower weights to documents that are less similar to x^t. They used $\text{cosim}(x^t, x^i) = 1 - \cos(x^t, x^i)$ as the baseline distance function to measure similarity between documents. To incorporate feature labeling into LWLR, they

changed the baseline distance function to include two components: (i) distance between documents x^t and x^i based on all the words present in x^t and x^i, i.e. cosim(x^t, x^i) and (ii) distance between documents x^t and x^i based on all the features that have been labeled by user.

The second component of the distance function is computed as the difference between contributions of class-relevant and class-irrelevant features in x^t, where x^t is l_2-normalized tf-idf feature vector. Considering binary classification, $y \in \{+, -\}$, if the label of x^i is '+', the class-relevant features in x^t will be all the features that have been labeled as '+', and the class-irrelevant features in x^t will be all the features that have been labeled as '−'. Similarly, if the label of x^i is '−', the class-relevant features in x^t will be all the features that have been labeled as '−', and the class-irrelevant features in x^t will be all the features that have been labeled as '+'. Let $\mathcal{R}$ be a set of class-relevant features in x^t and let $\mathcal{I}$ be a set of class-irrelevant features in x^t. Their modified distance function for incorporating feature labels into LWLR becomes:

$$f(x^t, x^i) = \text{cosim}(x^t, x^i) \left(\sum_{j \in \mathcal{R}} x^t_j - \sum_{j \in \mathcal{I}} x^t_j \right) \tag{14}$$

Since the above distance function can sometimes become negative, the weight $w(x^t, x^i)$ is computed as:

$$w(x^t, x^i) = \exp\left(-\frac{\max(0,\, f(x^t, x^i))^2}{k^2} \right) \tag{15}$$

For simplicity, in Eq. 14, we present formulation of their approach for binary classification. We refer the reader to Das et al. (2013) for a general formulation of their approach for multi-class classification.

Next, we present the results to empirically compare our classifier-agnostic approach with the three classifier-specific approaches: Melville and Sindhwani (2009), Zaidan et al. (2007), and Das et al. (2013).

4.2 Results

In this section, we first describe the experimental settings used to compare our approach to three baselines, Zaidan et al. (2007), Melville and Sindhwani (2009), and Das et al. (2013), and then present the results for empirical comparison. Note that the results for our approach and the baselines depend on hyper-parameters used in the experiments, hence, in order to have a fair comparison between our approach and the baselines, we compared them under two settings. First, we compared them using the best possible hyper-parameter settings. We ran several experiments using a wide range of values for all hyper-parameters and report the best possible performance, measured as the highest area under the learning curve, for each method. This is essentially equivalent to tuning parameters using the test data itself. We performed this test to observe how different methods would behave at their best. Second, we compared them using hyper-parameters that were optimized at each iteration of learning using cross validation on the labeled set, $\mathcal{L}$ obtained including and up to that iteration of active learning. We also provide results for learning without rationales (Lw/oR) using best parameters and using hyper-parameters optimized using cross validation on labeled data.

We used the same four document classification datasets described in Sect. 3.2.1. Since the results in Sect. 3.2 showed that tf-idf representation gave better results than the binary representation, in this section, we present results using only the tf-idf representation of the

datasets. We repeated each experiment 10 times, starting with a different bootstrap, and report average results on 10 different trials.

Our method using multinomial naïve Bayes classifier (LwR-MNB) needs to tune the following hyper-parameters: (i) the Dirichlet prior, α, for the features (ii) weight for the rationale features, r, and (iii) weight for the other features, o. The method in Melville and Sindhwani (2009) needs to tune the following hyper-parameters: (i) smoothing parameter for the instance model, α, (ii) polarity level for the feature model, (γ), and (iii) weights for combining the instance model and feature model (β and $1-\beta$ respectively).

Our method using support vector machines (LwR-SVM) needs to tune the following parameters: (i) regularization parameter, C, (ii) weight for the rationale features, r, and (iii) weight for the other features, o. Zaidan et al. (2007) approach needs to tune the following hyper-parameters: (i) regularization parameter, C, for the pseudoexamples, x_{ij}, (ii) regularization parameter, $C_{contrast}$, for the contrast examples, v_{ij}, and (iii) margin between the original and contrast examples, μ.

Das et al. (2013) used locally-weighted logistic regression specifically to incorporate feature labels into learning. Our method to incorporate rationales is independent of the classifier, hence we compared our approach to Das et al. (2013) using both logistic regression and locally-weighted logistic regression to see whether the improvements provided by incorporating rationales stem from using locally-weighted logistic regression. Our method using locally-weighted logistic regression classifier (LwR-LWLR) needs to tune the following parameters: (i) regularization parameter, C, (ii) kernel width, k, (iii) weight for the rationale features, r, and (iv) weight for the other features, o. Our method using vanilla logistic regression classifier (LwR-LR) needs to tune the following parameters: (i) regularization parameter, C, (ii) weight for the rationale features, r, and (iii) weight for the other features, o. Das et al. (2013) approach needs to tune the following parameters: (i) regularization parameter, C and (ii) kernel width, k.

For each instance, x^t, in the test data, LWLR builds a model around a small neighborhood of x^t, based on distances between the test instance and training instances, x^i. This method requires learning a logistic function for each test instance, and is therefore computationally very expensive. In this study, we compare our approach to the baselines using best hyper-parameters, which requires repeating each experiment several times with all possible hyper-parameter combinations. Moreover, our cross validation experiments require tuning hyper-parameters at each step of learning. To reduce the running time of LWLR experiments, we reduced the test data by randomly subsampling 500 test instances. To further reduce the running time, we searched for one parameter at a time, fixing others; that is, we did not perform a joint search over all the hyper-parameters for LWLR experiments.

4.2.1 Comparison to baselines under best parameter settings

In this section, we present results comparing the best learning curves obtained using our approach and the baselines. We bootstrapped the initial model using 10 instances chosen randomly, picking 5 documents from each class. At each iteration of learning, we selected 10 documents randomly from the unlabeled pool, $\mathcal{U}$. We repeated the experiments using a wide range of hyper-parameters for our approach and the baselines and plotted the best learning curve for each method.

For our approach using multinomial naïve Bayes, we searched for α between 10^{-6} and 10^2. For our approach using support vector machines, we searched for C between 10^{-2} and 10^2. For our approach using locally-weighted logistic regression, we searched for C between 10^{-3} and 10^3 and k between 0.1 and 1. For our approach using multinomial naïve Bayes

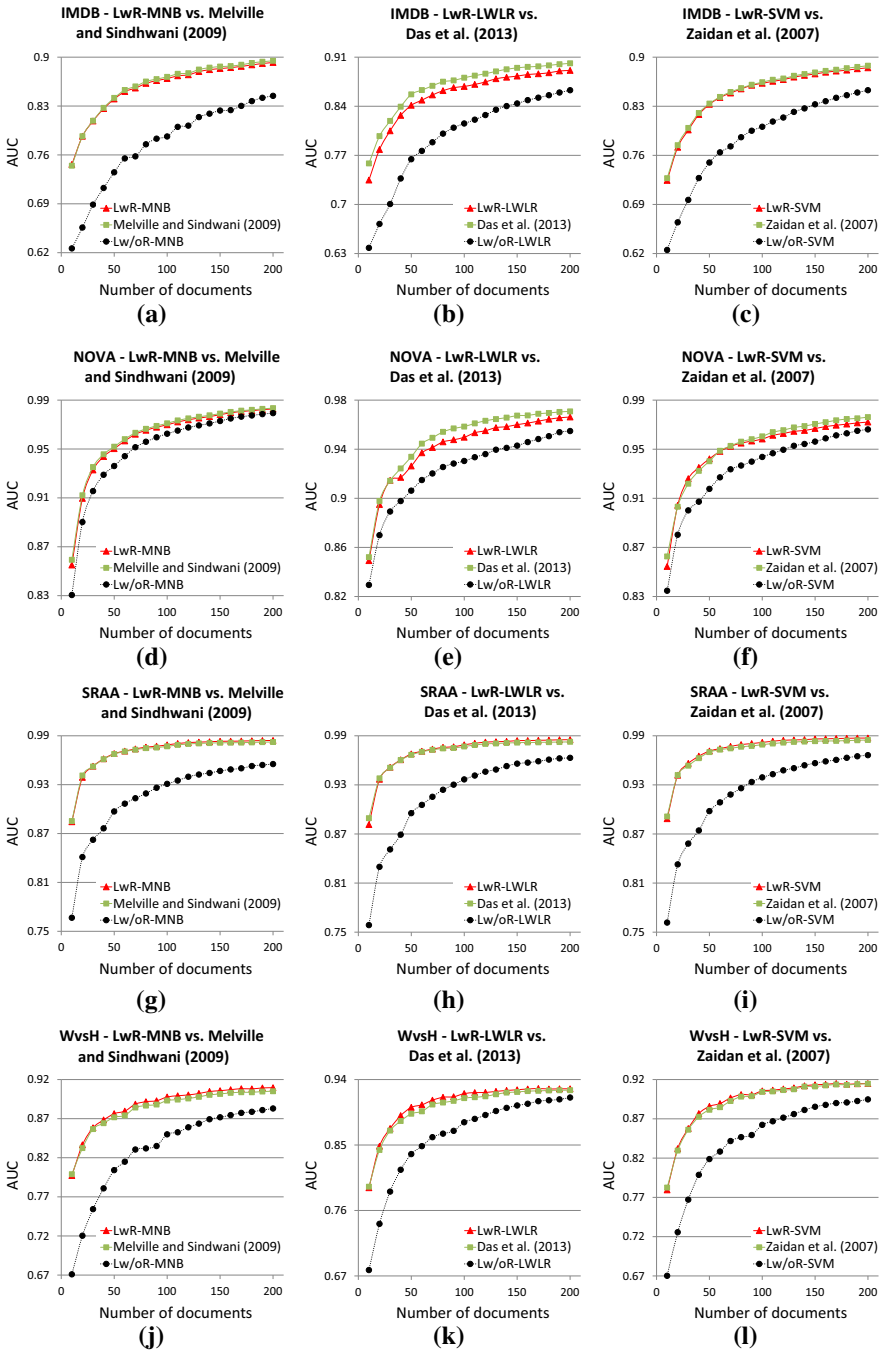

Fig. 5 Results comparing our approach to the three baselines using best hyper-parameters. LwR-MNB performs similar to Melville and Sindhwani (2009) on all four datasets (**a**, **d**, **g**, and **j**). LwR-LWLR performs similar to Das et al. (2013) on all four datasets (**b**, **e**, **h**, and **k**). LwR-SVM performs similar to Zaidan et al. (2007) on all four datasets (**c**, **f**, **i**, and **l**)

Table 5 Hyper-parameter settings for Lw/oR-SVM, LwR-SVM, and Zaidan et al. (2007) that gave the best learning curves

Dataset	Lw/oR-SVM	LwR-SVM			Zaidan et al. (2007)		
	C	C	r	o	C	$C_{contrast}$	μ
IMDB	0.1	0.1	10	1	0.5	0.5	0.1
NOVA	10	0.1	10	1	1	1	0.1
SRAA	10	10	1	0.01	0.1	10	0.1
WvsH	0.1	10	1	0.1	0.2	0.2	0.1

and support vector machines, we searched for weights r and o between 10^{-4} and 10^{7}. For our approach using locally-weighted logistic regression, we searched for weights r and o between 10^{-3} and 10^{3}. In Zaidan et al. (2007), for C and $C_{contrast}$, we searched for values between 10^{-3} and 10^{3}, and μ between 10^{-2} and 10^{2}. In Melville and Sindhwani (2009), we searched for α between 10^{-6} and 10^{2}, γ between 1 and 10^{5}, and β between 0 and 1. In Das et al. (2013), we searched for C between 10^{-3} and 10^{3} and k between 0.1 and 1.

Figure 5 presents the learning curves comparing LwR-SVM to Zaidan et al. (2007), LwR-MNB to Melville and Sindhwani (2009), and LwR-LWLR to Das et al. (2013). These results show that under best parameter settings, our classifier-agnostic approach performs as good as other classifier-specific approaches. The results for our approach using logistic regression and locally-weighted logistic regression are very similar under best parameter settings, however, LWLR is computationally very expensive. We omit the learning curves for LwR-LR in Fig. 5, as it is very similar to LwR-LWLR.

We report the hyper-parameter values that gave us the best possible learning curves (learning curves with the highest area under the AUC curve) for our approach and the baselines in Tables 5, 6, and 7. For our approaches, LwR-SVM, LwR-MNB, and LwR-LWLR, as expected, $r > o$ gave the best results. For Zaidan et al. (2007), we found that $\mu = 0.1$ and setting $C <= C_{contrast}$ gave the best results. Melville and Sindhwani (2009) used the weights for combining the instance model (β) and feature model ($1 - \beta$) as 0.5 and 0.5 respectively. However, we found that for the four text datasets we used in this study, placing a much higher weight (e.g. 0.9 or 0.99) on the instance model gave better results than using their default weights for combining the two models. Note that if we place a weight of 1 for the instance model (i.e. $\beta = 1$), the weight for the feature model will be zero, and this will give the same results as Lw/oR-MNB. Das et al. (2013) reported that setting $k = \sqrt{0.5}$ for LWLR-FL gave reasonable good macro-F1 scores, however, for the four text datasets, we found that $k > 0.4$ gave good results for AUC measure.

4.2.2 Comparison to baselines by tuning parameters using cross validation

In this section, we present the results comparing our approach with the baselines under the setting where we search for optimal hyper-parameters using cross validation on labeled data, $\mathcal{L}$, at each iteration of learning. We performed 5 fold cross validation on $\mathcal{L}$ and optimized all the hyper-parameters for the AUC measure, since AUC is the target performance measure in our experiments.

AUC of a classifier is equivalent to the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance. In an active learning setting, the labeled data ($\mathcal{L}$) is severely limited, consisting of only a few instances. When we use 5 fold cross validation, each fold containing only 20% of the instances is evaluated to produce an AUC score, which does not give an accurate measure of ranking. Hence, in order

Table 6 Hyper-parameter settings for Lw/oR-MNB, LwR-MNB, and Melville and Sindhwani (2009) that gave the best learning curves

Dataset	Lw/oR-MNB	LwR-MNB			Melville and Sindhwani (2009)		
	α	α	r	o	α	γ	β
IMDB	1	1	100	1	1	100,000	0.99
NOVA	0.1	1	250	10	0.1	100,000	0.9
SRAA	0.01	1	125	0.1	10	100,000	0.99
WvsH	1	1	75	1	0.9	100,000	0.9

Table 7 Hyper-parameter settings for Lw/oR-LWLR, LwR-LWLR, and Das et al. (2013) that gave the best learning curves

Dataset	Lw/oR-LWLR		LwR-LWLR				Das et al. (2013)	
	C	k	C	k	r	o	C	k
IMDB	1	0.7	1	0.7	10	1	1000	1
NOVA	1000	1	1000	1	1	0.1	1000	1
SRAA	1000	1	1000	1	1	0.01	100	0.4
WvsH	10	0.5	10	0.5	1	0.1	1000	1

to fully utilize the scores assigned by the classifier to instances in all the folds, we merge-sorted the instances in all the folds using their assigned scores, and computed AUC score based on instances in all the folds. This is similar to the approach described in Fawcett (2006).

Figure 6 presents the learning curves comparing LwR-SVM to Zaidan et al. (2007), LwR-MNB to Melville and Sindhwani (2009), and LwR-LWLR to Das et al. (2013). As these results show, when we optimize the hyper-parameters using cross validation on training data, LwR-SVM performs very similar to Zaidan et al. (2007), LwR-MNB performs very similar to Melville and Sindhwani (2009), and LwR-LWLR performs very similar to Das et al. (2013). We performed t tests comparing the learning curves obtained using our method and the baselines and found that the differences are not statistically significant in most cases.

The results for our approach using logistic regression (LwR-LR) and using locally-weighted logistic regression (LwR-LWLR) have some differences, when the hyper-parameters are optimized using cross validation on labeled set. For experiments using LWLR, we did not perform a grid search for the parameters, and optimized only one parameter at a time, which could result in sub-optimal hyper-parameters. Moreover, our approach using LWLR needs to tune four hyper-parameters (C, k, r, and o) and Das et al. (2013) needs to tune two hyper-parameters (C and k).

These results show that our approach to incorporate rationales is as effective as three other approaches from the literature, Zaidan et al. (2007), Melville and Sindhwani (2009), and Das et al. (2013), that were designed specifically for incorporating rationales and feature annotations into support vector machines, multinomial naïve Bayes, and locally-weighted logistic regression respectively. Our approach has the additional benefit of being independent of the underlying classifier.

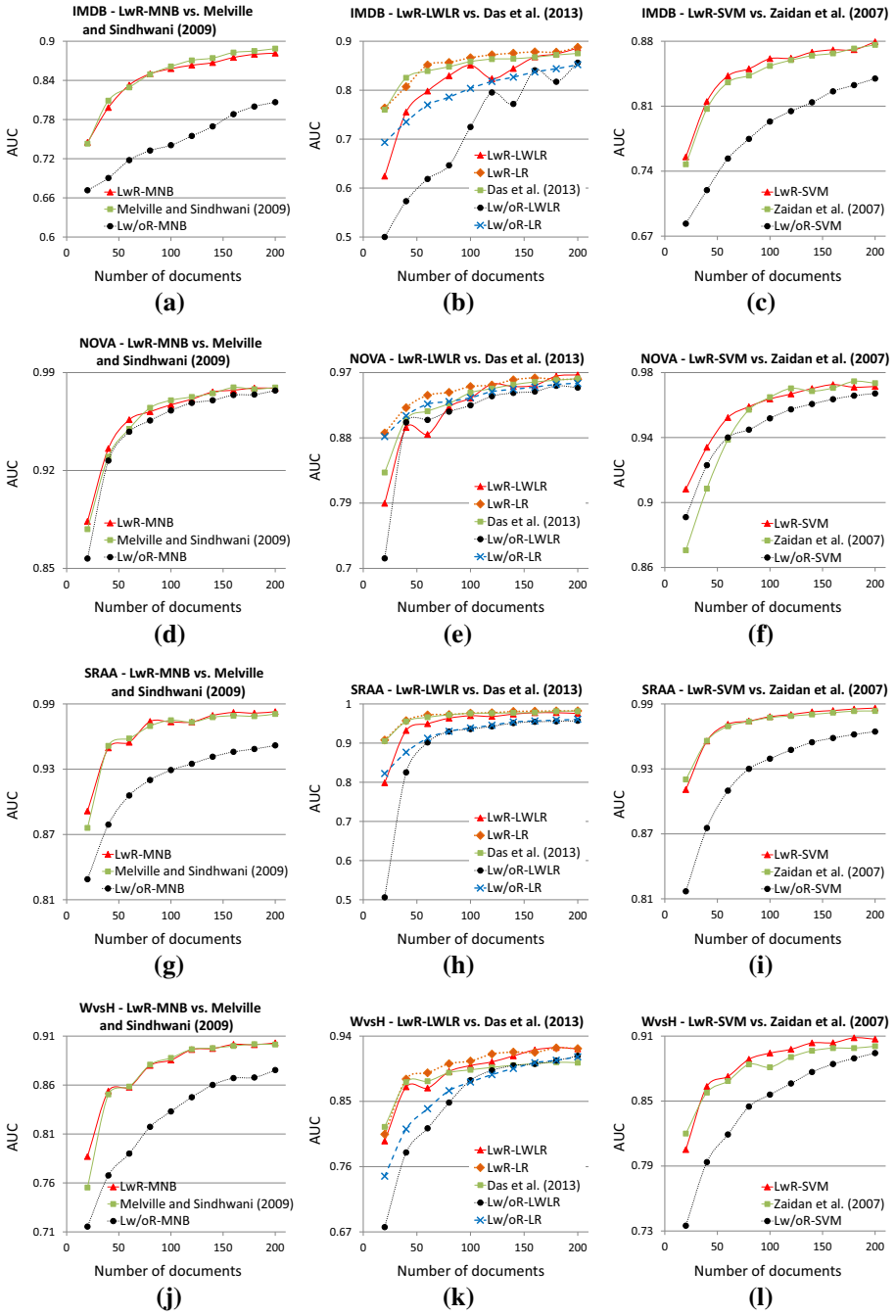

Fig. 6 Results comparing our approach to the three baselines with hyper-parameters tuned using cross-validation on labeled data. LwR-MNB performs similar to Melville and Sindhwani (2009) on all four datasets (**a**, **d**, **g**, and **j**). LwR-LWLR performs similar to Das et al. (2013) on all four datasets (**b**, **e**, **h**, and **k**). LwR-SVM performs similar to Zaidan et al. (2007) on all four datasets (**c**, **f**, **i**, and **l**)

5 Active learning with rationales

So far we have seen that LwR provides drastic improvements over Lw/oR and our approach performs as well as other classifier-specific approaches in the literature. In previous sections, we made sure that both LwR and Lw/oR saw the same documents and we chose those documents randomly from the unlabeled set of documents. Active learning (Settles 2012) aims to carefully choose instances for labeling to improve over random sampling. Many successful active learning approaches have been developed for annotating instances (Lewis and Gale 1994; Seung et al. 1992; Roy and McCallum 2001). Ramirez-Loaiza et al. (2016) provide an empirical evaluation of common active learning strategies. Several approaches have been developed for annotating features (Druck et al. 2009; Das et al. 2013) and rotating between annotating instances and annotating features (Raghavan and Allan 2007; Druck et al. 2009; Attenberg et al. 2010; Melville and Sindhwani 2009). In this section, we introduce an active learning strategy that is specifically tailored for the learning with rationales framework.

5.1 Active learning to select documents based on rationales

Arguably, one of the most successful active learning strategies for text categorization is uncertainty sampling, which was first introduced by Lewis and Catlett (1994) for probabilistic classifiers and later formalized for support vector machines by Tong and Koller (2001). The idea is to label instances for which the underlying classifier is uncertain, i.e., the instances that are close to the decision boundary of the model. It has been successfully applied to text classification tasks in numerous publications, including Zhu and Hovy (2007), Sindhwani et al. (2009), and Segal et al. (2006).

We adapt uncertainty sampling for the learning with rationales framework. To put simply, when the underlying model is uncertain about an unlabeled document, we examine whether the unlabeled document contains words/phrases that were returned as rationales for any of the existing labeled documents. More formally, let R^+ denote the union of all the rationales returned for the positive documents so far. Similarly, let R^- denote the union of all the rationales returned for the negative documents so far. An unlabeled document can be one of these three types:

- Category 1: has no words in common with R^+ and R^-.
- Category 2: has word(s) in common with either R^+ or R^- but not both.
- Category 3: has at least one word in common with R^+ and at least one word in common with R^-.

One would imagine that annotating each of the Category 1, Category 2, and Category 3 documents has its own advantage. Annotating Category 1 documents has the potential to elicit new domain knowledge, i.e., terms that were not provided as a rationale for any of the existing labeled documents. It also carries the risk of containing little to no useful information for the classifier (e.g., a neutral review). For Category 2 documents, even though the document shares a word that was returned as a rationale for another document, the classifier is still uncertain about the document either because that word is not weighted high enough by the classifier and/or there are other words that pull the classification decision in the other direction, making the classifier uncertain. Category 3 documents contain conflicting words/phrases and are potentially harder cases, and annotating Category 3 documents has the potential to resolve conflicts for the classifier.

Building on our previous work on uncertainty sampling (Sharma and Bilgic 2013), we devised an active learning approach, where given uncertain documents, the active learner

prefers documents of Category 3 over Categories 1 and 2. We call this strategy as *uncertain-prefer-conflict* (UNC-PC) because Category 3 documents carry conflicting words (with respect to rationales) whereas Category 1 and Category 2 documents do not. The difference between this approach and our previous work (Sharma and Bilgic 2013) is that in Sharma and Bilgic (2013), we selected uncertain instances based on model's perceived conflict whereas in this work, we are selecting documents based on conflict caused by the domain knowledge provided by the labeler. Next, we compare the vanilla uncertainty sampling (UNC) and UNC-PC strategies using LwR to see if using uncertain Category 3 documents could improve active learning.

5.2 Active learning with rationales experiments

We used the same four text datasets and evaluated our method UNC-PC using multinomial naïve Bayes, logistic regression, and support vector machines. For the active learning strategies, we used a bootstrap of 10 random documents, and labeled five documents at each round of active learning. We used a budget of 200 documents for all methods. UNC simply picks the top five uncertain documents, whereas UNC-PC looks at top 20 uncertain documents and picks five uncertain documents giving preference to the conflicting cases (Category 3) over the non-conflicting cases (Category 1 and Category 2). We repeated each experiment 10 times starting with a different bootstrap at each trial and report the average results.

Figure 7 presents the learning curves comparing UNC-PC with UNC for multinomial naïve Bayes. Since the performances of both LwR and Lw/oR using tf-idf representation are better than the performance using binary representation, we compared UNC-PC to UNC for LwR using only the tf-idf representation. We see that for multinomial naïve Bayes, UNC-PC improves over traditional uncertainty sampling, UNC, on two datasets, and hurts performance on one dataset. The trends are similar for other classifiers and hence we omit them for simplicity.

We performed paired t tests to compare the learning curves of UNC-PC with the learning curves of UNC, to test whether the average of one learning curve is significantly better or worse than the average of the other learning curve. If UNC-PC has a higher average AUC than UNC with a t test significance level of 0.05 or better, it is a Win, if it has significantly lower performance, it is a Loss, and if the difference is not statistically significant, the result is a Tie.

Table 8 shows the datasets for which UNC-PC wins, ties, or loses compared to UNC. The t test results show that UNC-PC wins on two out of four datasets for MNB and LR, and wins on three datasets for SVM. However, as these results and Fig. 7 show, even though UNC-PC has potential, it is far from perfect, leaving room for improvement.

6 Future work

An exciting future research direction is to allow the labelers to provide richer feedback. This is especially useful for resolving conflicts that stem from seemingly conflicting words and phrases. For example, for the movie review "The plot was great, but the performance of the actors was terrible. Avoid it." the positive word "great" is at odds with the negative words "terrible" and "avoid". If the labeler is allowed to provide richer feedback, stating that the word "great" refers to the plot, "terrible" refers to the performance, and "avoid" refers to the movie, then the learner might be able to learn to resolve similar conflicts in other documents. However, this requires a conflict resolution mechanism in which the labeler can provide rich

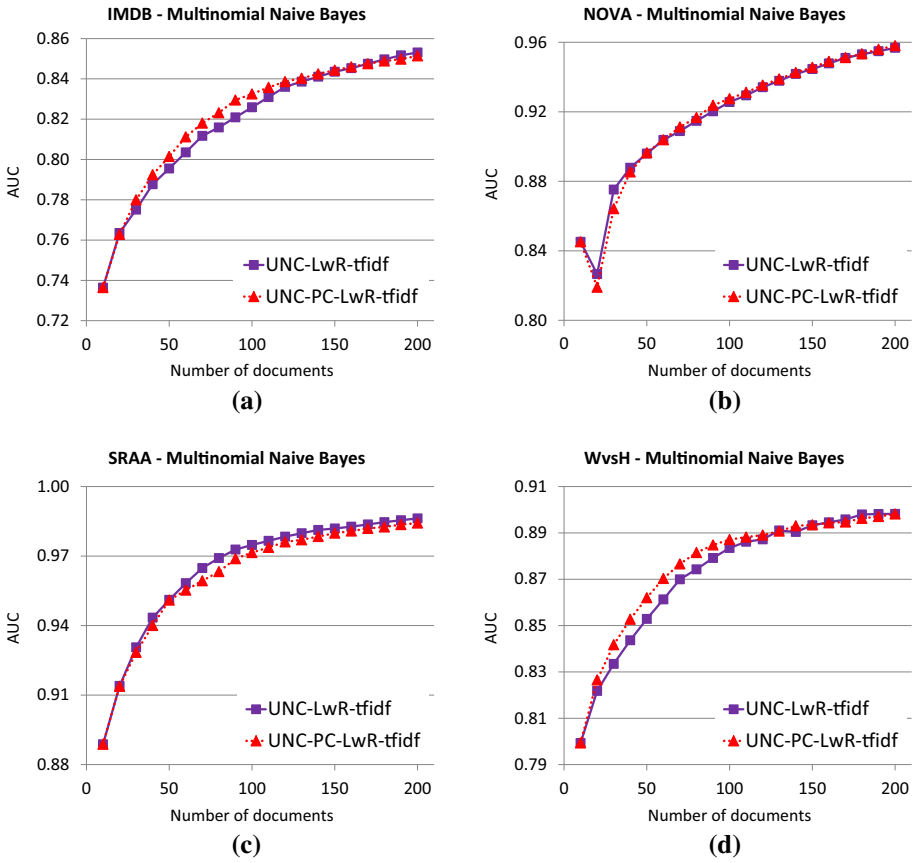

Fig. 7 Comparison of LwR using UNC and UNC-PC for all datasets with tf-idf representation and using multinomial naïve Bayes classifier

Table 8 Significant W/T/L counts for UNC-PC versus UNC. UNC-PC improves over UNC significantly for all three classifiers and most of the datasets

UNC-PC versus UNC	MNB	LR	SVM
Win	IMDB, WvsH	SRAA, NOVA	SRAA, NOVA, WvsH
Tie	NOVA	WvsH	–
Loss	SRAA	IMDB	IMDB

feedback *and* a learner that can utilize such rich feedback. This is an exciting future research direction that we would like to pursue.

We showed that our strategy to incorporate rationales works well for text classification. The proposed framework can potentially be used for non-text domains where the domain experts can provide rationales for their decisions, such as medical domain where the doctor can provide a rationale for his/her diagnosis and treatment decisions. In our framework, we place higher weights on rationales and lower weights on other features, thus our approach can be applied to domains where features represent presence/frequencies of characteristics, such as whether a patient is infant/young/old, whether the cholesterol level is low/medium/high,

etc. Each domain is expected to have its own unique research challenges and working with other domains is another interesting future research direction. Evaluating the framework on non-text domains is left as future work.

7 Related work

The closest related work deals with eliciting rationales from users and incorporating them into the learning. Zaidan et al. (2007) and Zaidan et al. (2008) incorporated rationales into the training of support vector machines for text classification. We provided a detailed description of Zaidan et al. (2007) in Sect. 4.1.1 and chose it as one of the baselines for our approach.

Donahue and Grauman (2011) extended the work of Zaidan et al. (2007) to incorporate rationales for visual recognition task. They proposed eliciting two forms of visual rationales from the labelers. First, they asked labelers to mark spatial regions in an image as rationales for choosing a label for the image. Second, they asked labelers to comment on the nameable visual attributes (based on a predefined vocabulary of visual attributes) that influenced their choices the most. For both forms of rationales, they created contrast examples that lack the rationale and incorporated the contrast examples and pseudoexamples into the training of support vector machines.

Parkash and Parikh (2012) proposed a method to incorporate labels and feature feedback for image classification task. They asked users to provide labels for images, and for each image that was predicted incorrectly by the classifier, they asked users to provide explanations in the form of attributed-based feedback. The attribute feedback was based on relative attributes (Parikh and Grauman 2011) that are mid-level concepts that can be shared across various class labels. In their approach, the feature feedback provided by the labelers is propagated to other unlabeled images that match the explanation provided by the labelers.

However, much of the work presented above is specific to a particular classifier, such as support vector machines. The framework we present is classifier-agnostic and we showed that our method works across classifiers and feature representations. Additionally, we provide a novel active learning approach tailored for the learning with rationales framework, whereas most of the previous work used random sampling and/or traditional uncertainty sampling for selecting documents for annotation.

Druck et al. (2009) proposed an approach for sequence labeling task in which the active learner selects useful queries and asks labelers to provide annotation for features, rather than annotation for instances. Similarly, Small et al. (2011) presented an approach for incorporating feature annotations into training of support vector machines for text classification. In their approach, they asked labelers to provide a ranked list of features, and added additional constraints into support vector machines to exploit the ranked features. Das et al. (2013) asked users to identify features from a list of labeled documents, and suggest features that would help the classifier to label future documents. They presented an approach to incorporate feature labels using a locally-weighted logistic regression classifier. In these three approaches, the learner elicits only feature annotations from labelers, whereas in our approach, the learner elicits a label and a rationale for documents.

Another line of related work is active learning with instance and feature annotations. For example, Melville and Sindhwani (2009) and Attenberg et al. (2010) presented a pooling multinomials approach to incorporate labeled instances and labeled features into multinomial naïve Bayes. We provided a detailed description of Melville and Sindhwani (2009) in Sect. 4.1.2 and chose it as one of the baselines for our approach.

Raghavan and Allan (2007) and Raghavan et al. (2006) proposed tandem learning to incorporate instance annotations and feature feedback into support vector machines. For features, they asked asked labelers to provide feedback as to whether the features are discriminative or not. They incorporated feature feedback by scaling all the important features by a higher weight, a, and scaling all the other features by a lower weight, b. The difference between their approach and our approach is that in their approach, features are not tied to any documents, and they scale all the important features that appear in all the documents by weight a and all other features by weight b, where $a = 10$ and $b = 1$, whereas in our approach, rationales are tied to the documents in which they appear as rationales and thus, our approach provides a more granular regularization, as explained in Sect. 3.1.

8 Conclusion

We introduced a novel framework to incorporate rationales into active learning for document classification. Our simple strategy to incorporate rationales can utilize any off-the-shelf classifier. The empirical evaluations on four text datasets with binary and tf-idf representations and three classifiers showed that our proposed method utilizes rationales effectively. We compared our classifier-agnostic approach to three classifier-specific approaches from the literature and showed that our method performs at least as well. Additionally, we presented an active learning strategy that is tailored specifically for the learning with rationales framework and empirically showed that it improved over traditional active learning on at least two out of four datasets using multinomial naïve Bayes, logistic regression, and support vector machines.

Acknowledgements This material is based upon work supported by the National Science Foundation CAREER Award No. 1350337.

References

Attenberg, J., Melville, P., & Provost, F. (2010). A unified approach to active dual supervision for labeling features and examples. In *European conference on machine learning and knowledge discovery in databases*, pp. 40–55.

Chapelle, O., Schölkopf, B., & Zien, A. (Eds.). (2006). *Semi-supervised learning*. Cambridge, MA: MIT Press.

Cleveland, W. S., & Devlin, S. J. (1988). Locally weighted regression: An approach to regression analysis by local fitting. *Journal of the American Statistical Association*, *83*(403), 596–610.

Das, S., Moore, T., Wong, W. K., Stumpf, S., Oberst, I., McIntosh, K., et al. (2013). End-user feature labeling: Supervised and semi-supervised approaches based on locally-weighted logistic regression. *Artificial Intelligence*, *204*, 56–74.

Donahue, J., & Grauman, K. (2011). Annotator rationales for visual recognition. In *2011 IEEE international conference on computer vision (ICCV)*, pp. 1395–1402.

Druck, G., Settles, B., & McCallum, A. (2009). Active learning by labeling features. In *Proceedings of the 2009 conference on empirical methods in natural language processing: Volume 1-volume 1*, pp. 81–90.

Fawcett, T. (2006). An introduction to roc analysis. *Pattern Recognition Letters*, *27*(8), 861–874.

Fung, G. M., Mangasarian, O. L., & Shavlik, J. W. (2002). Knowledge-based support vector machine classifiers. In *Advances in neural information processing systems*, pp. 521–528.

Girosi, F., & Chan, N. T. (1995). Prior knowledge and the creation of virtual examples for rbf networks. In *Neural networks for signal processing [1995] V. Proceedings of the 1995 IEEE workshop*, pp. 201–210.

Guyon, I. (2011). Results of active learning challenge.

Lewis, D., & Catlett, J. (1994). Heterogeneous uncertainty sampling for supervised learning. In *Proceedings of the eleventh international conference on machine learning*, pp. 148–156.

Lewis, D. D., & Gale, W. A. (1994). A sequential algorithm for training text classifiers. In *ACM SIGIR conference on research and development in information retrieval*, pp. 3–12.
Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011). Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, pp. 142–150.
Melville, P., Gryc, W., & Lawrence, R. D. (2009). Sentiment analysis of blogs by combining lexical knowledge with text classification. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 1275–1284.
Melville, P., & Sindhwani, V. (2009). Active dual supervision: Reducing the cost of annotating examples and features. In *Proceedings of the NAACL HLT 2009 workshop on active learning for natural language processing*, pp. 49–57.
Parikh, D., & Grauman, K. (2011). Relative attributes. In *2011 IEEE international conference on computer vision (ICCV)*. IEEE, pp. 503–510.
Parkash, A., & Parikh, D. (2012). Attributes for classifier feedback. In *Computer vision–ECCV 2012*. Springer, pp. 354–368.
Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.
Raghavan, H., & Allan, J. (2007). An interactive algorithm for asking and incorporating feature feedback into support vector machines. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 79–86.
Raghavan, H., Madani, O., & Jones, R. (2006). parkash:eccv2012. *Journal of Machine Learning Research*, *7*, 1655–1686.
Ramirez-Loaiza, M. E., Sharma, M., Kumar, G., & Bilgic, M. (2016). Active learning: An empirical study of common baselines. *Data Mining and Knowledge Discovery*, 1–27. doi:10.1007/s10618-016-0469-7.
Roy, N., & McCallum, A. (2001). Toward optimal active learning through sampling estimation of error reduction. In *International conference on machine learning*, pp. 441–448.
Segal, R., Markowitz, T., & Arnold, W. (2006). Fast uncertainty sampling for labeling large e-mail corpora. In *Conference on email and anti-spam*.
Settles, B. (2012). *Active learning. Synthesis lectures on artificial intelligence and machine learning*. San Rafael: Morgan & Claypool.
Seung, H. S., Opper, M., & Sompolinsky, H. (1992). Query by committee. In *ACM annual workshop on computational learning theory*, pp. 287–294.
Sharma, M., & Bilgic, M. (2013). Most-surely vs. least-surely uncertain. In *IEEE 13th international conference on data mining*, pp. 667–676.
Sharma, M., Zhuang, D., & Bilgic, M. (2015). Active learning with rationales for text classification. In *North American chapter of the association for computational linguistics human language technologies*, pp. 441–451.
Sindhwani, V., Melville, P., & Lawrence, R. D. (2009). Uncertainty sampling and transductive experimental design for active dual supervision. In *Proceedings of the international conference on machine learning*, pp. 953–960.
Small, K., Wallace, B., Trikalinos, T., & Brodley, C. E. (2011). The constrained weight space svm: Learning with ranked features. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 865–872.
Stumpf, S., Rajaram, V., Li, L., Burnett, M., Dietterich, T., Sullivan, E., et al. (2007). Toward harnessing user feedback for machine learning. In *Proceedings of the 12th international conference on intelligent user interfaces*, pp. 82–91.
Stumpf, S., Rajaram, V., Li, L., Wong, W. K., Burnett, M., Dietterich, T., et al. (2009). Interacting meaningfully with machine learning systems: Three experiments. *International Journal of Human-Computer Studies*, *67*(8), 639–662.
Tong, S., & Koller, D. (2001). Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, *2*, 45–66.
Towell, G. G., & Shavlik, J. W. (1994). Knowledge-based artificial neural networks. *Artificial Intelligence*, *70*(1), 119–165.
Towell, G. G., Shavlik, J. W., & Noordewier, M. (1990). Refinement of approximate domain theories by knowledge-based neural networks. In *Proceedings of the eighth national conference on artificial intelligence*, pp. 861–866.
Zaidan, O., Eisner, J., & Piatko, C. D. (2007). Using "annotator rationales" to improve machine learning for text categorization. In *HLT-NAACL*, pp. 260–267.
Zaidan, O. F., Eisner, J., & Piatko, C. (2008). Machine learning with annotator rationales to reduce annotation cost. In *Proceedings of the NIPS* 2008 workshop on cost sensitive learning*.

Zhu, J., & Hovy, E. (2007). Active learning for word sense disambiguation with methods for addressing the class imbalance problem. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning*, pp. 783–790.