Self-Tuned Descriptive Document Clustering Using a Predictive Network

Austin J. Brockmeier, *Member, IEEE*, Tingting Mu, *Member, IEEE*, Sophia Ananiadou, and John Y. Goulermas, *Senior Member, IEEE*

Abstract—Descriptive clustering consists of automatically organizing data instances into clusters and generating a descriptive summary for each cluster. The description should inform a user about the contents of each cluster without further examination of the specific instances, enabling a user to rapidly scan for relevant clusters. Selection of descriptions often relies on heuristic criteria. We model descriptive clustering as an auto-encoder network that predicts features from cluster assignments and predicts cluster assignments from a subset of features. The subset of features used for predicting a cluster serves as its description. For text documents, the occurrence or count of words, phrases, or other attributes provides a sparse feature representation with interpretable feature labels. In the proposed network, cluster predictions are made using logistic regression models, and feature predictions rely on logistic or multinomial regression models. Optimizing these models leads to a completely self-tuned descriptive clustering approach that automatically selects the number of clusters and the number of features for each cluster. We applied the methodology to a variety of short text documents and showed that the selected clustering, as evidenced by the selected feature subsets, are associated with a meaningful topical organization.

Index Terms—Descriptive clustering, feature selection, logistic regression, model selection, sparse models

1 Introduction

Exploratory data analysis techniques such as clustering can be used to identify subsets of data instances with common characteristics. Users can then explore the data by examining some instances in each cluster, rather than examining instances from the full dataset. This enables users to efficiently focus on relevant subsets of large datasets, especially for collections of documents [1]. In particular, *descriptive clustering* consists of automatically grouping sets of similar instances into clusters and automatically generating a human-interpretable description or summary for each cluster. Each cluster's description allows a user to ascertain the cluster's relevance without having to examine its contents. For text documents, a suitable description for each cluster may be a multi-word label, extracted title, or a list of characteristic words [2]. The quality of the clustering is important, such that it aligns with a user's idea of similarity, but it is equally important to provide a user with an informative and concise summary that accurately reflects the contents of the cluster. However, objective criteria for evaluating the descriptions as a whole, which do not resort to human evaluation, have been largely unexplored.

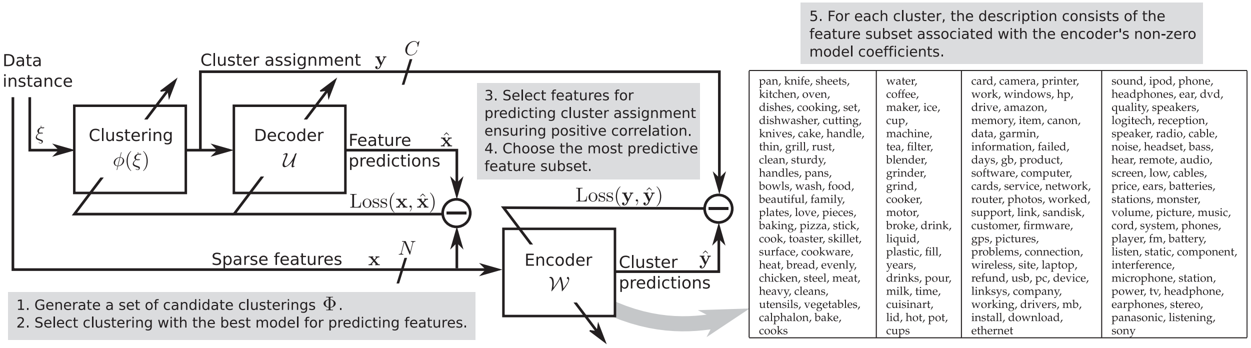
With the aim of defining an objective criterion, we consider a direct correspondence between description and prediction. We assume each instance is represented with sparse features (such as a bag of words), and each cluster will be described by a subset of features. A cluster's description should summarize its contents, such that the description alone should enable a user to predict whether an arbitrary instance belongs to a particular cluster. Likewise, a machine classifier trained using the features subset should also be predictive of the cluster membership. The classification accuracy provides an objective and quantitative criterion to compare among different feature subsets.

To serve as a concise description, the number of features used by the classifier must be limited (e.g., a linear classifier that uses all features is not easily interpretable). A relatively small set of predictive features can be identified using various feature selection methods [3], [4]. In particular, we identify features subsets by various statistical and information-theoretic criteria [5] and by training linear classifiers with additional sparsity-inducing regularizations [6], [7], [8], e.g., the ℓ_1-norm for the Lasso [9] or a combination of ℓ_1 and ℓ_2-norms for the Elastic Net [10], such that only a small set of features have non-zero coefficients. In a similar spirit, Lasso has been used for selecting predictive features for explaining classification models [11].

In addition to the cardinality constraint on the number of features, we only permit features that are positively correlated with a given cluster, i.e., features whose presence are indicative of the cluster. This constraint ensures that no cluster is described by the absence of features, which are present in other clusters. For instance, given a corpus of book and movie reviews, the positivity constraint avoids a cluster consisting of mainly of book reviews from being described as $\neg$`movie`, i.e., the absence of the word feature `movie`.

- *A.J. Brockmeier and J.Y. Goulermas are with the School of Electrical Engineering, Electronics, & Computer Science, University of Liverpool, Liverpool L69 3BX, United Kingdom.*
E-mail: ajbrockmeier@gmail.com, j.y.goulermas@liverpool.ac.uk.
- *T. Mu and S. Ananiadou are with the School of Computer Science, University of Manchester, Manchester M1 7DN, United Kingdom.*
E-mail: tingtingmu@me.com, sophia.ananiadou@manchester.ac.uk.

Manuscript received 1 Dec. 2016; revised 18 Nov. 2017; accepted 3 Dec. 2017. Date of publication 23 Feb. 2018; date of current version 10 Sept. 2018.
(Corresponding author: Austin J. Brockmeier.)
Recommended for acceptance by H. Lee.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TKDE.2017.2781721


Data instance
ξ
Clustering $\phi(\xi)$
Cluster assignment $\mathbf{y}$
C
Decoder $\mathcal{U}$
Feature predictions $\hat{\mathbf{x}}$
$\text{Loss}(\mathbf{x}, \hat{\mathbf{x}})$
$\text{Loss}(\mathbf{y}, \hat{\mathbf{y}})$
Sparse features $\mathbf{x}$
N
Encoder $\mathcal{W}$
Cluster predictions $\hat{\mathbf{y}}$

1. Generate a set of candidate clusterings Φ.
2. Select clustering with the best model for predicting features.
3. Select features for predicting cluster assignment ensuring positive correlation.
4. Choose the most predictive feature subset.
5. For each cluster, the description consists of the feature subset associated with the encoder's non-zero model coefficients.

pan, knife, sheets, kitchen, oven, dishes, cooking, set, dishwasher, cutting, knives, cake, handle, thin, grill, rust, clean, sturdy, handles, pans, bowls, wash, food, beautiful, family, plates, love, pieces, baking, pizza, stick, cook, toaster, skillet, surface, cookware, heat, bread, evenly, chicken, steel, meat, heavy, cleans, utensils, vegetables, calphalon, bake, cooks	water, coffee, maker, ice, cup, machine, tea, filter, blender, grinder, grind, cooker, motor, broke, drink, liquid, plastic, fill, years, drinks, pour, milk, time, cuisinart, lid, hot, pot, cups	card, camera, printer, work, windows, hp, drive, amazon, memory, item, canon, data, garmin, information, failed, days, gb, product, software, computer, cards, service, network, router, photos, worked, support, link, sandisk, customer, firmware, gps, pictures, problems, connection, wireless, site, laptop, refund, usb, pc, device, linksys, company, working, drivers, mb, install, download, ethernet	sound, ipod, phone, headphones, ear, dvd, quality, speakers, logitech, reception, speaker, radio, cable, noise, headset, bass, hear, remote, audio, screen, low, cables, price, ears, batteries, stations, monster, volume, picture, music, cord, system, phones, player, fm, battery, listen, static, component, interference, microphone, station, power, tv, headphone, earphones, stereo, panasonic, listening, sony

Fig. 1. Proposed approach and network architecture for descriptive clustering (the notation for the decoupled auto-encoder is described in Table 1).

In general, this constraint can be enforced by admitting only features that are positively correlated with a particular cluster; for linear classifiers, this can be done by enforcing the constraint that the coefficients are non-negative [12].

Constraining the number of features and constraining the positivity will inevitably limit the performance of the classifier. The natural question is exactly how many features are needed to ensure a 'reasonable' classification performance. We use a model order selection criterion as a principled approach to answer this question. We model the probability of class membership given the different subsets of features using logistic regression and use the Bayesian information criterion (BIC) [13] to select the model corresponding to a feature subset that is both predictive and parsimonious.

Given a clustering of the instances, we apply the aforementioned approach to automatically generate an interpretable classifier for each cluster. The features of the classifier form the description of the cluster, and the network of classifiers is a compact approximation of the original cluster assignments. The two benefits of this paradigm versus previous descriptive clustering approaches [2], [14], [15], [16] are the ability to quantify the description accuracy as well as a principled and automatic way to select the size of feature subsets that serve as the descriptors for each cluster.

Similarly, predictive performance can be used to select the clustering itself and in particular, the number of clusters. Although a variety of criteria for selecting the number of clusters exist [17], they are often specific to the underlying objective of the clustering algorithm. Independent from the clustering algorithm, we propose to select the number of clusters that is most predictive of the original feature occurrences. For each clustering, we assume the sparse features are conditionally independent given the cluster assignment and model them by either a multinomial distribution, if the feature values are non-negative counts, or a multivariate Bernoulli, if the feature values are binary. The optimal clustering is then chosen by model selection.

TABLE 1
Notation for the Descriptive Clustering Auto-Encoder Network

Notation	Description
ξ	Data instance
$\mathbf{x}$	An instance's sparse feature vector
$x^{(i)}$	Value of the i-th feature
N	Number of features
$\mathbf{y}$	An instance's cluster assignment vector
$y^{(c)}$	Binary membership variable for c-th cluster
C	Number of clusters
$\phi(\xi)$	Clustering function, returns an assignment vector
Φ	Set of candidate clusterings
$\text{Decode}(\mathbf{y}, \mathcal{U})$	Cluster-to-feature decoder with parameters $\mathcal{U}$
$\hat{\mathbf{x}}$	Decoder's predicted feature vector
$\text{Encode}(\mathbf{x}, \mathcal{W})$	Cluster encoder with parameters $\mathcal{W}$
$\hat{\mathbf{y}}$	Encoder's predicted cluster assignment
$\mathcal{S}^{(c)}$	Set of selected feature indices for c-th cluster

Our approach is a self-tuned method for descriptive clustering, which we refer to as predictive-descriptive clustering (PDC), and is based on two predictive networks, as shown in Fig. 1. By choosing from among different network architectures, both the number of clusters and the feature subsets for describing each cluster are automatically optimized. Although we assume sparse count or binary occurrence-based features to use for the description and model selection, the clusterings themselves can be derived from arbitrary representations. We explore the approach in the context of short text documents with words and phrases as features, but the approach is applicable to any data with sparse count-valued features.

Our main contributions are (1) a new quantitative evaluation for descriptive clustering; (2) PDC as a unified framework to automatically select both the clustering itself (across different numbers of clusters) and also select a set of features (including its size) to describe each cluster; (3) a comparison of feature selection algorithms including regularized and constrained logistic regression; (4) a direct comparison to an existing topic modeling approach [18], which shows that PDC both performs better and is more efficient; and (5) motivating examples of PDC on a variety of text datasets.

We begin by discussing related work in exploratory data analysis for text datasets in Section 2. We then present the proposed methodology in Section 3, detailing the selection of clustering, selection of candidate feature subsets, and selection of model size. In Section 4, we apply the proposed approach to publicly available text datasets (movie, book, and product reviews along with Usenet newsgroup posts, news summaries, academic grant abstracts, and recipe ingredient lists) and show that meaningful and descriptive features subsets are selected for the clusters. Furthermore, we show that the interpretable classifiers that use the feature subsets are accurate.

2 Relation to Existing Approaches

There has been extensive research on clustering and other unsupervised methods, namely topic modeling but also low-dimensional embeddings [19], [20], [21], [22], for exploring text datasets. Some particularly relevant work has focused on website search results [15]. We highlight approaches that have considered user interpretation of the results in the form of descriptive keywords, phrases, or titles that summarize the semantic content of clusters and topics for a user. For datasets with known ground-truth topic categories, document clustering can be evaluated using correspondence measures, but comparisons of descriptive labels have often relied on human evaluation. Comparisons among description mechanisms is especially challenging, since the datasets, clustering or modeling paradigms, and form of the description varies widely. Although some user evaluations have concentrated on which labels users prefer [23], evaluation should concentrate on whether the descriptions aid a user in predicting the most relevant cluster or topic [24], [25]. To our knowledge, no previous approach has posed descriptive clustering as a prediction problem with objective quantification in terms of classification performance. The other unique contributions of our approach include a principled approach to select the number of features in the description, and an automatic approach for selecting the number of clusters that is independent of the clustering algorithm but dependent on the cluster assignments and the binary feature representation.

2.1 Descriptive Clustering for Text Datasets

The motivation for applying descriptive clustering to text datasets is that it can be used as an information retrieval mechanism. A user can efficiently scan the descriptions for relevancy versus having to determine which clusters are relevant by manually checking the document instances. Scatter-gather [2], [24], [26], [27] is an iterative procedure that uses multiple stages of descriptive clustering to help a user find relevant documents. An initial clustering is given along with some description or preview of each cluster to the users, who are then asked to select clusters of interest. Instances within the selected clusters are combined and clustered again. This continues until a user hones in on a relevant set of documents. The quality of the automatic description is crucial to enable a user to recognize which clusters are relevant.

This exploratory approach should be contrasted to classic query-based information retrieval systems. While query-based systems predominate web searches, exploratory analysis is useful when the user does not know what topics are within a corpus (which could vary between a set of full-text documents to a set of short summaries of each result returned by a search engine) or is unable to formulate a query to retrieve relevant instances. In particular, the exploratory approach is useful for a user that believes, “I will know it when I see it.”

Descriptive clustering can be performed by first clustering and then finding the set of features associated with each cluster. This enables any applicable clustering algorithms to be used. Selecting features that best inform a user on the contents of a cluster (the purpose of this study) is the subsequent challenge. The most basic approach is to describe each cluster by the most likely words in the cluster [2], titles (if available) near the center of each cluster [2], or phrases with similar context as the most likely words [14]. However, these features may not be optimal for discriminating between different clusters. Other scoring criteria such as mutual information [28] (i.e., information gain [29] rather than pointwise mutual information [30]) may be used to select more discriminating features (e.g., keywords or phrases) for the clusters.

Another approach is to first, or simultaneously, group features such that the grouping of features affects the clustering. Grouping features [31] can mitigate issues with feature sparsity and noise [32], and the use of grouped features has been shown to improve clustering performance [33], [34]. A novel approach is to directly learn a joint vector space representing both features (descriptive phrases) and instances [16]. The vectors representing the features and instances are optimized such that nearby vectors are semantically similar. Given this joint vector space, each cluster of instances can be labeled by some of the features residing within its boundaries.

With the aforementioned approaches it is not clear how to objectively measure the selected feature lists or labels that serve as descriptors. Our hypothesis is that any description is only useful if it would enable a user to accurately predict the contents of the cluster. In this case, finding sets of features to describe each cluster can be seen as a feature selection problem. For instance, one can train a decision tree[35] for each cluster to classify instances directly by the presence or absence of certain features. The boolean expression corresponding to the decision tree serves as a description of the set [36].

A decision tree and clustering can be formed simultaneously using hierarchical clustering based on the presence or absence of individual features [37] or phrases as in suffix-tree clustering [38], [39]. These approaches explicitly link the description with the organization of the documents and by consequence cannot be applied to arbitrary clusterings.

2.2 Topic Models and Automatic Topic Labeling

Besides clustering, other models can be used for exploratory analysis of large text datasets. Latent semantic indexing [40] is a matrix decomposition technique related to principal component analysis that can be applied to a bag-of-words representation of documents to group together words used in similar contexts and also group together documents which contain these contexts. A user could browse the list of highly weighted words associated with each latent dimension, which serves as a sort of description, to choose relevant dimensions. Alternatively, instead of a list, a single phrase that best describes each dimension can be used and then documents can be assigned based on the presence of this phrase [41].

Topic models offer a probabilistic interpretation of the themes present in bag-of-words representation of document collections [42], [43]. Each document is modeled as a mixture of topics/themes, with each topic or theme associated with a distribution of words. Specifically, latent topic models [44], [45] assume the words in each document are drawn from a mixture of word distributions, where each word distribution is associated with a topic. Each document has its own mixture of topics, and each topic defines a distribution over all the words. Given a document, the topics are unknown (latent) and are inferred from the distribution of words appearing in the document or in each sentence [46].

Browsing lists of the most probable words for each topic is not an efficient method to interpret topic models. Instead, various approaches to automatically label the individual topics with descriptive phrases, or a subset of distinguishing features, are useful for annotating models with a large number of topics [18], [23], [47]. Thus, the problem of automatically labeling topic models is closely related to descriptive clustering. Clustering bag-of-word representations can be seen as an extreme form of topic modeling, where each document is associated with a single topic. Since a clustering can be treated as a topic model, automatic topic labeling techniques can be applied to descriptive clustering.

In particular, we compare our feature selection approach to the topic labeling method proposed by Mei et al. [18]. Their method most closely resembles ours in that it is based solely on the statistics of the corpus (this is in contrast to other topic labeling methods that rely on external resources, for instance on the alignment of topics to the titles of Wikipedia articles [23], [47] or finding a hypernym via WordNet [48] for a topic's common words [49]). In Mei et al.'s approach, features are scored based on how predictive they are of topic-related words. The score is adjusted to discriminate between topics, and further adjusted when multiple features are selected in order to ensure they are not redundant [50]. This ensures the selected features are truly comprehensive. The drawbacks of their approach are its computational complexity, since a multinomial model over the features is formed for each candidate descriptor, and the score adjustments are based on trade-off parameters that must be selected by a user. Furthermore, automatically selecting the number of terms per topic is not considered.

Our proposed PDC framework can also be applied to topic models, but topic models are only useful for document retrieval when each topic can be clearly associated with a subset of documents, and each topic can be succinctly described. Topics that appear uniformly throughout the dataset may be useful for a global summary but are not interesting for searching for a subset of relevant documents. One approach to achieve localized topics is to combine clustering and topic modeling [51]. Another approach is to aim for a parsimonious model [52] that uses the minimal number of topics each associated with topic specific words, where the number of topics and words are directly optimized via a model order selection criterion. Soleimani and Miller's parsimonious model approach [52] yields models where each document is associated with only a few topics. Together this means the parsimonious modeling more closely resembles descriptive clustering. A key difference with our approach is that many more features are associated with each topic in the parsimonious model. This is because their model seeks to predict the features with the topics, while ours seeks to predict the topics from a subset of features.

2.3 Feature Selection

Given a particular cluster, predicting whether instances belong to the cluster is a standard classification problem, and selecting the best subset of features for this task is a feature selection problem [3], [4], [29]. Choosing a small subset of maximally predictive features is a difficult task. One tractable approach is to use a greedy algorithm that adds (or removes) features until a desired cardinality is reached. For small feature sets, one can consider a stepwise approach, where at each step the feature that either improves (or contributes the least) to the classification performance is added (or removed). However, the stepwise approach, which must fit a model for each candidate feature, cannot scale to cases with tens of thousands of features as encountered with text data. Tractable greedy algorithms that can scale to large number of features can be divided into three groups:

- *Filter approaches* [3] that do not assume specific form to the classifier but use a specific criterion to judge the relevance of individual features or feature subsets. The simplest approach is to rank features by a selection criterion (for instance, mutual information, which has been shown to perform well for document classification [28], weighted likelihood ratio [53], or other heuristics [20]) and select the top-ranked subset.
- *Joint filter approaches* that consider the dependency and (possible redundancy) among features [54]. In particular, the information theoretic criteria [5] that consider the pairwise dependence include CMIM [55], MRMR [56], and JMI [57], [58].
- *Heuristic approaches* that are specific to the classifiers. For instance, decision trees [35] can be trained by choosing the feature that reduces a metric such as the Gini impurity value. For logistic regression, the magnitude of gradient can be used as a criterion for feature inclusion. Methods using this approach include grafting [59], logistic regression based orthogonal matching pursuit [60], and greedy cardinality-constrained optimization [61].

Another tractable approach for features selection is based on fitting a linear classifier with additional sparsity-inducing regularizations on the coefficients of features in the objective function. With certain choices of regularization parameters, many of the optimal coefficients are exactly zero and only a small set of features are associated with non-zero coefficients [6], [7], [8]. Convex regularizations of this sort are the ℓ_1-norm (Lasso) [9] and the $\ell_1 + \ell_2$–norm (Elastic Net) [10].

Any of these approaches can be adjusted to ensure that the features are positively correlated with a target class, i.e., a feature's occurrence rate given the class should be higher than its average rate. For the regularization-based approaches, the positivity constraints on the coefficients can be enforced during logistic regression model optimization.

3 Predictive-Descriptive Clustering (PDC)

The PDC framework consists of two prediction tasks: predicting the original feature occurrences based on the cluster assignments and predicting whether an instance belongs to a specific cluster using only a small set of feature dimensions that serve as the description for the cluster. The first task provides a quantitative objective to automatically select from clusterings with different numbers of clusters. Each cluster is associated with a certain distribution of the features, with some features occurring more (or less) frequently for instances within that cluster. If all the instances assigned to the same cluster have similar feature distributions, then the knowledge of the cluster assignment will be predictive of the feature occurrences. The second task of predicting the cluster membership is clearly dependent on the chosen clustering. The amount of information carried by the clustering increases with more clusters, but the difficulty of predicting cluster membership also increases with more, finer-grained clusters. Additionally, for a fixed number of clusters there is an inherent trade-off between prediction performance and the number of features.

As a tractable approach to choose the number of clusters and features, we use a multistage modeling process. At each stage, we estimate a set of candidate models and select the best model from the candidate set.

- *Create candidate clusterings.* A set of possible clusterings are formed. These clusterings may vary in the number of clusters, arise from different clustering algorithms, or use different data representations.
- *Select the most predictive clustering.* For each clustering, a model is trained to predict feature occurrences

from the cluster assignments. The clustering associated with the most predictive model is selected.

- *Create descriptive feature subsets.* For each cluster, different subsets of features are chosen by a feature selection mechanism. In particular, candidate feature subsets are identified using logistic regression with positivity constraints and sparsity-inducing regularization by varying the amount of regularization.
- *Select the most informative feature subset.* A logistic regression model is trained for each candidate feature subset. The best model is selected by a model order selection criterion that balances cardinality with predictive performance.

Each stage is associated with a standard modeling process: clustering, regression, feature selection, and model selection. Before detailing each stage, we introduce an interpretation of the PDC framework as an approach for estimating an auto-encoder neural network with a binary hidden layer [62], [63], [64].

3.1 PDC as a Binary Auto-Encoder Network

The predictive architecture in the PDC framework can be interpreted as an auto-encoder network [64]. The network consists of an encoder that maps each instance to its cluster assignment vector, which serves as the hidden layer of the network, and a decoder that tries to reconstruct the original features based on the cluster assignment. With a standard auto-encoder, the output of the encoder is directly fed to the input, and both the encoder and decoder are adapted in order to minimize the reconstruction error of the decoder, subject to any additional constraints on the hidden layer.

Alternatively, in our stage-wise approach, we decouple the auto-encoder by using a clustering algorithm to provide the cluster assignments that serve both as the template for training the encoder and as the input to the decoder, as shown in Fig. 1. We proceed to formulate both the coupled and decoupled forms of the auto-encoder. Table 1 contains the main notation.

We assume each data instance, denoted ξ, is associated with a sparse feature vector $\mathbf{x} = [x^{(1)}, \dots, x^{(N)}] \in \mathbb{Z}_+^N$, where N is the number of features and $x^{(i)}$ is the number of times the ith feature occurs in this instance or a binary value indicating if the ith feature was present in the instance. The encoder assigns each instance to one or more clusters based on the presence of features. A cluster assignment with C clusters is represented by the vector $\mathbf{y} = [y^{(1)}, \dots, y^{(C)}] \in \{0,1\}^C$, where $y^{(c)} = 1$ if the instance is assigned to cluster c. The decoder predicts the feature occurrences from the cluster assignments; the prediction is denoted $\hat{\mathbf{x}} = \mathrm{Decode}(\mathbf{y}, \mathcal{U})$, where $\mathcal{U}$ denotes the parameters of the decoder. The cluster assignment vector $\mathbf{y}$ correspond to the hidden layer activations. The objective for training the auto-encoder is to minimize both the expected loss of the feature predictions and also the number of features used in the encoder, and can be written as

$$\begin{aligned}&\min_{C,\mathcal{W},\mathcal{U}} \mathbb{E}[\mathrm{Loss}(\mathbf{x}, \hat{\mathbf{x}})] + \alpha\Omega(\mathcal{W})\\ &\quad \hat{\mathbf{x}} = \mathrm{Decode}(\mathbf{y}, \mathcal{U}), \quad \mathbf{y} = \mathrm{Encode}(\mathbf{x}, \mathcal{W}) \\ &\quad \text{subject to } \mathbf{y} \in \{0,1\}^C,\end{aligned} \tag{1}$$

where $\mathcal{W}$ is the set of parameters of the encoding model, $\Omega(\mathcal{W})$ is a penalty function based on the number of features used in the encoding model, and α is a trade-off parameter.

Even without the additional penalty on the number of features, finding the optimal auto-encoder is a difficult problem to solve [64]. Rather than attempt to jointly learn all of the parameters, we decouple the decoder from the encoder, using a separate clustering function $\phi : \xi \mapsto \mathbf{y}$ to define the cluster assignments (ϕ maps each instance ξ to a cluster assignment vector $\mathbf{y}$). The clustering assignments defined by ϕ are used as the input to the decoder and also as the target for the descriptive encoder. The decoupled auto-encoder optimization can be written as

$$\begin{aligned}&\min_{\phi,\mathcal{U},\mathcal{W}} \mathbb{E}[\mathrm{Loss}(\mathbf{x}, \hat{\mathbf{x}}) + \beta\,\mathrm{Loss}(\mathbf{y}, \hat{\mathbf{y}})] + \alpha\,\Omega(\mathcal{W}),\\ &\hat{\mathbf{x}} = \mathrm{Decode}(\mathbf{y}, \mathcal{U}), \quad \mathbf{y} = \phi(\xi), \quad \hat{\mathbf{y}} = \mathrm{Encode}(\mathbf{x}, \mathcal{W}),\end{aligned} \tag{2}$$

where $\hat{\mathbf{y}}$ is the predicted cluster assignments from the descriptive encoder and β is a trade-off parameter between the loss functions.

This joint optimization is still difficult; however, for a fixed clustering, the encoder and decoder can be independently optimized, and the resulting objective value can be used to select from a set of different clusterings $\Phi = \{\phi_1, \dots, \phi_K\}$. Moreover, as in the coupled auto-encoder, each candidate clustering can be evaluated solely on the decoder performance (this avoids training an encoder for each possible clustering). The optimization corresponding to selecting the most predictive clustering can be written as

$$\begin{aligned}\phi^\star &= \arg\min_{\phi\in\Phi} \min_{\mathcal{U}} \mathbb{E}[\mathrm{Loss}(\mathbf{x}, \hat{\mathbf{x}})],\\ \hat{\mathbf{x}} &= \mathrm{Decode}(\mathbf{y}, \mathcal{U}), \quad \mathbf{y} = \phi(\xi).\end{aligned} \tag{3}$$

Given the vector of cluster assignments for the best clustering $\mathbf{y} = \phi^\star(\xi)$, the feature selection problem is then

$$\begin{aligned}&\min_{\mathcal{W}} \mathbb{E}[\mathrm{Loss}(\mathbf{y}, \hat{\mathbf{y}})] + \alpha\,\Omega(\mathcal{W}),\\ &\hat{\mathbf{y}} = \mathrm{Encode}(\mathbf{x}, \mathcal{W}),\end{aligned} \tag{4}$$

and can be performed independently for each cluster:

$$\begin{aligned}&\min_{\mathcal{S}^{(c)},\mathcal{W}^{(c)}} \mathbb{E}[\mathrm{Loss}(y^{(c)}, \hat{y}^{(c)})] + \alpha\,\Omega(\mathcal{W}^{(c)}),\\ &\hat{y}^{(c)} = \mathrm{Encode}(\mathbf{x}^{(c)}, \mathcal{W}^{(c)}), \quad \mathbf{x}^{(c)} = [x^{(d)}]_{d\in\mathcal{S}^{(c)}},\end{aligned} \tag{5}$$

where $\mathcal{S}^{(c)}$ is the set of selected feature indices. A complete solution to the descriptive clustering problem is the cluster assignment and the feature subsets $(\phi, \{\mathcal{S}^{(1)}, \dots, \mathcal{S}^{(C)}\})$. With this mathematical formulation we proceed to detail the optimizations involved in each step of the approach.

3.2 Predictive Clustering Selection

The PDC framework can be used with the cluster assignments of any clustering algorithm that produces flat clusterings. As a baseline algorithm we use spectral clustering [65], which can be efficiently implemented for sparse data when cosine similarity is used, and vary the number of clusters to create a set of candidate clusterings $\Phi = \{\phi_1, \dots, \phi_K\}$. (An alternative is to use a coarser set of cluster sizes for less computation.) We automatically select from among different clusterings based on how well the feature vectors can be predicted from the cluster assignments.

3.2.1 Predicting Features from Cluster Assignments

For the expected loss, we explore two approaches: in the first approach, we model binary features using logistic regression and use the cross-validated average of the negative log-likelihood; as a more scalable alternative, we model count-values features using multinomial regression.

Binary Features (Bernoulli Model). We model binary feature vectors based on the assumption that they are independent Bernoulli random variables given the cluster assignment. The probability[1] of a particular feature X being present given the cluster assignment $\mathbf{y} = [y^{(1)}, \ldots, y^{(C)}]$ is modeled as

$$\Pr(X = 1|\mathbf{y}) = \frac{1}{1 + e^{-u_0 - \mathbf{u}\cdot\mathbf{y}}} = \frac{1}{1 + e^{-\mathbf{u}'\cdot\mathbf{y}'}}, \tag{6}$$

where for compactness the bias u_0 is included with the coefficients in the parameter vector $\mathbf{u}' = [u_0, \mathbf{u}]$ and a constant feature is added to the cluster assignments $\mathbf{y}' = [1, \mathbf{y}]$. For a sample $\{(\mathbf{y}_i, x_i)\}_{i=1}^n$ of n instances the log-likelihood is

$$\ln \mathcal{L}\big(\mathbf{u}', |\{\mathbf{y}_i, x_i\}_i\big) = \sum_{i=1}^{n} x_i(\mathbf{u}' \cdot \mathbf{y}'_i) - \ln(1 + e^{\mathbf{u}'\cdot\mathbf{y}'_i}). \tag{7}$$

In practice, if a feature only occurs in a subset of clusters, the coefficient for the feature can become arbitrarily large while maximizing the log-likelihood. This behavior can be avoided by constraining the norm of the coefficients to be less than some value. For any constraint value, an optimization with an equivalent solution is obtained by minimizing the negative log-likelihood function combined with a scaled penalty on the squared ℓ_2-norm of the coefficients:

$$\underset{\mathbf{u}'}{\arg\min} - \ln \mathcal{L}\big(\mathbf{u}'|\{\mathbf{y}_i, x_i\}_i\big) + \frac{\tau}{2}\|\mathbf{u}\|_2^2. \tag{8}$$

To select the best clustering among multiple clustering candidates, each clustering is evaluated in terms of the cross-validated log-likelihood for the optimized coefficients. For a particular clustering defined by the mapping $\phi : \xi \mapsto \mathbf{y}$, the parameters for each feature are optimized and fixed, and the negative log-likelihood on a separate test sample $\{(\phi(\tilde{\xi}_i), \tilde{\mathbf{x}}_i)\}_i$ is combined across all N features:

$$\mathbb{E}[\mathrm{Loss}(\mathbf{x},\ \mathrm{Decode}(\mathbf{y}, \mathcal{U}))] = \sum_{j=1}^{N} - \ln \mathcal{L}(\mathbf{u}'_j, \{\tilde{\mathbf{y}}_i, \tilde{x}_i^{(j)}\}_i), \tag{9}$$

where $\tilde{\mathbf{y}}_i = \phi(\tilde{\xi}_i)$ and $\mathcal{U} = \{\mathbf{u}'_j\}_{j=1}^N$ is the set of coefficients.

In practice, some of the features may be extremely sparse (only a few non-zeros). Rather than attempt to predict these sparse features, we use only the subset of features that occur in at least 1 percent of the instances in the cross-validation sets.

Count-Valued Features (Multinomial Model). For sparse feature vectors corresponding to occurrence counts, such as bag-of-words, we assume the counts follow a multinomial distribution for each cluster. For a multinomial random vector $\mathbf{X} = [X^{(1)}, \ldots, X^{(N)}]$ described by the probability distribution $\mathbf{p} = [p_{(1)}, \ldots, p_{(N)}]$, $p_{(j)} \geq 0$, $\sum_j p_{(j)} = 1$, the probability mass function for observing the count vector $\mathbf{x} = [x^{(1)}, \ldots, x^{(N)}] \in \mathbb{Z}_+^N$ with a total count of $m = \sum_j x^{(j)}$ is

$$\Pr(\mathbf{X}{=}\mathbf{x}) = \frac{m!}{x^{(1)}! \cdots x^{(N)}!} p_{(1)}^{x^{(1)}} p_{(2)}^{x^{(2)}} \cdots p_{(N)}^{x^{(N)}}. \tag{10}$$

Given a sample $\{(\mathbf{x}_i, m_i)\}_{i=1}^n$ from this distribution, the log-likelihood function is

$$\ln \mathcal{L}(\hat{\mathbf{p}}|\{\mathbf{x}_i\}) = \sum_i \sum_j \ln m_i! - \ln\Big(x_i^{(j)}!\Big) + x_i^{(j)} \ln \hat{p}_{(j)}. \tag{11}$$

Rather than use the maximum likelihood estimate of $\hat{\mathbf{p}}$, we adopt additive smoothing to estimate the multinomial parameters for each cluster as $\hat{p}_{(j)}^{(c)} = \frac{1 + r_{(j)}^{(c)}}{N + \sum_{k=1}^{N} r_{(k)}^{(c)}}$, where $r_{(j)}^{(c)} = \sum_{i:y_i^{(c)}=1} x_i^{(j)}$ (the number of times the j-th word occurs in the cth cluster) and $\mathcal{U} = \hat{\mathbf{p}}$. The empirical log-likelihood of the sample itself is used to estimate the quality of the clustering, and to compare clusterings of different size we use the Akaike information criterion (AIC) [66]:

$$\mathbb{E}[\mathrm{Loss}(\mathbf{x},\ \mathrm{Decode}(\mathbf{y}, \mathcal{U}))] \approx -2 \ln \mathcal{L}(\hat{\mathbf{p}}|\{\mathbf{x}_i\}) + 2C. \tag{12}$$

Using AIC is more efficient than cross-validation and has the same asymptotic performance for model selection [67].

3.3 Descriptive Feature Selection

Given the clustering, training an interpretable encoder for each cluster is a supervised learning problem consisting of both feature selection and classifier training. Each cluster is treated as a class, and the corresponding dimension of the cluster membership vector $y^{(c)}$ is treated as the indicator for class membership. For each cluster $c \in \{1, \ldots, C\}$ the problem is to predict $y^{(c)}$ using a subset of features:

$$\begin{gathered}\underset{\mathcal{W}^{(c)}}{\arg\min}\ \mathbb{E}[\mathrm{Loss}(y^{(c)}, \hat{y}^{(c)})] + \alpha\, \Omega(\mathcal{W}^{(c)}) \\ \text{subject to } w \geq 0 \quad \forall w \in \mathcal{W}^{(c)}, \\ \mathbf{y} = [y^{(1)}, \ldots, y^{(C)}] = \phi(\xi), \quad \hat{y}^{(c)} = \mathrm{Encode}(\mathbf{x}, \mathcal{W}^{(c)}), \\ \Omega(\mathcal{W}^{(c)}) = |\{w \in \mathcal{W}^{(c)} : w > 0\}|,\end{gathered} \tag{13}$$

where $\mathcal{W}^{(c)}$ denotes the coefficients of the encoder model associated with the features, $\mathbf{y}$ represents the original cluster assignment vector and $\hat{y}^{(c)}$ is encoder's assignment for cluster c. The constraint ensures the features are positively associated with a cluster.

Optimizing Eq. (13) is not tractable due to the integral nature of the feature count $\Omega(\mathcal{W}^{(c)})$. Instead, for each cluster a candidate set of encoders using features subsets of varying cardinality are optimized to predict the cluster assignment, and the feature subset for the encoder that minimizes the cost is selected. Feature subsets can be generated by various algorithms or by optimizing a linear classifier with sparsity-inducing regularization. We select α by equating Eq. (13) to BIC by using the negative log-likelihood for the expected loss and setting $\alpha = \ln \sqrt{n}$, where n is the number of instances in the sample.

3.3.1 *Regularized and Constrained Logistic Regression for Feature Subset Generation*

Candidate feature subsets can be obtained by combining logistic regression with sparsity-inducing regularizations. For a given cluster, the probability that an instance is assigned to the cluster is modeled as a conditionally Bernoulli random variable:

$$\Pr(Y = 1|\mathbf{x}) = f_{\mathbf{w}'}(\mathbf{x}) = \frac{1}{1 + e^{-w_0 - \mathbf{w}\cdot\mathbf{x}}}, \tag{14}$$

where for compactness the bias w_0 is combined with the coefficients $\mathbf{w}' = [w_0, \mathbf{w}]$ and a constant is added to the feature vector $\mathbf{x}'_i = [1, \mathbf{x}]$. Given a sample $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, the

[1] For a Bernoulli random variable X with a mean frequency of p, the probability mass function can be written as $\Pr(X = x) = p^x(1-p)^{(1-x)}$, $x \in \{0, 1\}$. Given a sample $\{x_i\}_i$ from this distribution, the log-likelihood function is $\ln \mathcal{L}(\hat{p}|\{x_i\}) = \sum_i x_i \ln \hat{p} + (1 - x_i) \ln(1 - \hat{p})$.

log-likelihood function is

$$\begin{aligned}\ln L(\mathbf{w}') &= \sum_{i=1}^{n} y_i \ln f_{\mathbf{w}'}(\mathbf{x}_i) + (1-y_i)\ln(1-f_{\mathbf{w}'}(\mathbf{x}_i)) \\ &= \sum_{i=1}^{n} y_i(\mathbf{w}'\cdot\mathbf{x}_i') - \ln(1+e^{\mathbf{w}'\cdot\mathbf{x}_i'}).\end{aligned} \tag{15}$$

When instances from each class are linearly separable (which is likely to occur with clustering), coefficients may grow without bound to maximize the log-likelihood function. This can be remedied by adding the squared ℓ_2-norm of the coefficients to the negative log-likelihood function to ensure a minimization problem with a finite solution. The regularized optimization is

$$\arg\min_{\mathbf{w}'} - \ln L(\mathbf{w}') + \frac{\tau}{2}\|\mathbf{w}\|_2^2. \tag{16}$$

This regularization reduces the magnitude of the coefficients, but in general, all of the features are associated with non-zero coefficients. Due to the geometry of the equivalent constraints, replacing the ℓ_2-norm with an ℓ_1-norm, i.e., Lasso [9], can yield solutions where many coefficients are exactly zero. The Lasso-regularized optimization is

$$\arg\min_{\mathbf{w}'} - \ln L(\mathbf{w}') + \gamma\|\mathbf{w}\|_1. \tag{17}$$

The choice of γ affects the number of features with non-zero coefficients, as a relatively large value of γ will yield a solution with few non-zero coefficients. The features with non-zero coefficients can be considered selected features. An entire suite of candidate feature subsets can be found by sweeping the value of γ (known as the regularization path). This regularization path can be computed as efficiently as solving for the optimization for a single choice of γ [68]. The candidate feature subsets $\mathcal{S}_1, \ldots, \mathcal{S}_J$ are formed at the sequence of change-points $\gamma_1, \ldots, \gamma_J$ (parameter values where a new feature assumes a non-zero coefficient or a coefficient becomes zero) along the regularization path. A particular subset can be selected by using a model order selection criterion, which we discuss in Section 3.3.2.

It is straightforward to combine the positive constraints with regularized logistic regression. The positively-constrained version of logistic regression with Lasso regularization is

$$\arg\min_{\mathbf{w}': w_1 \geq 0, \ldots, w_N \geq 0} - \ln L(\mathbf{w}') + \gamma \sum_{i=1}^{N} w_i. \tag{18}$$

Instead of Lasso, an alternative is to use Elastic-Net regularization [10], which uses a combination of both the ℓ_1-norm and ℓ_2-norm, and is more stable for a large number of correlated features [10]. The optimization for Elastic-Net regularization is

$$\arg\min_{\mathbf{w}': w_1 \geq 0, \ldots, w_N \geq 0} - \ln L(\mathbf{w}') + \gamma \sum_{i=1}^{N}\left(\mu w_i + \tfrac{1-\mu}{2} w_i{}^2\right). \tag{19}$$

Again, candidate feature subsets $\mathcal{S}_1, \ldots, \mathcal{S}_J$ correspond to features with non-zero coefficients for the sequence of parameter values $\gamma_1, \ldots, \gamma_J$ at which a new feature assumes a non-zero coefficient or an existing coefficient becomes zero along the regularization path.

Both optimizations (Eqs. (18) and (19)) are convex and smooth—the positivity constraints avoid the challenges of optimization with the non-smooth ℓ_1-norm regularization. Although a generic optimization can be used to solve this problem, in order to scale to a large number of features, a solver specialized for Lasso and Elastic Net should be used to compute the regularization path. To further increase the scalability, implementations[2] that apply rules to filter out many of the irrelevant features [71] are necessary.

3.3.2 Feature Subset Selection via BIC

To select a particular feature subset from among a candidate set, we form a logistic regression model for each feature subset and rank them using the Bayesian information criterion [13]. Given the candidate feature subsets $\mathcal{S}_1, \ldots, \mathcal{S}_J$ for a particular cluster and their corresponding cardinalities $|\mathcal{S}_1|, \ldots, |\mathcal{S}_J|$, the optimal feature subset is $\mathcal{S}_{j^\star}$, where $j^\star$ is chosen as

$$\begin{aligned} j^\star &= \arg\min_{j \in \{1,\ldots,J\}} - \ln L(\tilde{\mathbf{w}}_j') + |\mathcal{S}_j| \ln \sqrt{n}, \\ \tilde{\mathbf{w}}_j' &= \arg\min_{\mathbf{w}': w_i = 0,\, i \in \overline{\mathcal{S}}_j} - \ln L(\mathbf{w}') + \frac{\tau}{2}\|\mathbf{w}\|_2^2, \end{aligned} \tag{20}$$

and $\overline{\mathcal{S}}_j$ denotes the set of features not included in the jth features subset.

It is noteworthy that new coefficients are estimated for the feature subsets generated by Lasso or Elastic Net (a process known as debiasing [72], [73]), which is important since sparsity-inducing regularizations cause the non-zero coefficients to be biased towards zero. This bias would otherwise affect comparisons between feature subsets chosen with different amounts of regularization. We also note that the choice of BIC instead of the AIC is motivated by the theoretical consistency of BIC [67], since we desire an interpretable number of features that should be stable across changes in sample size. Yet, in practical situations, the candidate feature sets may need to be limited in size: a user may want at most a dozen features to describe a cluster. While a limit may decrease the predictive performance, it will also decrease the computation in terms of the number of candidate feature sets that need to be evaluation.

4 Experimental Results

In this section, we empirically evaluate the PDC framework on publicly available datasets. First, we evaluate the performance of various feature selection approaches for selecting predictive and descriptive feature labels. We use ground-truth class labels and evaluate how well the selected descriptive features can predict class membership while comparing against baseline linear classifiers, which use all features. Second, we evaluate descriptive clustering in terms of the information content of the descriptive features and the clustering itself.

4.1 Datasets

We use the 20-Newsgroup dataset [74], a compilation of online reviews [75], [76], the Reuters-21578 Distribution 1.0 newswire articles,[3] ingredient lists from Yummly's recipe

[2] We use LIBLINEAR [69] for logistic regression with ℓ_2-norm regularization, and GLMNET [70] for logistic regression with Lasso and Elastic-Net regularization.

[3] The corpus is on the UCI Machine Learning Repository [77].

dataset,[4] the NSF research award abstracts 1990-2003 data set [77], and news articles provided[5] by Antonio Gulli. The 20-Newsgroup dataset[6] consists of Usenet posts divided among 20 topics, with some of the topics categorized together (*baseball* and *hockey* are both under *rec*). The reviews cover 5 topics[7]: *movies*, *books*, *dvds*, *electronics*, and *kitchen*; for each topic there are 2000 reviews with 1000 positive sentiment and 1000 negative sentiment reviews. Like previous usage of this datatset [53], we explore both topic and sentiment classification within a given topic to compare the feature selection algorithms, but for clustering we ignore the sentiment and only consider mixtures of different topics. For the Reuters dataset, we use the 8 largest categories of single-category articles.[8] We use the training portion of the recipe dataset that is categorized by cuisine.[9] The NSF abstracts are classified by organization, we keep only abstracts from the 35 most common organizations.[10] For AG's news dataset, we use a subset of the 4 largest categories [79].[11] The list of subsets and datasets are given in Table 2. Stop words[12] are removed (excepting the 20-News subsets and recipes). For each subset/dataset, features that appear in only one instance are removed.

4.2 Implementation Details

For representing text documents, we use a bag-of-words representation. Each feature dimension is weighted by the logarithm of the inverse occurrence rate, the standard term-frequency inverse-document frequency (TF-IDF), and instances are normalized to have unit-norm. For clustering text documents, we use spectral clustering applied to the similarity matrix implicitly formed from the cosine similarity between the TF-IDF vectors [80], and the spectral clustering algorithm of Ng et al. [81] is applied to the similarity matrix. All instances are used by the clustering algorithm, but a subset of instances are used as training instances for selecting feature subsets and evaluating the predictive features. The number of clusters is varied between 2 and 26.

A logistic regression model is trained for each feature subset, with regularization of $\tau = \frac{1}{10}$ to ensure bounded coefficients, and the feature subset with minimal BIC is selected as in Eq. (20). To transform the selected logistic regression model to a binary classifier, the probability estimates are thresholded, with the threshold that maximizes the F_1-score (the harmonic mean of precision and recall) in the training data [82].

4.3 Evaluation

For evaluating the classifier and cluster predictions, we use the F_1-score per cluster/class and summarize the results by using the macro-average of the F_1 score. For computing the correspondence between ground-truth topics and known classes, we must take into consideration that an instance may be assigned to multiple clusters, or not assigned to any cluster. Each instance is given equal weight, and when an instance is assigned to multiple clusters, its weight is divided equally among its cluster assignments. Unassigned instances are considered to be grouped together into an additional outlier cluster. This ensures a valid contingency table for computing cluster correspondence metrics—in particular, normalized mutual information (the mutual information between two discrete variables divided by the maximum entropy of either variable [83])—in order to compare the correspondence between ground-truth categories and the classifier-based clusters assignments. We also use the normalized mutual information to compare the original cluster assignments to the ground-truth topic categories.

4.4 Comparison of Feature Selection Performance

The PDC framework is based on selecting an interpretable set of predictive features. For the purpose of interpretability we limit the number of features and constrain them to be positively correlated with the class/cluster of interest. We examine how much the positivity constraints limit the classification performance and how well the different feature selection approaches perform.

We compare a range of feature selection algorithms with and without positivity constraints. These include ranking methods that simply select top-ranked features for different criteria: weighted log-likelihood ratio (WLLR) [53] (which is only defined for selecting positively correlated features), mutual information (MI), and the chi-squared statistic (CHI^2); and forward-selection algorithms using information theoretic criteria [5]: JMI [57], [58], MRMR [56], and CMIM [55]. To estimate these quantities, co-occurrence statistics are computed after the input feature vectors are transformed to binary vectors (removing any information about counts, weightings, and instance normalization). We limit the forward-selection algorithms to 250 features with the highest mutual information for each class/cluster. For regularization-based feature selection we compare Lasso and Elastic Net with positivity constraints. For the latter, the ℓ_1-ℓ_2-norm trade-off is fixed at $\mu = 0.2$ in Eq. (19). For each feature selection algorithm, training consists of selecting feature subsets of varying size (up to a maximum of 50 features per class/cluster), then the selection process discussed in Section 4.2 is applied to select the number of features for each algorithm.

[4] The dataset is at https://www.kaggle.com/c/whats-cooking.

[5] The dataset is available at http://www.di.unipi.it/~gulli/AG_corpus_of_news_articles.html.

[6] For the 20 Newsgroup corpus, we use two versions: a preprocessed MATLAB/OCTAVE version of the bag-of-words representation provided by Jason Rennie, http://qwone.com/~jason/20Newsgroups/that we divide into various subsets, and a tokenized "bydate" training-testing split of the full dataset http://ana.cachopo.org/datasets-for-single-label-text-categorization [78].

[7] Movie reviews are taken from the polarity dataset v2.0 available at http://www.cs.cornell.edu/people/pabo/movie-review-data/. Book and product reviews crawled from the online retailer Amazon are available in XML format at http://www.cs.jhu.edu/~mdredze/datasets/sentiment/index2.html. For the reviews, apostrophes and numbers are removed, and the text is tokenized by whitespace and punctuation.

[8] A tokenized version of this subset with a training-testing split [78] is available at http://ana.cachopo.org/datasets-for-single-label-text-categorization.

[9] The Yummly recipes are provided as lists of ingredients. Each ingredient is a character string that is processed by splitting the string into tokens by whitespace, numerals, hyphens, periods, commas, and parentheses. Features are formed from all subsequences of consecutive tokens, and additional features are appended by mapping together possessive forms with missing apostrophes and by combining features with both singular and plural forms.

[10] We retain the NSF abstracts that have an award number listed in the abstract file, are referenced in the list files (`idnsfid.txt`), and are longer than one line and have at least 17 tokens (after removing stop words), where tokenization is based on whitespace and punctuation (apostrophes and internal hyphens are retained).

[11] https://github.com/mhjabreel/CharCNN/tree/master/data/ag_news_csv.

[12] We use a list of 571 English stop words, http://members.unine.ch/jacques.savoy/clef/.

TABLE 2
Dataset/Subset Profiles: Classes, Instances (n), and Features (N)

	Classes	n	N
Reviews (+/-):			
movie	2	2000	24093
kitchen	2	2000	4665
dvd	2	2000	10423
electronics	2	2000	5167
books	2	2000	10551
Reviews (topic):			
movie, kitchen (m k)	2	4000	25543
movie, books (m b)	2	4000	27436
movie, electronics (m e)	2	4000	25778
kitchen, electronics (k e)	2	4000	7786
dvd, electronics (d e)	2	4000	13048
movie, books, electronics, kitchen (m b e k)	4	8000	30079
dvd, movie, electronics, kitchen (d m e k)	4	8000	29552
dvd, books, movie, kitchen (d b m k)	4	8000	31101
dvd, books, electronics, movie (d b e m)	4	8000	31299
dvd, books, electronics, movie, kitchen (d b e m k)	5	10000	32375
20-News (subsets):			
med guns	2	1896	15204
autos space hardware	3	2951	16071
rec: autos, motorcycles, baseball, hockey	4	3968	18728
sci: crypto, electronics, med, space	4	3945	22198
mix: ms-windows, forsale, baseball, space, politics.misc	5	4677	23059
comp: graphics, ms-windows, pc.hardware mac.hardware, x-windows	5	4852	20382
mix2: ms-windows, autos, baseball, med, space	5	4913	24457
sci comp: sci.* comp.*	9	8797	33501
sci comp rec: sci.* comp.* rec.*	13	12765	42049
20-News (all)	20	18820	41747
Reuters	8	7674	10250
Recipe	20	39774	10506
NSF	35	125730	119773
AG's news	4	127600	39763

To give an upper bound on classification performance using unconstrained classifiers, we use the LIBLINEAR package to train linear classifiers using different loss functions (ℓ_1-loss or logistic loss) with ℓ_2-norm regularization. The regularization trade-off parameter is chosen by LIBLINEAR's built-in cross-validation. We note that these classifiers do not yield readily interpretable feature subsets.

On datasets without predefined training and testing splits, 10 Monte Carlo runs of training and testing splits are used to assess feature selection performance. The results are included in Table 3.

We note the classifiers that used feature selection are all outperformed by the baseline linear classifiers. In this classification task, feature selection does not improve prediction performance. Second, we note that the gap between the performance of the baseline classifiers and cardinality-constrained methods is smaller than the gap between the performance of the positively-constrained methods. Nonetheless, we maintain these constraints for interpretability such that the features are positively correlated with each class.

We compare the scaling performance of the top three performing feature selection algorithms with the linear classifiers across different training set sizes for the AG's news dataset (the largest dataset). Fig. 2 shows that the vocabulary size of the training set grows sub-linearly with the size of the corpus. The figure also shows the classification performance, which stabilizes at around 10000 training instances, and the computation time.[13] The methods all scale linearly with the number of training instances with the ℓ_1-loss support vector machines having the longest running times for training set sizes above 10000, whereas logistic regression and the CMIM-based feature selection algorithm are the fastest. Based on the marginal difference in performance among these three feature selection algorithms, and the efficiency of the CMIM-based algorithm, we choose CMIM+ for further comparisons.

4.5 Predictive and Descriptive Clustering

We apply the PDC framework to identify and label clusters of documents within each of the mixture-of-topics datasets listed in Table 2. First, we compare the cluster prediction performance using the CMIM algorithm with positivity constraints versus an existing topic labeling approach [18], and qualitatively examine the selected features. Second, we evaluate the automatic selection of the number of clusters, testing whether the number of clusters correlates with the ground-truth number of topics or whether the number of clusters maximizes the information content.

4.5.1 Comparison with Existing Topic Labeling Approach

We compare against the method proposed by Mei, Shen, and Zhai [18] to select features for describing multinomial topic models (we refer to this method as MSZ). For each topic, the MSZ method selects features that—when used as conditioning variables—approximate the multinomial distribution of features within the topic and also discriminate the topic from other topics. The score is further adjusted to remove redundancy between selected features [50]. The method requires parameter choices for controlling the trade-off between approximation and discrimination (μ) and for lowering the score of redundant features (λ). The number of features to select for each topic is left to the user.

We use the MSZ scoring to generate candidate feature subsets with up to 50 features per cluster and follow the same procedure used in the previous section to select the best performing subset using BIC. The main drawback of the MSZ method is that it estimates a multinomial distribution for each candidate feature, which has a computational complexity of $\mathcal{O}(N^2)$. This is further exacerbated with redundancy elimination, as each candidate feature must be compared to features already included in the set by the divergence between their multinomial distributions. For scalability, we first find features (from those with at least 5 occurrences) with the highest score for each target cluster and keep only the top 250 when performing the redundancy elimination (the same approach used for the forward selection models). We use the parameter choices of $\mu = 1$ and

[13] Time for both feature subset generation and selection is logged in MATLAB on Mac OS X with a 2.8 GHz Intel Core i7 and 16 GB RAM.

TABLE 3
Comparison of Classification Performance of Feature Selection Algorithms versus Baseline Linear Classifiers

		Reviews		20-News	20-News	Reuters	Recipe	NSF	AG's	Ave. rank	
		+/-	topic	(subsets)	(all)				news		
	$\lvert\mathcal{S}\rvert$	1000	1000	1000	*11293	*5485	10000	10000	*120000	group	overall
Positive constraints:											
WLLR+	26.7	0.670	0.791	0.725	0.661	0.836	0.513	0.518	0.749	6.4	15.1
MI+	26.9	0.671	0.791	0.723	0.665	0.833	0.519	0.521	0.751	5.9	14.5
JMI+	26.6	0.671	0.791	0.727	0.668	0.831	0.520	0.525	0.753	5.4	13.1
CHI^2+	26.4	0.671	0.793	0.725	0.669	0.834	0.526	0.524	0.750	5.3	13.3
MRMR+	26.2	0.668	0.792	0.731	0.674	0.824	0.544	0.530	0.758	4.3	11.0
CMIM+	27.0	0.669	0.795	0.736	0.682	0.827	0.564	0.523	0.764	3.5	9.2
Elastic Net+	28.1	0.673	0.802	0.733	0.683	0.841	0.536	0.536	0.754	2.7	8.7
Lasso+	27.7	0.671	0.803	0.740	0.689	0.840	0.547	0.515	0.758	2.6	8.5
Unconstrained:											
MI	28.3	0.728	0.799	0.734	0.667	0.853	0.516	0.523	0.751	5.7	11.0
JMI	28.1	0.729	0.801	0.737	0.667	0.838	0.519	0.527	0.752	5.4	10.4
CHI^2	27.6	0.730	0.803	0.733	0.668	0.838	0.527	0.525	0.753	5.3	10.1
MRMR	27.9	0.729	0.808	0.741	0.676	0.843	0.548	0.532	0.756	3.6	6.7
Lasso	29.0	0.734	0.825	0.748	0.692	0.844	0.532	0.499	0.763	2.9	6.4
CMIM	28.4	0.729	0.810	0.746	0.678	0.845	0.562	0.525	0.764	2.7	5.7
Elastic Net	29.3	0.732	0.818	0.741	0.681	0.857	0.538	0.537	0.761	2.4	5.4
Baseline linear classifiers:											
L-logit R-ℓ_2	—	0.782	0.902	0.789	0.734	0.858	0.598	0.550	0.911	1.8	2.7
L-ℓ_1 R-ℓ_2	—	0.776	0.907	0.828	0.777	0.888	0.607	0.596	0.915	1.2	1.2

*Columns correspond to the average number of features ($\lvert\mathcal{S}\rvert$), macro-averaged F_1-score for different datasets (training set size is listed and * indicates a pre-defined training-testing split), and average rank based on the F_1-score within each algorithm group and overall.*

$\lambda = 0.5$. We also test without redundancy elimination ($\lambda = 1$) and with a higher level ($\lambda = 0.2$), but the performance is worse for both cases.

We assess how well the selected features subsets can predict cluster membership in terms of the F_1 score, and normalized mutual information is used to assess the correspondence between the classifier predictions and the cluster assignments. The results for CMIM+ and MSZ are detailed in Table 4.

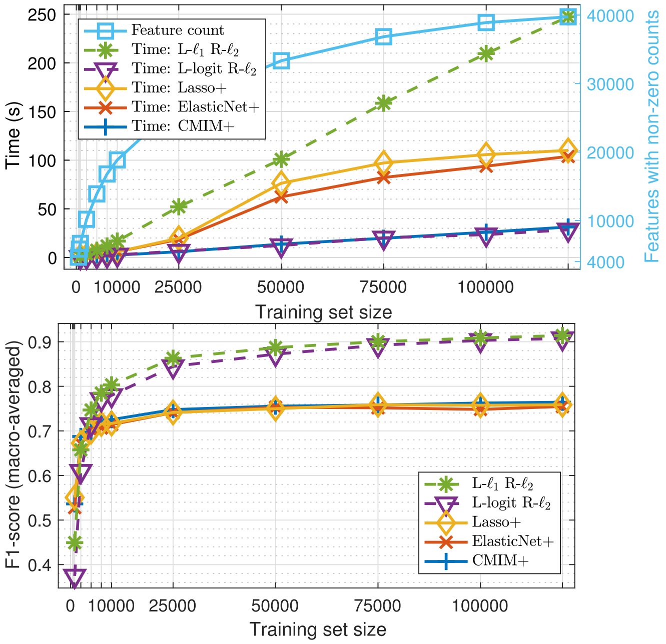


Fig. 2. Performance (macro-averaged F_1-score and running time) across different training set sizes for the AG's News dataset.

TABLE 4
Cluster Classification Performance: $\mathrm{F}_1(Y, \hat{Y})$—Macro-Averaged F_1 and $\mathrm{NMI}(Y, \hat{Y})$—Normalized Mutual Information between Classifier Output and Clusters

		$\mathrm{F}_1(Y, \hat{Y})$		$\mathrm{NMI}(Y, \hat{Y})$	
Number of clusters C		CMIM+	MSZ	CMIM+	MSZ
Reviews (topic):					
m k	2	**0.92**	0.90	**0.64**	**0.64**
m b	6	**0.54**	0.52	0.26	**0.27**
m e	2	**0.94**	**0.94**	0.66	**0.67**
k e	9	**0.64**	0.61	**0.35**	0.33
d e	8	**0.64**	0.62	**0.37**	**0.37**
m b e k	4	**0.80**	0.79	**0.52**	0.49
d m e k	4	**0.74**	0.70	**0.43**	0.40
d b m k	4	**0.73**	**0.73**	**0.40**	**0.40**
d b e m	5	0.63	**0.65**	0.33	**0.36**
d b e m k	5	**0.72**	0.70	**0.43**	0.42
20-News (subsets):					
med guns	8	**0.66**	0.58	**0.37**	0.35
autos space hardware	8	**0.62**	0.54	**0.37**	0.16
rec	8	**0.59**	0.50	**0.36**	0.16
sci	10	**0.63**	0.56	**0.35**	0.16
mix	8	**0.67**	0.56	**0.41**	0.37
comp	11	**0.50**	0.42	**0.25**	0.07
mix2	8	**0.72**	0.67	**0.45**	0.21
sci comp	8	**0.67**	0.56	**0.35**	0.09
sci comp rec	9	**0.65**	0.54	**0.39**	0.16
20-News (all)	15	**0.74**	0.72	**0.56**	0.54
Reuters	13	0.74	**0.76**	0.60	**0.66**
Recipe	21	**0.68**	0.67	**0.49**	0.48
NSF	14	**0.75**	0.73	**0.54**	0.52
AG's news	26	**0.74**	**0.74**	**0.55**	**0.55**

Proposed approach (CMIM+) compared to (MSZ) [18].

TABLE 5
Selected Features, Classifier-Topic Contingency Table, and Cluster-Topic Contingency Table for Descriptive Clustering on the *DVD*, *Movie*, *Electronics*, *Kitchen* Reviews Dataset

(A)	Cluster 1	Cluster 2	Cluster 3	Cluster 4	
dvd	164	**986**	596	23	
movie	1	19	**1760**	0	
electronics	487	45	4	**1196**	
kitchen	**1626**	9	3	81	
(CMIM+)					
	amazon, baking, blender, clean, coffee, cooking, counter, cream, cuisinart, dishwasher, easy, grill, heat, item, kitchen, kitchenaid, knives, le, maker, oven, pan, plastic, pot, price, product, products, sheets, shipping, stainless	concert, documentary, dvd, dvds, edition, episode, fan, fantastic, loved, movie, videos, watch, watched, workout	action, actors, character, characters, director, film, james, life, movie, people, played, plays, plot, real, scene, story, world	adapter, batteries, battery, cable, headphones, ipod, laptop, logitech, mp, pc, plug, software, sound, speakers, support, unit, usb, volume, wireless	
dvd	37.7	**435.7**	858.2	66.5	371
movie	5.7	36.2	**1671.7**	14.5	52
electronics	**810.8**	57.3	13.6	413.4	437
kitchen	86.3	9.0	5.3	**1144.3**	474
(MSZ)					
	amazon, bought, buy, clean, cleaning, coffee, cooking, counter, cuisinart, cup, dishwasher, easy, item, kitchen, large, lid, medium, months, oven, pan, plastic, pot, price, product, products, purchased, replacement, stainless, warranty, water	dvd, movie, watch	character, characters, comic, director, film, films, finds, played, plays, relationship, scene, sense, story, threatening, world, young	batteries, battery, bought, headphones, ipod, mp, pc, sound, speakers, unit, volume	
dvd	29.1	**486.7**	758.4	88.7	406
movie	2.0	71.3	**1641.8**	13.8	51
electronics	**515.3**	53.7	17.0	575.0	571
kitchen	75.0	3.5	3.0	**1136.5**	501

The number of clusters was chosen automatically to maximize the prediction of the original feature occurrences in the training set. (A) *Contingency table between topics and original clusters. (CMIM+) Features selected using CMIM with positivity constraints, and classifier-topic contingency table. (MSZ) Features selected using the MSZ approach [18]. (Features common between multiple clusters are underlined. The number of unassigned instances per topic are listed in the last column.)*

Across the datasets, CMIM+ consistently outperforms MSZ in terms of both F_1-score and normalized mutual information (one-tailed sign-test with significance threshold of 0.1 and Bonferroni's correction for multiple testing). As mentioned, the MSZ approach is more computationally demanding: per cluster, the average computation time for feature selection for the MSZ approach is 7.09 s, which is significantly slower than the 0.34 s for CMIM+.

An example of the descriptive clustering in terms of the selected features, classifier-topic contingency table, and cluster-topic contingency table is shown in Table 5.

From the contingency table, there is a clear matching between the topics and the clusters, and the features chosen by CMIM+ corroborate this. Assuming a user is interested in one of the original topics, the selected features appear to be sufficient to guide the user to a specific cluster.

4.5.2 *Selecting the Number of Clusters*

We turn our attention to evaluating the automatic selection of the number of clusters in the PDC framework. Ideally, the selected number of clusters would maximize the information carried by the clustering and classifiers about the topics. To test this, we run an experiment wherein we train a classifier for every possible number of clusters and evaluate the information between the classifier predictions and the ground-truth topics. We find that the selected number of clusters often maximizes the amount of information carried about the original clusters and that the selected number varies proportionally to the ideal number.

Three examples with varying number of ground-truth topics are show in Fig. 3. On these examples, the automatically selected number (based on Eq. (9)) of clusters adapts to the data, matching the ground-truth number of clusters on the first two examples and choosing a reasonable but smaller number of clusters on the full 20 Newsgroup dataset. The optimal number of clusters may be different from the number of ground-truth categories, since the categories themselves may be too coarse or too fine.

As a surrogate baseline, we use an oracle to select C that maximizes the normalized mutual information between the topics and the clusters and another oracle that maximizes the normalized mutual information between the cluster and the classifier output. The performance of the selection process (based on Eq. (9)) compared to the two oracles is in Table 6.

We test the hypotheses that the selected number of clusters using either Eq. (9) or Eq. (12) are positively correlated with the number selected by the oracles, and whether the optimized numbers for the two measures are correlated. We use a significance threshold of 0.05 with Bonferroni's correction, and find that the rank correlation is significant for both oracles and measures:

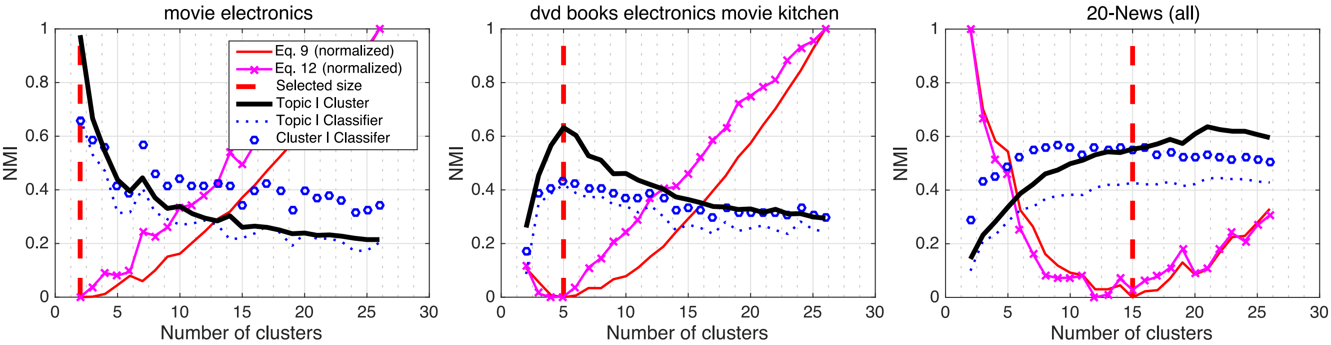


Fig. 3. Automatic selection of the number of clusters assessed in terms of the normalized mutual information between the topic categories and clusters; topics and classifier output; and clusters and classifier output. The selected number of clusters (2, 5, and 15) are chosen to minimize the feature prediction error Eq. (9), which correlates with Eq. (12)—the AIC for a multinomial model of features occurring in at least 5 instances.

	Rank correlation	Corrected p-value
Eq. (9) ∣ Oracle: NMI(Topic, Cluster)	0.5943	**0.0055**
Eq. (9) ∣ Oracle: NMI(Cluster, Classifier)	0.6123	**0.0037**
Eq. (12) ∣ Oracle: NMI(Topic, Cluster)	0.8092	$\mathbf{4.21\times10^{-6}}$
Eq. (12) ∣ Oracle: NMI(Cluster, Classifier)	0.5932	**0.0056**
Eq. (9) ∣ Eq. (12)	0.8469	$\mathbf{4.52\times10^{-6}}$

Computing AIC (Eq. (12)) across the clusterings of different size is much more efficient than the binary loss used Eq. (9). We report some example running time in Table 7. Running times and feature sets for all datasets are included in the supplementary material (available at http://ieeeexplore.ieee.org).

5 Discussion

The self-tuned nature of the proposed descriptive document clustering enables it to select both the number of clusters and the number of features used to describe each cluster; nonetheless, a user must choose a suitable range that is both computationally feasible and appropriate for the datasets. In our tests, the upper limit was only met once on the largest dataset (AG's news). In such a case, a user may wish to increase the range to find a more optimal clustering.

In addition to text, the proposed descriptive clustering can be applied to any data with sparse count-valued features. Example experiments are included in the supplementary material (available at http://ieeeexplore.ieee.org).

TABLE 6
Normalized Mutual Information (NMI) between True Topic Categories and Clusters, and between Clusters and Classifier Predictions, for Different Numbers of Clusters

	NMI(Topic, Cluster)		NMI(Cluster, Classifier)	
	Oracle	C^{*}	Oracle	C^{*}
Reviews (topic):				
m k	0.97 (2)	0.97 (2)	0.63 (2)	0.63 (2)
m b	0.83 (2)	0.31 (6)	0.51 (2)	0.25 (6)
m e	0.98 (2)	0.98 (2)	0.66 (2)	0.66 (2)
k e	0.41 (2)	0.14 (9)	0.35 (17)	0.33 (9)
d e	0.54 (3)	0.25 (8)	0.39 (5)	0.34 (8)
m b e k	0.73 (4)	0.73 (4)	0.53 (4)	0.53 (4)
d m e k	0.60 (4)	0.60 (4)	0.43 (6)	0.43 (4)
d b m k	0.64 (4)	0.64 (4)	0.40 (5)	0.39 (4)
d b e m	0.56 (4)	0.53 (5)	0.37 (4)	0.33 (5)
d b e m k	0.63 (5)	0.63 (5)	0.43 (5)	0.43 (5)
20-News (subsets):				
med guns	0.48 (2)	0.23 (8)	0.40 (11)	0.38 (8)
autos space hardware	0.63 (3)	0.32 (8)	0.41 (3)	0.34 (8)
rec	0.54 (4)	0.43 (8)	0.36 (6)	0.34 (8)
sci	0.40 (5)	0.33 (10)	0.35 (11)	0.35 (10)
mix	0.61 (5)	0.51 (8)	0.43 (6)	0.42 (8)
comp	0.32 (5)	0.20 (11)	0.29 (7)	0.22 (11)
mix2	0.69 (5)	0.54 (8)	0.47 (9)	0.47 (8)
sci comp	0.41 (8)	0.41 (8)	0.34 (8)	0.34 (8)
sci comp rec	0.54 (12)	0.47 (9)	0.41 (7)	0.39 (9)
20-News (all)	0.64 (21)	0.55 (15)	0.57 (9)	0.55 (15)
Reuters	0.62 (4)	0.46 (13)	0.64 (4)	0.50 (13)
Recipe	0.28 (9)	0.26 (21)	0.51 (5)	0.42 (21)
NSF	0.44 (23)	0.39 (14)	0.51 (8)	0.49 (14)
AG's news	0.44 (5)	0.22 (26)	0.51 (23)	0.49 (26)

The proposed prediction-based selection C^{*} (based on Eq. 9) is compared to an oracle that maximizes the NMI for each case. The selected number of clusters is in parentheses.

TABLE 7
Computation Time for Evaluating Clusterings of Different Sizes

	Reuters	Recipe	NSF	AG's news
Eq. (9)	180.1 s	219.0 s	968.4 s	1832.3 s
Eq. (12)	0.5 s	4.3 s	34.0 s	13.9 s

6 Conclusion

We posed descriptive clustering as two coupled prediction tasks: 1) choosing a clustering that is predictive of the features and 2) predicting the cluster assignment from a subset of features. Using predictive performance as the objective criterion, the parameters of descriptive clustering—the number of clusters and the number of features per cluster—are chosen by model selection. With the resulting solution, each cluster is described by a minimal subset of features necessary to predict whether an instance belongs to the cluster. Our hypothesis is that a user will also be able to predict the cluster membership of documents using the descriptive features selected by the algorithm. Given some relevancy requirements, a user can then quickly locate clusters likely to contain relevant documents.

We evaluated this self-tuned approach on datasets with count-valued features. For feature selection we used both information theoretic feature selection and linear classifiers trained by logistic regression with sparsity-inducing regularizations and positivity constraints. The results showed that these feature selection approaches perform accurately and yield feature subsets that are indicative of the cluster content. Furthermore, the predictive approach selected a meaningful terms of number of clusters and number of features per cluster. This objective self-tuning distinguishes the proposed framework from previous descriptive clustering approaches that have not addressed parameter selection and have relied on subjective criterion for assessing descriptors.

Currently, we have begun using the PDC framework to generate descriptive clustering to help users screen large collections of abstracts to support the development of systematic reviews, especially in domains such as public health.[14] For future work, we plan to investigate PDC using more complex features including multi-word expressions, named entities, and clusters of features themselves.

Acknowledgments

This research was partially supported by Medical Research Council grants MR/L01078X/1 and MR/N015665/1.

References

[1] G. Salton, *The SMART Retrieval System–Experiments in Automatic Document Processing*. Upper Saddle River, New Jersey, USA: Prentice-Hall, 1971.

[2] D. R. Cutting, D. R. Karger, J. O. Pedersen, and J. W. Tukey, "Scatter/gather: A cluster-based approach to browsing large document collections," in *Proc. Int. ACM SIGIR Conf. Res. Dev. Inf. Retrieval*, 1992, pp. 318–329.

[14] http://www.nactem.ac.uk/robotanalyst

[3] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artif. Intell.*, vol. 97, no. 1, pp. 273–324, 1997.

[4] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, no. 3, pp. 1157–1182, 2003.

[5] G. Brown, A. Pocock, M.-J. Zhao, and M. Luján, "Conditional likelihood maximisation: A unifying framework for information theoretic feature selection," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 27–66, 2012.

[6] P. S. Bradley and O. L. Mangasarian, "Feature selection via concave minimization and support vector machines," in *Proc. Int. Conf. Mach. Learn.*, 1998, pp. 82–90.

[7] J. Zhu, S. Rosset, R. Tibshirani, and T. J. Hastie, "1-norm support vector machines," *Adv. Neural Inf. Process. Syst.*, vol. 16, 2004, pp. 49–56.

[8] K. Koh, S.-J. Kim, and S. Boyd, "An interior-point method for large-scale l1-regularized logistic regression," *J. Mach. Learn. Res.*, vol. 8, no. 7, pp. 1519–1555, 2007.

[9] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Stat. Soc. Ser. B*, vol. 58, no. 1, pp. 267–288, 1996.

[10] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J. Roy. Stat. Soc. Ser. B*, vol. 67, no. 2, pp. 301–320, 2005.

[11] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you?: Explaining the predictions of any classifier," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 1135–1144.

[12] J. Chorowski and J. M. Zurada, "Learning understandable neural networks with nonnegative weight constraints," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 1, pp. 62–69, Jan. 2015.

[13] G. Schwarz, "Estimating the dimension of a model," *Ann. Stat.*, vol. 6, no. 2, pp. 461–464, 1978.

[14] D. Weiss, "Descriptive clustering as a method for exploring text collections," Ph.D. dissertation, Institute of Computing Science, Poznan University of Technology, Poznań, Poland, 2006.

[15] C. Carpineto, S. Osiński, G. Romano, and D. Weiss, "A survey of web clustering engines," *ACM Comput. Surv.*, vol. 41, no. 3, 2009, Art. no. 17.

[16] T. Mu, J. Y. Goulermas, I. Korkontzelos, and S. Ananiadou, "Descriptive document clustering via discriminant learning in a co-embedded space of multilevel similarities," *J. Assoc. Inf. Sci. Technol.*, vol. 67, no. 1, pp. 106–133, 2016.

[17] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Trans. Neural Netw.*, vol. 16, no. 3, pp. 645–678, May 2005.

[18] Q. Mei, X. Shen, and C. Zhai, "Automatic labeling of multinomial topic models," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2007, pp. 490–499.

[19] S. Kaski, T. Honkela, K. Lagus, and T. Kohonen, "Websom–self-organizing maps of document collections," *Neurocomputing*, vol. 21, no. 1, pp. 101–117, 1998.

[20] K. Lagus and S. Kaski, "Keyword selection method for characterizing text document maps," in *Proc. Int. Conf. Artif. Neural Netw.*, 1999, pp. 371–376.

[21] T. Kohonen, S. Kaski, K. Lagus, J. Salojarvi, J. Honkela, V. Paatero, and A. Saarela, "Self organization of a massive document collection," *IEEE Trans. Neural Netw.*, vol. 11, no. 3, pp. 574–585, May 2000.

[22] A. Skupin, "A cartographic approach to visualizing conference abstracts," *IEEE Comput. Graph. Appl.*, vol. 22, no. 1, pp. 50–58, Jan. 2002.

[23] J. H. Lau, K. Grieser, D. Newman, and T. Baldwin, "Automatic labelling of topic models," in *Proc. Annu. Meeting Assoc. Comput. Ling.*, 2011, pp. 1536–1545.

[24] M. A. Hearst and J. O. Pedersen, "Reexamining the cluster hypothesis: Scatter/gather on retrieval results," in *Proc. Int. ACM SIGIR Conf. Res. Dev. Inf. Retrieval*, 1996, pp. 76–84.

[25] N. Aletras, T. Baldwin, J. H. Lau, and M. Stevenson, "Representing topics labels for exploring digital libraries," in *Proc. ACM/IEEE-CS Joint Conf. Digit. Libr.*, 2014, pp. 239–248.

[26] D. R. Cutting, D. R. Karger, and J. O. Pedersen, "Constant interaction-time scatter/gather browsing of very large document collections," in *Proc. Int. ACM SIGIR Conf. Res. Dev. Inf. Retrieval*, 1993, pp. 126–134.

[27] M. A. Hearst, *The Use of Categories and Clusters for Organizing Retrieval Results*. Netherlands: Springer, 1999, pp. 333–374.

[28] S. Dumais, J. Platt, D. Heckerman, and M. Sahami, "Inductive learning algorithms and representations for text categorization," in *Proc. Int. Conf. Inform. Knowl. Manag.*, 1998, pp. 148–155.

[29] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," in *Proc. Int. Conf. Mach. Learn.*, 1997, pp. 412–420.

[30] K. W. Church and P. Hanks, "Word association norms, mutual information, and lexicography," *Comput. Ling.*, vol. 16, no. 1, pp. 22–29, 1990.

[31] F. Pereira, N. Tishby, and L. Lee, "Distributional clustering of English words," in *Proc. Annu. Meeting Assoc. Comput. Ling.*, 1993, pp. 183–190.

[32] R. Bekkerman, R. El-Yaniv, N. Tishby, and Y. Winter, "Distributional word clusters vs. words for text categorization," *J. Mach. Learn. Res.*, vol. 3, no. 3, pp. 1183–1208, 2003.

[33] N. Slonim and N. Tishby, "Document clustering using word clusters via the information bottleneck method," in *Proc. Int. ACM SIGIR Conf. Res. Dev. Inf. Retrieval*, 2000, pp. 208–215.

[34] T. Li, S. Ma, and M. Ogihara, "Document clustering via adaptive subspace iteration," in *Proc. Int. ACM SIGIR Conf. Res. Dev. Inf. Retrieval*, 2004, pp. 218–225.

[35] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and Regression Trees*. Boca Raton, FL, USA: CRC press, 1984.

[36] J. C. French, D. E. Brown, and N.-H. Kim, "A classification approach to boolean query reformulation," *J. Am. Soc. Inf. Sci. Technol.*, vol. 48, no. 8, pp. 694–706, 1997.

[37] K. Kummamuru, R. Lotlikar, S. Roy, K. Singal, and R. Krishnapuram, "A hierarchical monothetic document clustering algorithm for summarization and browsing search results," in *Proc. Int. Conf. World Wide Web*, 2004, pp. 658–665.

[38] O. Zamir and O. Etzioni, "Web document clustering: A feasibility demonstration," in *Proc. Int. ACM SIGIR Conf. Res. Dev. Inf. Retrieval*, 1998, pp. 46–54.

[39] A. Bernardini, C. Carpineto, and M. D'Amico, "Full-subtopic retrieval with keyphrase-based search results clustering," in *Proc. IEEE/WIC/ACM Int. Joint Conf. Web Intell. Intell. Agent Technol.*, 2009, pp. 206–213.

[40] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman, "Indexing by latent semantic analysis," *J. Am. Soc. Inf. Sci. Technol.*, vol. 41, no. 6, pp. 391–407, 1990.

[41] S. Osiński, J. Stefanowski, and D. Weiss, "Lingo: Search results clustering algorithm based on singular value decomposition," in *Proc. Int. IIS Intell. Inf. Process. Web Mining Conf.*, 2004, pp. 359–368.

[42] T. Hofmann, "Probabilistic latent semantic indexing," in *Proc. Int. ACM SIGIR Conf. Res. Dev. Inf. Retrieval*, 1999, pp. 50–57.

[43] W. J. Wilbur, "A thematic analysis of the AIDS literature," in *Proc. Pacific Symp. Biocomputing*, 2001, pp. 386–397.

[44] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.

[45] T. L. Griffiths and M. Steyvers, "Finding scientific topics," *Proc. Natl. Acad. Sci. USA*, vol. 101, no. suppl 1, pp. 5228–5235, 2004.

[46] J.-T. Chien, "Hierarchical theme and topic modeling," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 3, pp. 565–578, Mar. 2016.

[47] U. Scaiella, P. Ferragina, A. Marino, and M. Ciaramita, "Topical clustering of search results," in *Proc. ACM Int. Conf. Web Search Data Mining*, 2012, pp. 223–232.

[48] Princeton University, 2010. [Online]. Available: http://wordnet.princeton.edu

[49] Y.-H. Tseng, "Generic title labeling for clustered documents," *Expert Syst. Appl.*, vol. 37, no. 3, pp. 2247–2254, 2010.

[50] J. Carbonell and J. Goldstein, "The use of MMR, diversity-based reranking for reordering documents and producing summaries," in *Proc. Int. ACM SIGIR Conf. Res. Dev. Inf. Retrieval*, 1998, pp. 335–336.

[51] P. Xie and E. P. Xing, "Integrating document clustering and topic modeling," in *Proc. Conf. Uncertainty Artif. Intell.*, 2013, pp. 694–703.

[52] H. Soleimani and D. J. Miller, "Parsimonious topic models with salient word discovery," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 3, pp. 824–837, Mar. 2015.

[53] S. Dasgupta and V. Ng, "Towards subjectifying text clustering," in *Proc. Int. ACM SIGIR Conf. Res. Dev. Inf. Retrieval*, 2010, pp. 483–490.

[54] D. Koller and M. Sahami, "Toward optimal feature selection," in *Proc. Int. Conf. Mach. Learn.*, 1996, pp. 284–292.

[55] F. Fleuret, "Fast binary feature selection with conditional mutual information," *J. Mach. Learn. Res.*, vol. 5, no. 11, pp. 1531–1555, 2004.

[56] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.

[57] H. H. Yang and J. Moody, "Data visualization and feature selection: New algorithms for nongaussian data," *Adv. Neural Inf. Process. Syst.*, vol. 12, pp. 687–693, 2000.

[58] P. E. Meyer, C. Schretter, and G. Bontempi, "Information-theoretic feature selection in microarray data using variable complementarity," *IEEE J. Sel. Top. Signal Process.*, vol. 2, no. 3, pp. 261–274, Jun. 2008.

[59] S. Perkins, K. Lacker, and J. Theiler, "Grafting: Fast, incremental feature selection by gradient descent in function space," *J. Mach. Learn. Res.*, vol. 3, no. 3, pp. 1333–1356, 2003.

[60] G. Swirszcz, N. Abe, and A. C. Lozano, "Grouped orthogonal matching pursuit for variable selection and prediction," *Adv. Neural Inf. Process. Syst.*, vol. 22, pp. 1150–1158, 2009.

[61] S. Bahmani, B. Raj, and P. T. Boufounos, "Greedy sparsity-constrained optimization," *J. Mach. Learn. Res.*, vol. 14, no. 3, pp. 807–841, 2013.

[62] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for Boltzmann machines," *Cogn. Sci.*, vol. 9, no. 1, pp. 147–169, 1985.

[63] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1*. Cambridge, MA, USA: MIT Press, 1986, pp. 318–362.

[64] P. Baldi, "Autoencoders, unsupervised learning, and deep architectures," in *Proc. JMLR: Workshop Unsupervised Transfer Learn.*, vol. 27, 2012, pp. 37–50.

[65] U. von Luxburg, "A tutorial on spectral clustering," *Stat. Comput.*, vol. 17, no. 4, pp. 395–416, 2007.

[66] H. Akaike, "Information theory and an extension of the maximum likelihood principle," in *Proc. 2nd Int. Symp. Inf. Theory*, 1973, pp. 267–281.

[67] S. Arlot and A. Celisse, "A survey of cross-validation procedures for model selection," *Stat. Surveys*, vol. 4, pp. 40–79, 2010.

[68] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *J. Stat. Softw.*, vol. 33, no. 1, pp. 1–22, 2010.

[69] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *J. Mach. Learn. Res.*, vol. 9, no. 8, pp. 1871–1874, 2008.

[70] J. Qian, T. Hastie, J. Friedman, R. Tibshirani, and N. Simon, "Glmnet for matlab 2013, " 2013. [Online]. Available: http://www.stanford.edu/ hastie/glmnet_matlab

[71] R. Tibshirani, J. Bien, J. Friedman, T. Hastie, N. Simon, J. Taylor, and R. J. Tibshirani, "Strong rules for discarding predictors in lasso-type problems," *J. Roy. Stat. Soc. Ser. B*, vol. 74, no. 2, pp. 245–266, 2012.

[72] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *IEEE J. Sel. Top. Signal Process.*, vol. 1, no. 4, pp. 586–597, Dec. 2007.

[73] M. Tan, I. W. Tsang, and L. Wang, "Towards ultrahigh dimensional feature selection for big data," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1371–1429, 2014.

[74] K. Lang, "Newsweeder: Learning to filter netnews," in *Proc. Int. Conf. Mach. Learn.*, 1995, pp. 331–339.

[75] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," in *Proc. Annu. Meeting Assoc. Comput. Ling.*, 2004, pp. 271–278.

[76] J. Blitzer, M. Dredze, and F. Pereira, "Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification," in *Proc. Annu. Meeting Assoc. Comput. Ling.*, 2007, pp. 440–447.

[77] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[78] A. Cardoso-Cachopo, "Improving methods for single-label text categorization," PhD thesis, Instituto Superior Tecnico, Universidade Tecnica de Lisboa, Lisbon, Portugal, 2007.

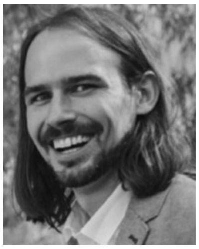
[79] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Proc. 28th Int. Conf. Adv. Neural Inf. Process. Syst.*, 2015, pp. 649–657.

[80] G. Salton, A. Wong, and C.-S. Yang, "A vector space model for automatic indexing," *Commun. ACM*, vol. 18, no. 11, pp. 613–620, 1975.

[81] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," *Adv. Neural Inf. Process. Syst.*, vol. 14, pp. 849–856, 2002.

[82] Z. C. Lipton, C. Elkan, and B. Naryanaswamy, "Optimal thresholding of classifiers to maximize F1 measure," in *Proc. Joint Euro. Conf. Mach. Learn. Knowl. Discovery Databases*, 2014, pp. 225–239.

[83] T. O. Kvålseth, "Entropy and correlation: Some comments," *IEEE Trans. Syst. Man Cybern.*, vol. 17, no. 3, pp. 517–519, May 1987.



Austin J. Brockmeier (S'05-M'15) received the BS degree in computer engineering from the University of Nebraska–Lincoln (Omaha campus) in 2009 and the PhD degree in electrical and computer engineering from the University of Florida, in 2014. He is currently a research fellow in the School of Computer Science, University of Manchester. His research interests include machine learning and signal processing with applications to biomedicine and text mining. He is a member of the IEEE.

Tingting Mu (M'05) received the BEng degree in electronic engineering and information science from the University of Science and Technology of China, in 2004 and the PhD degree in electrical engineering and electronics from the University of Liverpool, in 2008. She is currently a lecturer in the School of Computer Science, University of Manchester. Her research interests include machine learning, mathematical modeling and optimization, with applications to vision and language understanding, text mining, and bioinformatics. She is a member of the IEEE.

Sophia Ananiadou is a professor of computer science with the University of Manchester and the director of the UK National Centre for Text Mining (NaCTeM). Her main research contributions have been in the areas of event extraction, named entity recognition, word sense disambiguation, terminology management, and development of text mining platforms and large-scale terminological resources. Research in these areas guided and informed the development of several scalable and advanced search systems, resources and tools for biology, medicine, public health, biodiversity, etc., which are currently available from the NaCTeM portal. Her h-index is 48.

John Y. Goulermas (M'98-S'10) received the BSc (1st class) degree in computation from the University of Manchester (UMIST), in 1994, and the MSc and PhD degrees from the Control Systems Center, UMIST, in 1996 and 2000, respectively. He is currently a reader in the Department of Computer Science, University of Liverpool. His research interests include data analysis methods and machine learning, and he has worked with image analysis, biomedical engineering, industrial monitoring, and security applications. He is a senior member of the IEEE.