

Generative Maximum Entropy Learning for Multiclass Classification

Ambedkar Dukkipati, Gaurav Pandey, Debarghya Ghoshdastidar, Paramita Koley, D. M. V. Satya Sriram

Department of Computer Science and Automation

Indian Institute of Science

Bangalore, 560012, India

Email: {ad, gp88, debarghya.g, paramita2000, dmvsriram}@csa.iisc.ernet.in

Abstract—Maximum entropy approach to classification is very well studied in applied statistics and machine learning and almost all the methods that exists in literature are discriminative in nature. In this paper, we introduce a maximum entropy classification method with feature selection for large dimensional data such as text datasets that is generative in nature. To tackle the curse of dimensionality of large data sets, we employ conditional independence assumption (Naive Bayes) and we perform feature selection simultaneously, by enforcing a ‘maximum discrimination’ between estimated class conditional densities. For two class problems, in the proposed method, we use Jeffreys (J) divergence to discriminate the class conditional densities. To extend our method to the multi-class case, we propose a completely new approach by considering a multi-distribution divergence: we replace Jeffreys divergence by Jensen-Shannon (JS) divergence to discriminate conditional densities of multiple classes. In order to reduce computational complexity, we employ a modified Jensen-Shannon divergence (JS_{GM}), based on AM-GM inequality. We show that the resulting divergence is a natural generalization of Jeffreys divergence to a multiple distributions case. As far as the theoretical justifications are concerned we show that when one intends to select the best features in a generative maximum entropy approach, maximum discrimination using J -divergence emerges naturally in binary classification. Performance and comparative study of the proposed algorithms have been demonstrated on large dimensional text and gene expression datasets that show our methods scale up very well with large dimensional datasets.

Keywords-Maximum Entropy; Jefferys Divergence; Jensen-Shannon Divergence; Text categorization;

I. INTRODUCTION

Broadly, supervised learning can be divided into discriminative learning and generative learning [1]. In the generative approach to classification the aim is to model the joint distribution of the data and the class labels from the training data. One can then compute the class conditional densities for each class and then assign the instance to the class with highest posterior probability. Examples of generative classifiers include linear discriminant analysis (LDA) and Bayes classifier.

On the other hand, one can directly model a discriminating function without actually constructing a model for the data. A discriminating function may be chosen so as to minimize some measure of error on the training data. Such an approach is termed as a discriminative classification. Statistical

discriminative classification, a subclass of discriminative classification, models the posterior density directly, which is then used to classify a new instance. Examples of discriminative classifiers include logistic Regression and support vector machines (SVM).

The generative classifiers have a smaller variance than their discriminative counterparts, and hence require lesser data for training [1] to achieve their asymptotic error. This is in contrast to discriminative models, which tend to overfit the data, when the number of training instances is small. Furthermore, incomplete data and latent variables can be taken care of in generative models. Moreover, since generative models model the generation of the entire data (hidden as well as observed), one can incorporate complex dependencies between data/features in the model, thereby allowing the construction of models that are closer to the true data generating mechanism. Any domain knowledge about the data-generating mechanism can be incorporated in a generative model quite easily. Lastly, generative models are more intuitive to understand than their discriminative counterparts.

After the works of [2], [3], a variety of statistical methods and machine learning techniques have taken up ideas from information theory [4], [5]. Maximum entropy or minimum divergence methods form a subclass of such techniques where the aim is to make minimum assumption about the data. These approaches proved to be more useful in the field of natural language processing and text classification [6], [7], where the curse of dimensionality becomes more significant. Variants of maximum entropy techniques have also been considered in literature. For instance, regularized maximum entropy models have been considered in [5].

In spite of the vast literature surrounding maximum entropy models, almost all classification methods considered are discriminative in nature. The reason for this is that the partition function for most generative models cannot be obtained in closed form. However, as mentioned earlier, when the number of training instances is small, a discriminative model tends to overfit the data, and hence, a generative model must be preferred. Hence, we explore generative maximum entropy models in this paper and compare it with other discriminative methods. The use of a generative model also allows us to incorporate feature

selection simultaneously.

Contributions

We propose a new method of classification using a generative model to estimate the class conditional densities. Furthermore, we perform feature selection using a discriminative criteria based on Jeffreys divergence. We call this method as Maximum Entropy with Maximum Discrimination (MeMd). The basic approach is based on the idea presented in [8] that does not scale up for large datasets, and further, extension to multi-class classification is not studied. We improve on the time complexity of the algorithm in [8] by assuming class conditional independence of all the features.

Most of the classification methods that are designed for binary case can be extended to multi-class case by formulating the problem as several binary classification problems [9]. Among these ‘one-vs-all’ is well known and successfully applied to SVMs [10].

One of our main contributions in this paper is the unique way we extend our binary classification method (MeMd) to multi-class case. The main idea is to use Jensen-Shannon divergence, which can be naturally defined for more than two distributions (these divergences are known as multi-distribution divergences [11]). To simplify the calculations, as well as the computational complexity, we replace arithmetic mean in JS -divergences with geometric mean and study performance of MeMd. We show that this leads to a multi-distribution extension of Jeffreys divergence.

We perform experimental study of the proposed method on some large benchmark text datasets and show that the proposed method is able to do drastic dimensionality reduction and also give good accuracies comparable to SVMs and outperforms discriminative approaches.

II. PRELIMINARIES AND BACKGROUND

We dedicate this portion to set up the notations. The problem at hand is that of classification, *i.e.*, we are given a training data set with class labels $\{c_1, c_2, \dots, c_M\}$. In binary classification, we have $M = 2$. Consider each instance of the data to be of the form $\mathbf{x} = (x_1, x_2, \dots, x_d)$, where $x_i \in \mathcal{X}_i$ is the i^{th} feature of the data. So the input space is $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times \dots \times \mathcal{X}_d$. In this setting, our objective is to rank the features according to their ‘discriminative’ ability.

A key step in the proposed algorithms is estimation of the class conditional densities, denoted as $P_{c_j}(\cdot) := P(\cdot|c_j)$, for each class. This step requires some statistics of the data in the form of expected values of certain functions $\Gamma = \{\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \dots, \phi_l(\mathbf{x})\}$, where $\phi_j, j = 1, \dots, l$, is defined over the input space \mathcal{X} . These are often termed as *feature functions* [5], and should not be confused with the features. In fact, the feature functions can be chosen so that they result in the moments of the individual features, for

example it can be of the form $\phi(\mathbf{x}) = x_i^k$, whose expected value gives the k^{th} moment of the i^{th} feature.

Let $\mathbf{X} = (X_1, \dots, X_d)$ be a random vector that takes values from the set \mathcal{X} . Suppose the only information (observations) available about the distribution of \mathbf{X} is in the form of expected values of the real-valued feature functions $\Gamma = \{\phi_1(\mathbf{x}), \dots, \phi_l(\mathbf{x})\}$. We therefore have,

$$E_P[\phi_r(\mathbf{X})] = \int_{\mathcal{X}} \phi_r(\mathbf{x})P(\mathbf{x}) d\mathbf{x} \quad , \quad r = 1, 2, \dots, l, \quad (1)$$

where these expected values are assumed to be known. In the maximum entropy approach to density estimation one choose the distribution of the form

$$P(\mathbf{x}) = \exp\left(-\lambda_0 - \sum_{j=1}^l \lambda_j \phi_j(\mathbf{x})\right) \quad , \quad (2)$$

where $\lambda_0, \lambda_1, \dots, \lambda_l$ are obtained by replacing (2) in (1) and $\int_{\mathcal{X}} P(\mathbf{x})d\mathbf{x} = 1$. This is known as the maximum entropy (ME) distribution.

In presence of an observed data $\{\mathbf{x}^{(k)}, k = 1, \dots, N\}$, in the maximum entropy modeling one assumes that the expected values of the moment constraint functions *i.e.*, $E_P[\phi_r(\mathbf{x})]$ can be approximated by observed statistics or sample means of $\phi_r(\mathbf{x})$, $r = 1, \dots, l$ [7]. Therefore we set

$$E_P[\phi_r(\mathbf{x})] \approx \frac{1}{N} \sum_{k=1}^N \phi_r(\mathbf{x}^{(k)}) = \mu_r^{\text{emp}} \quad , \quad r = 1, \dots, l. \quad (3)$$

For the estimation of $\Lambda = (\lambda_0, \dots, \lambda_l)$ one can show that the maximum likelihood estimator Λ' is given by,

$$\Lambda' = \underset{\Lambda}{\text{argmax}} \frac{1}{N} \sum_{i=1}^N \ln P(\mathbf{x}^{(i)}; \Lambda, \Gamma) \quad , \quad (4)$$

where, $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}$ are assumed to be *i.i.d.* samples drawn from a unknown distribution. To solve (4), one can use iterative methods like gradient descent method or iterative scaling methods [12], [13].

Moving on to divergences, a symmetrized version of KL -divergence [3] is known as Jeffreys divergence [14], or simply J -divergence. Given two pdfs P and Q , J -divergence between them is defined as

$$\begin{aligned} J(P \parallel Q) &= KL(P \parallel Q) + KL(Q \parallel P) \\ &= \int_{\mathcal{X}} (P(x) - Q(x)) \ln \frac{P(x)}{Q(x)} dx \quad . \quad (5) \end{aligned}$$

Although J -divergence was proposed in the context of statistical estimation problems to provide measures of the discrepancy between two laws [14], its connection to KL -divergence has made it popular in classification tasks [15], [16]. Its relationship with other divergence measures have been studied in [17]. In fact, this J -divergence can also be obtained as a f -divergence [18] with a coupled convex function.

Jensen-Shannon (JS) divergence appeared in the literature relatively recently [19], and the unique characteristic of this divergence is that one can measure a divergence between more than two probability distributions. Hence, one can term this as a multi-distribution divergence.

Let P_1, \dots, P_M be probability distributions and let $\bar{P} = \sum_{i=1}^M \pi_i P_i$ be a convex combination of P_1, \dots, P_M , where $\pi_i \in [0, 1]$, $i = 1, \dots, M$ and $\sum_{i=1}^M \pi_i = 1$. Then JS -divergence (or, information radius) among P_1, \dots, P_M is defined as

$$JS(P_1, \dots, P_M) = \sum_{i=1}^M \pi_i KL(P_i \parallel \bar{P}) . \quad (6)$$

JS -divergence is non-negative, symmetric and bounded. For $k = 2$, it has been shown that it is the square of a metric [20]. Grosse et al. [21] studied several interpretations and connections of JS -divergence. However, the close association of this divergence to classification [19] makes it quite interesting in our work.

Related work

Since the seminal work of [22], a plethora of maximum entropy techniques for learning have been introduced in literature. For text classification, a simple discriminative maximum entropy technique was used in [7] to estimate the posterior distribution of class variable conditioned on training data. Each feature is assumed to be a function of the data and the class label. By maximizing the conditional entropy of the class labels, we get a distribution of the form

$$P(c|x) = \frac{1}{Z(x)} \exp \left(\sum_{i=1}^m \lambda_i f_i(x, c) \right), \quad (7)$$

where Z is the normalizing constant. The parameters λ_i , $1 \leq i \leq m$ are obtained by maximizing the conditional log likelihood function. It was shown that for some datasets, the discriminative maximum entropy model outperformed the more commonly used multinomial naive Bayes model significantly. An extension of this approach for sequential data is presented in [23].

Now we proceed to the proposed generative maximum entropy classification and feature selection method.

III. MEMD FOR BINARY CLASSIFICATION

A. Why maximum discrimination?

What is a ‘natural’ way to select features when the intention is to do maximum entropy Bayes classification? In order to answer this question, we first explain what we mean by ‘natural’ with the help of an example. For notational convenience, we assume that the classes are labeled as $+1$ and -1 in this subsection. We will revert to our original notations at the end of this subsection.

Let us assume that our decision surfaces (or classification boundaries) are hyperplanes of the form $f(\mathbf{x}) = \{\mathbf{w}^T \Phi(\mathbf{x}) +$

$b : \|\mathbf{w}\| = 1\}$, where $\Phi = (\phi_1, \dots, \phi_l)$ is a vector of feature functions. Furthermore, assume that the weight vector \mathbf{w} is known and the aim is to select a subset of feature functions. The obvious strategy to select features in such a scenario would be to select them so as to maximize either of the two quantities

$$\hat{\Phi}_{sum} = \operatorname{argmax}_{\Phi} \sum_{i=1}^N y^{(i)} \left(\mathbf{w}^T \Phi(\mathbf{x}^{(i)}) + b \right)$$

$$\hat{\Phi}_{min} = \operatorname{argmax}_{\Phi} \min_{i=1}^N y^{(i)} \left(\mathbf{w}^T \Phi(\mathbf{x}^{(i)}) + b \right),$$

where $\mathbf{x}^{(1)}, \mathbf{x}^{(2)} \dots \mathbf{x}^{(N)}$ are N *i.i.d* samples that constitute the training data and $y^{(i)}$ are their corresponding class labels taking values in $\{+1, -1\}$. Here, the quantity $y^{(i)}(\mathbf{w}^T \Phi(\mathbf{x}^{(i)}) + b)$ measures the classification margin for the point $\mathbf{x}^{(i)}$. Ideally, we would like it to be as high as possible for all points.

The question that we wish to address is whether we can follow a similar approach to select subset of features for the Bayes classifier. In Bayes classification, a point \mathbf{x} is assigned the label $+1$ if

$$\pi_+ P_+(\mathbf{x}) > \pi_- P_-(\mathbf{x}),$$

which can be rewritten as

$$\log \frac{\pi_+ P_+(\mathbf{x})}{\pi_- P_-(\mathbf{x})} > 0.$$

Here π_+, π_- are the prior probabilities and P_+ and P_- denote the class conditional probabilities of the two classes. Hence, the above quantity plays the same role as the equation of hyperplane in the former case. Therefore, we can define our Bayes’ classification margin for the point \mathbf{x} as $y \log \frac{\pi_+ P_+(\mathbf{x})}{\pi_- P_-(\mathbf{x})}$ which must be positive for all correctly classified points. As in the case of hyperplanes, we can select features so as to maximize either of the two quantities.

$$\Gamma_{sum}^* = \operatorname{argmax}_{S \in 2^\Gamma} \sum_{i=1}^N y^{(i)} \log \frac{\pi_+ P_+(\mathbf{x}^{(i)}; S)}{\pi_- P_-(\mathbf{x}^{(i)}; S)} \quad (8)$$

$$\Gamma_{min}^* = \operatorname{argmax}_{S \in 2^\Gamma} \min_{i=1}^N y^{(i)} \log \frac{\pi_+ P_+(\mathbf{x}^{(i)}; S)}{\pi_- P_-(\mathbf{x}^{(i)}; S)},$$

where Γ is the set of all feature functions and $S \subset \Gamma$.

If the class conditional distributions are obtained using maximum entropy by using the expected values of feature functions in S , we can further simplify (8) by plugging in the equation for maximum entropy distribution for the two classes. The corresponding feature selection problem then

becomes

$$\Gamma_{sum}^* = \arg \max_{S \in 2^\Gamma} \left(\log \frac{Z_2}{Z_1} + \log \frac{\pi_+}{\pi_-} \right) \left(\sum_{i=1}^N y^{(i)} \right) + \sum_{\phi \in S} (\lambda'_\phi - \lambda_\phi) \left(\sum_{i=1}^N y^{(i)} \phi(\mathbf{x}^{(i)}) \right), \quad (9)$$

where

$$P_+^*(\mathbf{x}^{(i)}; S) = \frac{1}{Z_1} \exp \left(- \sum_{\phi \in S} \lambda_\phi \phi(\mathbf{x}^{(i)}) \right) \quad (10)$$

$$P_-^*(\mathbf{x}^{(i)}; S) = \frac{1}{Z_2} \exp \left(- \sum_{\phi \in S} \lambda'_\phi \phi(\mathbf{x}^{(i)}) \right). \quad (11)$$

We use the superscript “*” to indicate that the class conditional distributions are the ME distributions.

If the number of training points in the two classes are equal, the first term in (9) can be discarded. Furthermore, by separating the terms in the two classes, we get

$$\Gamma_{sum}^* = \arg \max_{S \in 2^\Gamma} \left(\sum_{\phi \in S} (\lambda'_\phi - \lambda_\phi) (\mu_\phi - \mu'_\phi) \right) \times N \quad (12)$$

It is easy to see that this is exactly N times the J -divergence between the distributions in (10) and (11). One can similarly show that when the number of points in the two classes are not equal, we still obtain the above equation if we assign proper weights to instances in the two classes (based on the number of points in the class). Reverting back to our original notations, we can say that if one intends to use Bayes classifier where the class conditional distributions are obtained using maximum entropy, a natural way to do so would be to select features as below.

$$\Gamma^* = \arg \max_{S \in 2^\Gamma} J(P_{c_1}^*(x; S) \| P_{c_2}^*(x; S)), \quad (13)$$

where $P_{c_j}^*(x; S)$ indicates the ME distribution estimated for class c_j using expected values of the feature functions that are in $S \subset \Gamma$.

B. The MeMd Approach

The MeMd approach can be formulated as follows. Given a set of feature functions $\Gamma = \{\phi_1, \dots, \phi_l\}$, the problem is to find the subset $\Gamma^* \subset \Gamma$ such that

$$\Gamma^* = \arg \max_{S \in 2^\Gamma} J(P_{c_1}^*(x; S) \| P_{c_2}^*(x; S)), \quad (14)$$

where $P_{c_j}^*(x; S)$ indicates the ME distribution estimated for class c_j , $j \in \{1, 2\}$, using expected values of the feature functions that are in $S \subset \Gamma$.

This problem is intractable particularly when large number of features are involved (which is the case for high-dimensional data) since it involves estimation of J -divergence for 2^l subsets to find the optimal subset from

the given set of l feature functions. The problem was studied in Dukkipati et al. [8] where a greedy search was used to select the features thereby reducing the complexity from exponential to $O(l^2)$. Even with this greedy approach, the method does not scale up well for large dimensional data such as text data. Moreover estimating the ME distributions, and hence finding the exact value of J -divergence between the estimated class conditional densities is computationally demanding especially for large dimensional data.

Our strategy is to use naive Bayes approach since for text data, naive Bayes classifiers have shown to outperform (or given comparable performance) compared with other classifiers [24] and its good performance is attributed to optimality under zero-one loss function [25]. Therefore we have

$$P_{c_j}(\mathbf{x}) = \prod_{i=1}^d P_{c_j}^{(i)}(x_i),$$

where c_j is the class label and $P_{c_j}^{(i)}$ is the marginal density for x_i , the i^{th} feature of the data \mathbf{x} .

This leads to simplification of the greedy step of the algorithm proposed by Dukkipati et al. [8] to great extent as shown in following result.

Theorem 1. *The feature chosen at the k^{th} step of the greedy approach is the one with k^{th} largest J -divergence between the marginals of class conditional densities.*

Proof: Using the additivity of KL -divergence under independence, J -divergence between the two class conditional densities P_{c_1} and P_{c_2} can be written as

$$J(P_{c_1} \| P_{c_2}) = J \left(\prod_{i=1}^d P_{c_1}^{(i)} \left\| \prod_{i=1}^d P_{c_2}^{(i)} \right. \right) = \sum_{i=1}^d J(P_{c_1}^{(i)} \| P_{c_2}^{(i)}). \quad (15)$$

Suppose a set S of $(k-1)$ features are already chosen. The corresponding approximation of the class conditional density is

$$P_{c_j}(\mathbf{x}) \approx \prod_{i \in S} P_{c_j}^{(i)}(x_i), \quad j = 1, 2,$$

and the optimal feature is

$$\begin{aligned} j^* &= \arg \max_{j \notin S} J \left(\prod_{i \in S \cup \{j\}} P_{c_1}^{(i)} \left\| \prod_{i \in S \cup \{j\}} P_{c_2}^{(i)} \right. \right) \\ &= \arg \max_{j \notin S} \left\{ J \left(\prod_{i \in S} P_{c_1}^{(i)} \left\| \prod_{i \in S} P_{c_2}^{(i)} \right. \right) + J \left(P_{c_1}^{(j)} \| P_{c_2}^{(j)} \right) \right\} \\ &= \arg \max_{j \notin S} J \left(P_{c_1}^{(j)} \left\| P_{c_2}^{(j)} \right. \right). \end{aligned}$$

The above maximization is equivalent to choosing the feature with k^{th} largest J -divergence in the k^{th} step. ■

Thus J -divergence can be readily used to rank the features based on the discrimination between the two classes,

and prune out those features which gives rise to a small value of J -divergence between the estimated marginal densities $P_{c_j}^{(i)}(x_i)$, which are of the ME form

$$P_{c_j}^{(i)}(x_i) = \exp\left(-\lambda_{0_{ij}} - \sum_{k=1}^l \lambda_{k_{ij}} \phi_k(x_i)\right), \quad (16)$$

for each class c_j , $j = 1, 2$, and each feature x_i , $i = 1, 2, \dots, d$.

For classification, any standard method can be used. However, since the class conditional densities are estimated during the above process, Bayes decision rule [26] turns out to be an obvious choice, *i.e.*, a test pattern is assigned class c_1 if

$$P_{c_1}(\mathbf{x})P(c_1) > P_{c_2}(\mathbf{x})P(c_2),$$

otherwise to class c_2 , where $P(c_1)$ and $P(c_2)$ are the priors for each class. Using only the top K features, the class conditional densities can be approximated as

$$P_{c_j}(\mathbf{x}) \approx \prod_{i \in S} P_{c_j}^{(i)}(x_i), \quad j = 1, 2. \quad (17)$$

So, the decision rule can be written as

$$\prod_{i \in S} \frac{P_{c_1}^{(i)}(x_i)}{P_{c_2}^{(i)}(x_i)} > \frac{P(c_2)}{P(c_1)},$$

which, after taking logarithm, turns out to be

$$\sum_{i \in S} (\lambda_{0_{i2}} - \lambda_{0_{i1}} + \sum_{k=1}^l (\lambda_{k_{i2}} - \lambda_{k_{i1}}) \phi_k^{(i)}(x_i)) > \ln P(c_2) - \ln P(c_1). \quad (18)$$

We list the above method in Algorithm 1. The corresponding experimental results are presented in Section VI.

An interesting fact to note here is that for distributions of the form in (16), the J -divergence can be obtained in a simple form as given in the following result.

Remark 1. Suppose there are two ME distributions $P(x) = \exp\left(-\lambda_0 - \sum_{j=1}^l \lambda_j \phi_j(x)\right)$ and $Q(x) = \exp\left(-\lambda'_0 - \sum_{j=1}^l \lambda'_j \phi_j(x)\right)$, obtained using same set of feature functions $\{\phi_1, \dots, \phi_l\}$, but with different expected values $\{\mu_1, \dots, \mu_l\}$ and $\{\mu'_1, \dots, \mu'_l\}$, respectively. Then

$$J(P \parallel Q) = \sum_{j=1}^l (\lambda'_j - \lambda_j)(\mu_j - \mu'_j). \quad (19)$$

This result can be used to evaluate the J -divergence between the marginals of the class conditional densities for each feature, which can be used to rank all the features in $O(d)$ time (in decreasing value of J -divergence).

Algorithm 1 MEMD with Naive Bayes for binary classification

INPUT :

- Two labeled datasets of class c_1 and c_2 .
- Data of the form $\mathbf{x} = (x_1, x_2, \dots, x_n)$, x_i denoting i^{th} feature.
- A set of l constraints $\Gamma^{(i)} = \{\phi_1^{(i)}, \dots, \phi_l^{(i)}\}$ to be applied on each feature x_i , $i = 1, \dots, d$.

ALGORITHM :

- 1) ME densities for each $P_{c_j}^{(i)}$, $i = 1, 2, \dots, d$ and $j = 1, 2$ are estimated using (16).
 - 2) J -divergence for each feature (denote as J_i , $i = 1, \dots, d$) is calculated using (5).
 - 3) The features are ranked in descending order according to their J -divergence values, and the top K features are chosen (to be considered for classification).
 - 4) Bayes decision rule is used for classification using (18).
-

IV. MULTI CLASS CLASSIFICATION

A. One vs. All Approach

The most common technique in practice for multi-class classification is to use “one vs. all” approach, where the number of classifiers built is equal to the number of classes, *i.e.*, for each class we build a classifier for that class against all the other classes [9]. Incorporating such a technique does not affect the basic MeMd approach for ranking the features. Hence, Algorithm 1 can be easily extended to multi-class case by using one vs. all approach as presented in Algorithm 2.

Here, we consider a M -class problem, with classes c_1, c_2, \dots, c_M . The rest of the setting is same as Algorithm 2. The modification can be described as follows. For each class c_j , consider the class $c'_j = \cup_{k \neq j} c_k$. So, the ‘discriminative capacity’ of each feature for a particular class can be measured by its J -divergence between the class conditional densities c_j and c'_j ,

$$J_{ij} := J\left(P_{c_j}^{(i)} \parallel P_{c'_j}^{(i)}\right), \quad (20)$$

where $P_{c_j}^{(i)}$ and $P_{c'_j}^{(i)}$ are the ME marginal densities for classes c_j and c'_j , respectively, for the i^{th} feature. J_{ij} can be easily computed using (19). However, this provides us a J -divergence for each feature for a particular class. A natural way to obtain a J -divergence for each feature is

$$J_i = \sum_{j=1}^M J_{ij} P(c_j), \quad (21)$$

i.e., the divergence is averaged over all the classes, weighted by their prior probabilities $P(c_j)$. The algorithm is listed as below.

Algorithm 2 (MeMd-J) : MEMD for multi-class classification using one vs. all approach.

INPUT :

- Labeled datasets of M classes.
- Data of the form $\mathbf{x} = (x_1, x_2, \dots, x_n)$, x_i denoting i^{th} feature.
- A set of l constraints $\Gamma^{(i)} = \{\phi_1^{(i)}, \dots, \phi_l^{(i)}\}$ to be applied on each feature x_i , $i = 1, \dots, d$.

ALGORITHM :

- 1) Construct the classes $c'_j = \bigcup_{k \neq j} c_k$, $j = 1, 2, \dots, M$.
- 2) ME densities $P_{c_j}^{(i)}$ and $P_{c'_j}^{(i)}$, $i = 1, 2, \dots, d$ and $j = 1, 2, \dots, M$ are estimated using (16).
- 3) The average J -divergence for each feature is calculated using (21).
- 4) The features are ranked in descending order according to their average J -divergence values, and the top K features are chosen (to be considered for classification).
- 5) Bayes decision rule is used to assign a test pattern to c_{j^*} such that

$$j^* = \arg \max_{j=1, \dots, M} P_{c_j}(\mathbf{x})P(c_j),$$

where the class conditional densities are approximated as in (17) for $j = 1, 2, \dots, M$.

The above approach is an obvious extension of the binary classification problem. Although the algorithm is $O(d)$, it is relatively computationally inefficient in the sense that it requires estimation of the additional ME distributions $P_{c'_j}^{(i)}$ (which is the most time consuming step of MeMd approach). Now, we present our main contribution in this paper. To deal with the multi-class classification, we invoke a natural multi-distribution divergence, the JS -divergence.

B. Multi class Classification using JS -divergence

The JS -divergence, as mentioned in Section II can be defined over multiple distributions, and hence, would seem to be more useful for dealing with multi-class problems. However, its most interesting feature is presented in the following result [21]. We state it in a form more suitable for our purpose, and interpret it for multi-class classification similarity.

Proposition 1. Consider a M -class problem with class-conditional densities $P_{c_i}(x) = P(x|c_i)$, $i = 1, 2, \dots, M$. If X is a random data, and Z is an indicator random variable, i.e., $Z = i$ when $X \in c_i$, then

$$JS(P_{c_1}, \dots, P_{c_M}) = MI(X; Z).$$

Here, $MI(X; Z)$ is the mutual information between X and Z , and the JS -divergence is computed using the priors as

weights, (i.e., $\pi_j = P(c_j)$) as

$$JS(P_{c_1}, \dots, P_{c_M}) = \sum_{j=1}^M P(c_j) KL(P_{c_j} \| \bar{P}), \quad (22)$$

where $\bar{P}(X) = \sum_{j=1}^M P(c_j)P_{c_j}(X)$ is the probability of sampling the data X .

The problem of classification basically deals prediction of the value of Z , given the test data X . Hence, it is natural to devise a technique which would maximize the mutual information, or rather, maximize the JS -divergence among the class-conditional densities. Thus JS -divergence turns out to be a more natural measure of discrimination compared to J -divergence. However, unlike J -divergence, JS -divergence does not satisfy the additivity property, i.e., there is no equivalent for (15) for JS -divergence. Hence, the “correct” extension of the algorithm in case of JS -divergence becomes involved as we need to consider a quadratic time greedy approach, where at each iteration we select a new feature that maximizes JS -divergence among the classes and also has minimum mutual information with the previously chosen feature set.

The latter term leads to an $O(d^2)$ time complexity for the feature selection rule. A linear time algorithm similar to MeMd-J can be obtained if we assume that all the features are independent, which is a quite strict assumption, particularly for text datasets. In spite of the above issues, use of JS -divergence is quite significant due to its theoretical justifications. Moreover, use of a multi-distribution divergence significantly reduces the the number of models that needs to be estimated in a one-vs-all approach. Hence, we look for an approximation of the JS -divergence that some-what retains its properties, and can also exploit the linearity of J -divergence.

V. JS_{GM} - DISCRIMINATION

A. AM to GM

Although JS -divergence seemed to be a natural discriminative measure, its drawbacks from algorithmic point of view motivates us to look into some form of approximation in terms of J -divergence, which looked more promising (computationally). The idea comes from [27], which introduced the notion of average divergence by just taking average over KL -divergence between all possible pairs of distributions.

However, we approach the problem differently, but obtain a similar result. The modification is made at the basic definition of JS -divergence (6), where we replace the weighted mean $\bar{P} = \sum_{i=1}^k \pi_i P_i$ by the weighted geometric mean $\bar{P}' = \prod_{i=1}^k P_i^{\pi_i}$. Though \bar{P}' does not have any straightforward physical interpretation (in fact, it is not a distribution), it leads to simpler expressions as discussed below.

We call the corresponding divergence as JS -divergence with geometric mean, or simply JS_{GM} -divergence. Given distributions P_1, P_2, \dots, P_M with weights $\pi_1, \pi_2, \dots, \pi_M$, the JS_{GM} -divergence among them is given by

$$JS_{GM}(P_1, \dots, P_M) = \sum_{i=1}^M \pi_i KL(P_i \| \bar{P}') \\ = \sum_{i=1}^M \sum_{j \neq i} \pi_i \pi_j KL(P_i \| P_j). \quad (23)$$

In the case of uniform weights ($\pi_i = \frac{1}{M}$), it is same as the average divergence [27] upto a constant scaling factor. Observing the symmetric nature of (23), the divergence can also be written in terms of J -divergence as

$$JS_{GM}(P_1, \dots, P_M) = \frac{1}{2} \sum_{i=1}^M \sum_{j \neq i} \pi_i \pi_j J(P_i \| P_j). \quad (24)$$

The JS_{GM} -divergence satisfies the following inequality, which has been shown in [27] for uniform weights.

Proposition 2. For any set of weights $\pi_1, \pi_2, \dots, \pi_M$,

$$JS(P_1, \dots, P_M) \leq JS_{GM}(P_1, \dots, P_M)$$

Proof: The claim follows from the observation that KL -divergence is convex in its second argument. ■

The usefulness of the JS_{GM} -divergence stems from the fact that it is a multi-distribution divergence that generalizes J -divergence, which helps us to overcome the difficulties of the one vs. all approach. So it does not require estimation of extra ME distributions. At the same time, it can be expressed in terms of J -divergence (24), which helps us to exploit the nice properties of J -divergence discussed in Theorem 1 and Remark 1. So, the following algorithm using JS_{GM} -divergence is analogous to the binary classification algorithm with J -divergence (Algorithm 1). The equivalence can be easily observed by considering the following equivalence. For each feature, we replace

$$J\left(P_{c_j}^{(i)} \parallel P_{c_j'}^{(i)}\right) \text{ in Algo 2} \longleftrightarrow \\ \sum_{k \neq j} P(c_k) J\left(P_{c_j}^{(i)} \parallel P_{c_k}^{(i)}\right) \text{ in Algo 3}, \quad (25)$$

i.e., J -divergence between class conditional densities for c_j and all other classes taken together (c_j') is replaced by the weighted average of the J -divergences between c_j and other classes.

We can also extend the arguments in Section III-A to show that for maximum entropy Bayes classification in the multi-class scenario, JS_{GM} -divergence is a natural choice. The algorithm has linear time complexity $O(d)$, and requires half the time as compared to Algorithm 2 (one vs. all). However, the performance of both algorithms are quite similar as

Algorithm 3 (MeMd-JS) : MEMD for multi-class classification using JS_{GM} -divergence

INPUT :

- Labeled datasets of M classes.
- Data of the form $\mathbf{x} = (x_1, x_2, \dots, x_n)$, x_i denoting i^{th} feature.
- A set of l constraints $\Gamma^{(i)} = \{\phi_1^{(i)}, \dots, \phi_l^{(i)}\}$ to be applied on each feature x_i , $i = 1, \dots, d$.

ALGORITHM :

- 1) ME densities $P_{c_j}^{(i)}$, $i = 1, 2, \dots, d$ and $j = 1, 2, \dots, M$ are estimated using (16).
- 2) The J -divergence for each feature and each pair of classes is calculated using (22).
- 3) The JS_{GM} -divergence for each feature can be computed as

$$JS_{GM}^{(i)} = JS_{GM}(P_{c_1}^{(i)}, P_{c_2}^{(i)}, \dots, P_{c_M}^{(i)}) \\ = \sum_{j=1}^M \sum_{k \neq j} P(c_j) P(c_k) J\left(P_{c_j}^{(i)} \parallel P_{c_k}^{(i)}\right).$$

- 4) The features are ranked in descending order according to their JS_{GM} -divergence values, and the top K features are chosen (to be considered for classification).
- 5) Bayes decision rule is used to assign a test pattern to c_{j^*} such that

$$j^* = \arg \max_{j=1, \dots, M} P_{c_j}(\mathbf{x}) P(c_j),$$

where the class conditional densities are approximated as in (17) for $j = 1, 2, \dots, M$.

discussed in the experimental results.

VI. EXPERIMENTAL RESULTS

In this section, we compare the proposed algorithms with some popular algorithms, commonly used in practice. Table I provides a summary of the proposed algorithms, as well as the existing algorithms. We also introduce some notations for each algorithm, which will be used in sequel to refer to the algorithm. We note that the binary classification algorithm (Algorithm 1) is a special case of the multi-class “one vs. all” approach (Algorithm 2), and so, in sequel, we always refer to both together as MeMd-J algorithm.

A. Summary of computational complexity of algorithms

We first discuss about the computational complexity of the various algorithms. This is detailed in Table I. It shows the training and testing time complexities of the various algorithms in terms of number of features (d), number of classes (M) and number of training samples (N). For SVM, S denotes the number of support vectors, while for proposed algorithms K is the chosen number of features (obtained

Table I: Comparison of complexity of algorithms.

Algorithm	Notation	Classification	Training time		Testing time per sample
			Estimation	Feature ranking	
MeMd using one vs. all approach	MeMd-J	multiclass	$O(MNd)$	$O(Md + d \log d)$	$O(MK)$
MeMd using JS_{GM} -divergence	MeMd-JS	multiclass	$O(MNd)$	$O(M^2d + d \log d)$	$O(MK)$
MeMd using greedy approach [8]	MeMd	binary	$O(Nd^2)$		$O(Md)$
Support Vector Machine [28]	SVM	multiclass	#iterations* $O(Md)$		$O(M^2Sd)$
Discriminative approach using ME [7]	DME	multiclass	#iterations* $O(MNd)$		$O(Md)$

Table II: Performance comparison on gene expression datasets.

Dataset	Data attributes			10-fold cross validation accuracy		
	No. of classes	No. of samples	No. of features	SVM (linear)	MeMD-J (2-moment)	MeMd-JS (2-moment)
Colon cancer [29]	2	62	2000	84.00	86.40	
Leukemia [29], [30]	2	72	5147	96.89	98.97	
Embryonal tumors of CNS [29]	2	60	7129	62.50	63.75	
DLBCL [30]	2	77	7070	97.74	86.77	
Prostate cancer [30]	2	102	12533	89.51	89.75	
SRBCT [30]	4	83	2308	99.20	97.27	98.33
Human lung carcinomas [30]	5	203	12600	93.21	93.52	92.60
Global Cancer Map [29]	14	190	16063	66.85	66.98	66.98

from the ranking). The training time complexity for SVM assumes that the rows have been cached, as in the case of LIBSVM. SVM was implemented using LIBSVM [28], which uses a “one vs. one” approach. Furthermore, the number of iterations for SVM tend to be $O(N^2)$, whereas for DME, the algorithm gives best results when the number of iterations is small ($O(1)$) as mentioned in [7].

The greedy MeMd technique [8] is severely affected by curse of dimensionality. In addition to its quadratic complexity, implementations indicated that the 2-moment ME joint distribution (Gaussian) appeared to be unstable due to the inverse of the large dimensional covariance matrix. Hence, results corresponding to this algorithm have not been presented in the comparisons. We also skip the MeMd algorithm using JS -divergence since determination of true value of JS -divergence involves considerable amount of computation as it requires numerical calculation of integrals. Hence, we work with its approximate version (Algorithm 3). It is worth noting here that though MeMd-JS and MeMd-J have same order of complexity for parameter estimation, for each class, MeMd-J builds a model for all data not in the class. Hence, it estimates twice the number of parameters. Hence, using MeMd-JS over MeMd-J is computationally efficient in the following cases.

- 1) The number of points is large thereby making parameter estimation time for MeMd-J twice as MeMd-JS. Note that feature selection phase does not depend on the number of points, hence, having an M^2 term in MeMd-JS is not a problem.

- 2) The number of iterations required for parameter estimation is not $O(1)$, *i.e.*, when we use iterative scaling algorithm. However, this issue is not there if only 1-moment or 2-moment models are considered.

B. Experiments on biological datasets

We compare the performance of our algorithms (MeMd-J and MeMd-JS) with that of SVM on a variety of gene datasets, collected from [29], [30]. The details of the datasets are presented in Table II, where the number of features is same as the number of genes in these datasets. We again mention here that the greedy MeMd [8] and MeMd using JS -divergence have not been implemented due to their computational complexities addressed in the previous section.

Table II also lists the accuracies obtained from different algorithms using 10-fold cross-validation, where the folds are chosen randomly. The class conditional distributions are 2-moment ME distributions. The optimal number of features is selected by cross validation within the training sample. We note here that for binary classification, the procedure in both MeMd-J and MeMd-JS are same, *i.e.*, they result in the same algorithm. Hence, in these cases, the results for both algorithms are combined together.

The best accuracy for each dataset is highlighted. From Table II, one can observe that the MeMd classifier is more successful in distinguishing among the classes for most of the cases. This is because, a very small number of genes are actually involved in these diseases. By using a feature pruning strategy in the classification algorithm, the MeMd

classifier is able to prune away most of the genes that are not discriminatory.

C. Experiments on text datasets

We perform our experiments on the 20 Newsgroups dataset obtained from [31]. The dataset contains 20 classes, which have been grouped in different ways to construct different binary and multiclass problems. The data has been preprocessed prior to classification. We remove all stop words, and words with frequency less than some cut-off value (γ) from the entire document corpus. This is done since such words do not have much discriminative power. The value of γ is chosen to be 2 in all cases. Each document D_j of the text corpus is represented as a vector of term-weights $D_j = \langle W_{1j}, W_{2j}, \dots, W_{Tj} \rangle$, where W_{ij} represents the normalized frequency of the word w_i in the document D_j [32], i.e., $W_{ij} = \frac{N(w_i, D_j)}{\sum_{k=1}^T N(w_k, D_j)}$, where $N(w_i, d_j)$ is the number of times the word w_i occurs in the document D_j , and T is the total number of words in the corpus. Thus $0 \leq W_{kj} \leq 1$ represents how much the word w_k contributes to the semantics of the document D_j .

Table III presents the classification accuracy of the proposed methods (MeMd-J and MeMd-JS) along with SVM and DME using 2-fold cross-validation. The main reason behind using only two folds is that, in such a case, the training data size reduces considerably, and learning correctly from small data becomes an additional challenge. We demonstrate how the proposed generative approach overcomes this problem. Furthermore, MeMd-J and MEMd-JS algorithms use only the top K ranked features for classification. For choosing the optimal value of K , we employ the following strategy. We divide the training data in two portions: a considerable fraction (80%) of the training data is used to rank the features, while the remaining 20% is used as a “test” set on which classification is performed using varying number of features. Hence, we obtain a plot of the accuracy vs. the number of features. We choose K to be the minimum number of features, where the maximum accuracy is attained for that small portion of training data. We also note, as in previous section, that for binary classification MeMd-J and MeMd-JS are exactly same, and so we present their results together.

The best accuracies for each experiment is shown in bold. We also underline the cases, where results are very close to best case (more than -0.2% of maximum accuracy). Our observation can be summarized as follows: In all of the cases (except one), MeMd outperforms linear SVM, which showed quite poor performance. We also implemented SVM with polynomial and RBF, which exhibited even poorer performance, and hence, those results are not presented. On the other hand, we observed that DME performs quite well, providing the best accuracies in a considerable number of cases. Further, in multiclass problems, MeMd-J is always observed to perform marginally better than MeMd-JS.

VII. CONCLUSION

As far as our knowledge, this is the first work that proposes and studies a generative maximum entropy approach to classification. In this paper, we proposed a method of classification using maximum entropy with maximum discrimination (MeMd) which is a generative approach with simultaneous feature selection. The proposed method is suitable for large dimensional text dataset as the classifier is built in linear time with respect to the number of features and it provides a way to eliminate redundant features. Also this is the first work that uses multi-distribution divergence in multiclass classification. It will also be interesting to study the proposed methods as feature selection algorithms.

REFERENCES

- [1] A. Y. Ng and M. Jordan, “On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes,” in *Advances in Neural Information Processing Systems*. MIT Press, 2002, p. 14.
- [2] E. T. Jaynes, “Information theory and statistical mechanics I,” *Physical Review*, vol. 106, no. 4, pp. 620–630, 1957.
- [3] S. Kullback, *Information Theory and Statistics*. New York: Wiley, 1959.
- [4] S. C. Zhu, Y. N. Wu, and D. Mumford, “Minimax entropy principle and its application to texture modelling,” *Neural Computation*, vol. 9, pp. 1627–1660, 1997.
- [5] M. Dudik, S. J. Phillips, and R. E. Schapire, “Maximum entropy density estimation with generalized regularization and an application to species distribution modeling,” *Journal of Machine Learning Research*, vol. 8, pp. 1217–1260, 2007.
- [6] D. Beeferman, A. Berger, and J. Lafferty, “Statistical models for text segmentation,” *Machine Learning*, vol. 34, no. 1, pp. 177–210, 1999.
- [7] K. Nigam, J. Lafferty, and A. McCallum, “Using maximum entropy for text classification,” in *IJCAI-99 workshop on machine learning for information filtering*, 1999, pp. 61–67.
- [8] A. Dukkipati, A. K. Yadav, and M. N. Murty, “Maximum entropy model based classification with feature selection,” in *Proceedings of IEEE International Conference on Pattern Recognition (ICPR)*. IEEE Press, 2010, pp. 565–568.
- [9] E. Allwein, R. E. Schapire, and Y. Singer, “Reducing multiclass to binary: A unifying approach for margin classifiers,” *Journal of Machine Learning Research*, vol. 1, pp. 113–141, 2000.
- [10] R. Rifkin and A. Klautau, “In defense of one-vs-all classification,” *The Journal of Machine Learning Research*, vol. 5, pp. 101–141, 2004.
- [11] D. García-García and R. C. Williamson, “Divergences and risks for multiclass experiments,” in *Proceedings of Annual Conference on Learning Theory*, vol. 23, 2012, pp. 28.1–28.20.

Table III: Performance comparison on 20 Newsgroup dataset (‘/’ used to separate classes; names abbreviated in some cases).

Experiment	Data attributes			2-fold cross validation accuracy			
	No. of classes	No. of samples	No. of features	SVM (RBF)	DME	MeMD-J (1-moment)	MeMd-JS (1-moment)
rec.autos / rec.motorcycles	2	1588	7777	95.96	95.59	97.35	
talk.religion / soc.religion	2	1227	8776	91.35	92.33	91.69	
comp.os.ms-windows / comp.windows.x	2	1973	9939	92.80	93.60	93.81	
comp.sys.ibm / comp.sys.mac	2	1945	6970	89.61	90.48	89.77	
rec.(autos,motorcycles) / rec.sport	2	3581	13824	98.49	98.68	99.02	
sci.(space,encrypt) / sci.(med,elec)	2	3952	17277	96.63	96.93	95.04	
comp.windows(os,x) / comp.sys.(ibm,mac)	2	3918	13306	88.23	91.88	<u>91.75</u>	
all ‘talk’ categories	4	3253	17998	88.62	90.34	91.91	91.39
all ‘sci’ categories	4	3952	17275	94.63	95.26	<u>95.14</u>	94.88
all ‘rec’ categories	4	3581	13822	95.83	96.23	<u>96.14</u>	95.86
all ‘comp’ categories	4	4891	15929	81.08	83.41	82.11	82.03

- [12] J. N. Darroch and D. Ratcliff, “Generalized iterative scaling for log-linear models,” *The annals of mathematical statistics*, vol. 43, no. 5, pp. 1470–1480, 1972.
- [13] I. Csizsár, “A geometric interpretation of Darroch and Ratcliff’s generalized iterative scaling,” *The Annals of Statistics*, pp. 1409–1413, 1989.
- [14] H. Jeffreys, “An invariant form for the prior probability in estimating problems,” *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, vol. 186, no. 1007, pp. 453–461, 1946.
- [15] R. Nishii and S. Eguchi, “Image classification based on markov random field models with Jeffreys divergence,” *Journal of multivariate analysis*, vol. 97, pp. 1997–2008, 2006.
- [16] T. Deselaers, D. Keysers, and H. Ney, “Features for image retrieval: A quantitative comparison,” *Pattern Recognition*, pp. 228–236, 2004.
- [17] S. S. Dragomir, J. Sunde, and C. Buse, “New inequalities for Jeffreys divergence measure,” *Tamsui Oxford Journal of Mathematical Sciences*, vol. 16, pp. 295–309, 2000.
- [18] S. M. Ali and S. D. Silvey, “A general class of coefficients of divergence of one distribution from another,” *J. Roy. Statist. Soc. Ser. B*, vol. 28, pp. 131–142, 1966.
- [19] J. Lin, “Divergence measures based on the Shannon entropy,” *IEEE Transactions on Information Theory*, vol. 37, pp. 145–151, 1991.
- [20] D. M. Endres and J. E. Schindelin, “A new metric for probability distributions,” *IEEE Transactions on Information Theory*, vol. 49, pp. 1858–1860, 2003.
- [21] I. Grosse, P. Bernaola-Galván, P. Carpena, R. Román-Roldán, J. Oliver, and H. E. Stanley, “Analysis of symbolic sequences using the Jensen-Shannon divergence,” *Physical Review E*, vol. 65, 2002.
- [22] A. L. Berger, V. J. D. Pietra, and S. A. D. Pietra, “A maximum entropy approach to natural language processing,” *Computational Linguistics*, vol. 22, no. 1, pp. 39–71, 1996.
- [23] A. McCallum, D. Freitag, and F. Pereira, “Maximum entropy Markov models for information extraction and segmentation,” in *Proceedings of the Seventeenth International Conference on Machine Learning*, vol. 951, 2000, pp. 591–598.
- [24] N. Friedman, D. Geiger, and M. Goldszmidt, “Bayesian network classifiers,” *Machine Learning*, vol. 29, pp. 131–163, 1997.
- [25] P. Domingos and M. Pazzani, “On the optimality of the simple Bayesian classifier under zero-one loss,” *Machine learning*, vol. 29, pp. 103–130, 1997.
- [26] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. Wiley, 2001.
- [27] A. Sgarro, “Informational divergence and the dissimilarity of probability distributions,” *GNIM-CNR research activity*, December 1979.
- [28] C. C. Chang and C. J. Lin, “LIBSVM: A library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 27:1–27:27, 2011, software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [29] BioInformatics Research Group of Seville, “Dataset repository in ARFF (WEKA), Available at: <http://www.upo.es/eps/big5/datasets.html>.”
- [30] M. Mramor, G. Leban, J. Demšar, and B. Zupan, “Supplementary material to: Visualization-based cancer microarray data classification analysis,” *Bioinformatics*, vol. 23, no. 16, pp. 2147–2154, Available at: <http://www.biolab.si/supp/bi-cancer/projections/>, 2007.
- [31] K. Bache and M. Lichman, “UCI machine learning repository,” *University of California, Irvine, School of Information and Computer Sciences*, vol. Available at: <http://archive.ics.uci.edu/ml>, 2013.
- [32] F. Sebastiani, “Machine learning in automated text categorization,” *ACM Computing Surveys*, vol. 34, pp. 1–47, 2002.