Learning to Separate Text Content and Style for Classification

Dell Zhang[1] and Wee Sun Lee[2]

[1] School of Computer Science and Information Systems
Birkbeck, University of London
London WC1E 7HX, UK
dell.z@ieee.org
[2] Department of Computer Science and Singapore-MIT Alliance
National University of Singapore
Singapore 117543
leews@comp.nus.edu.sg

Abstract. Many text documents naturally have two kinds of labels. For example, we may label web pages from universities according to their categories, such as "student" or "faculty", or according the source universities, such as "Cornell" or "Texas". We call one kind of labels the content and the other kind the style. Given a set of documents, each with both content and style labels, we seek to effectively learn to classify a set of documents in a new style with no content labels into its content classes. Assuming that every document is generated using words drawn from a mixture of two multinomial component models, one content model and one style model, we propose a method named *Cartesian EM* that constructs content models and style models through Expectation Maximization and performs classification of the unknown content classes transductively. Our experiments on real-world datasets show the proposed method to be effective for *style independent text content classification.*

1 Introduction

Text classification [1] is a well established area of machine learning. A text classifier is first trained using documents with pre-assigned class labels and then offered test documents for which it must guess the best class labels.

We identify and address a special kind of text classification problem where every document is associated with a pair of independent labels (c_i, s_j) where $c_i \in C = \{c_1, ..., c_m\}$ and $s_j \in S = \{s_1, ..., s_n\}$. In other words, the label space is the Cartesian product of two independent sets of labels, $C \times S$, as shown in Figure 1. This problem setting extends the standard one-dimensional (1D) label space to two-dimensions (2D). Following the terminology of computational cognitive science and pattern recognition [2], we call C content labels and S style labels. Given a set of labeled training documents in $n-1$ styles $\{s_1, ..., s_{n-1}\}$ and a set of test documents

in a new style s_n, how should computers learn a classifier to predict the content class for each test document?

This machine learning problem is less explored, yet occurs frequently in practice. For example, consider a task of classifying academic web pages into several categories (such as "faculty" and "student"): one may have labeled pages from several universities (such as "Cornell", "Texas", and "Washington") and need to classify pages from another university (such as "Wisconsin"), where the categories can be considered as content classes and the universities can be regarded as style types. Other examples include: learning to classify articles from a new journal; learning to classify papers from a new author; learning to classify customer comments for a new product; learning to classify news or messages in a new period, and so on. The general problem is the same whenever we have a two (or more) dimensional label for each instance.

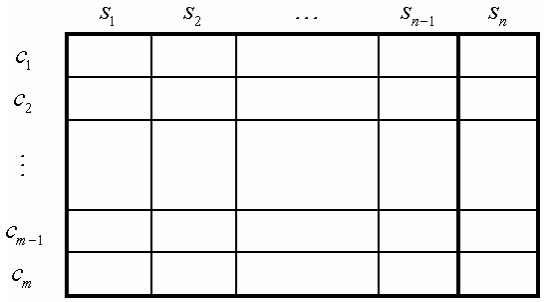

Fig. 1. The Cartesian product label space.

Since we care about the difference among content classes but not the difference among style types, it could be beneficial to separate the text content and style so that the classifier can focus on the discriminative content information but ignore the distractive style information. However, existing approaches to this problem, whether inductive or transductive, simply discard the style labels. Assuming that every document is generated using words drawn from a mixture of two multinomial component models, one content model and one style model, we propose a new method named *Cartesian EM* that constructs content models and style models through Expectation-Maximization [3] and performs classification transductively [4]. Our experiments on real-world datasets show that the proposed method can not only improve classification accuracy but also provide deeper insights about the data.

2 Learning with 1D Labels

2.1 Naïve Bayes

Naïve Bayes (NB) is a popular supervised learning algorithm for probabilistic text classification [5]. Though very simple, NB is competitively effective and highly efficient [6].

Given a set of labeled training documents $D = \{d_1, ..., d_{|D|}\}$, NB fits a generative model which is parameterized by θ and then applies it to classify unlabeled test documents. The generative model of NB assumes that a document d is generated by first choosing its class $c_i \in C = \{c_1, ..., c_m\}$ according to a prior distribution of classes, and then producing its words independently according to a multinomial distribution of terms conditioned on the chosen class [7].

The prior class probability $\Pr[c_i]$ can be estimated by

$$\Pr[c_i] = \left(\sum_{d \in D} \Pr[c_i \mid d] \right) \Big/ |D| , \tag{1}$$

where $\Pr[c_i \mid d]$ for a labeled document d is 1 if d is in class c_i or 0 otherwise.

With the "naïve" assumption that all the words in the document occur independently given the class, the conditional document probability $\Pr[d \mid c_i]$ can be estimated by

$$\Pr[d \mid c_i] = \prod_{w_k \in d} \left(\Pr[w_k \mid c_i] \right)^{n(w_k, d)} , \tag{2}$$

where $n(w_k, d)$ is the number of times word w_k occurs in document d. The conditional word probability $\Pr[w_k \mid c_i]$ can be estimated by

$$\Pr[w_k \mid c_i] = \frac{\eta + \sum_{d \in D} n(w_k, d) \Pr[c_i \mid d]}{\sum_{w \in V} \left(\eta + \sum_{d \in D} n(w, d) \Pr[c_i \mid d] \right)} , \tag{3}$$

where V is the vocabulary, and $0 < \eta \le 1$ is the Lidstone's smoothing parameter [6].

Given a test document d, NB classifies it into class

$$\hat{c}(d) = \arg\max_{c_i \in C} \Pr[c_i \mid d] . \tag{4}$$

The posterior class probability $\Pr[c_i \mid d]$ for an unlabeled document d can be computed via Bayes's rule:

$$\Pr[c_i \mid d] = \frac{\Pr[d \mid c_i] \Pr[c_i]}{\Pr[d]} = \frac{\Pr[d \mid c_i] \Pr[c_i]}{\sum_{c \in C} \Pr[d \mid c] \Pr[c]} . \tag{5}$$

2.2 1D EM

The parameter set θ of the above 1D generative model includes $\Pr[c_i]$ for all $c_i \in C$ and $\Pr[w_k \mid c_i]$ for all $w_k \in V$, $c_i \in C$. If the set of labeled documents is not large enough, NB may not be able to give a good estimation of θ therefore the classification accuracy may suffer. However, it has been shown that the model estimation could be improved by making use of additional unlabeled documents (such as the test documents). This is the idea of semi-supervised learning, i.e., learning from both labeled data and unlabeled data.

NB can be extended to semi-supervised learning by incorporating a set of unlabeled documents through Expectation-Maximization [8, 9]. The principle of

Expectation-Maximization (EM) will be further explained later in the next section. Since this EM based method is designed for the standard 1D label space, we call it *1D EM*. Please refer to [8, 9] for its detailed derivation.

```
Initialization: estimate θ from the labeled documents only,
using equations (1) and (3).
while ( θ has not converged ) {
        E-step: calculate the probabilistic class labels for
        the unlabeled documents based on the current θ, using
        equation (5).
        M-step: re-estimate θ from both the labeled documents
        and the unlabeled documents that have been assigned
        probabilistic class labels, using equations (1) and
        (3).
}
Classify the unlabeled documents using equation (4).
```

Fig. 2. The 1D EM algorithm.

3 Learning with 2D Labels

3.1 Cartesian Mixture Model

Let's consider the Cartesian product label space $C \times S$, where $C = \{c_1, ..., c_m\}$ and $S = \{s_1, ..., s_n\}$. In this 2D label space, every document $d \in D$ is associated with a pair of independent labels (c_i, s_j) where $c_i \in C$ and $s_j \in S$.

We introduce a 2D generative model, *Cartesian mixture model*, for this problem of learning with 2D labels. It naturally assumes that each content class $c_i \in C$ or style type $s_j \in S$ corresponds to a multinomial model. A document d is generated by first choosing its content class c_i and style type s_j according to a prior distribution of label-pairs, and then producing its words independently according to a mixture of two component models that correspond to c_i and s_j respectively. That is to say, every specific occurrence of word $w_k \in V$ in a (c_i, s_j) document is generated from either the content model $\Pr[w_k \mid c_i]$ or the style model $\Pr[w_k \mid s_j]$, though we do not know which one is actually responsible. Therefore the probability of a word $w_k \in V$ to occur in (c_i, s_j) documents can be calculated by

$$\Pr[w_k \mid c_i, s_j] = \lambda \Pr[w_k \mid c_i] + (1-\lambda) \Pr[w_k \mid s_j], \quad (6)$$

where $\lambda \in [0,1]$ is a parameter used for weighting the component models. In this paper, the same weighting parameter λ is used for all label-pairs, but our method can be easily extended to allow every label-pair have its own weighting parameter.

Since the content label and the style label are independent, we have

$$\Pr[c_i, s_j] = \Pr[c_i]\Pr[s_j]. \tag{7}$$

The prior content class probability $\Pr[c_i]$ can still be estimated using equation (1). The prior style type probability $\Pr[s_j]$ can be estimated similarly by

$$\Pr[s_j] = \left(\sum_{d \in D} \Pr[s_j \mid d]\right) \Big/ |D|, \tag{8}$$

where $\Pr[s_j \mid d]$ for a labeled document d is 1 if d is in style s_j or 0 otherwise.

Let $|d|$ denote the length of document d, and $o_{p,d}$ denote the word that occurs in the *p*-th position of document d. By imposing the extended "naïve" assumption that all the words in the document occur independently given the content class and the style type, the conditional document probability $\Pr[d \mid c_i, s_j]$ can be estimated by

$$\Pr[d \mid c_i, s_j] = \prod_{p=1}^{|d|} \Pr[o_{p,d} \mid c_i, s_j] = \prod_{w_k \in d} \left(\Pr[w_k \mid c_i, s_j]\right)^{n(w_k, d)}, \tag{9}$$

where $n(w_k, d)$ is the number of times word w_k occurs in document d. The conditional word probability $\Pr[w_k \mid c_i, s_j]$ given by equation (6) involves λ, $\Pr[w_k \mid c_i]$ and $\Pr[w_k \mid s_j]$ whose estimation will be discussed later.

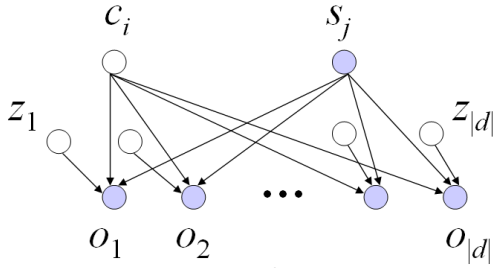

Fig. 3. The graphical model representation of a document as a Bayesian network.

In our problem setting (§1), the training documents are fully labeled, but the test documents are only half-labeled (we only know that the test document is in style s_n but we do not know which content class it belongs to). Given a test document d whose style is known to be s_n, we can predict its content class to be

$$\hat{c}(d) = \arg\max_{c_i \in C} \Pr[c_i \mid d, s_n]. \tag{10}$$

The posterior class probability $\Pr[c_i \mid d, s_n]$ for an half-labeled document d in style s_n can be calculated using Bayes's rule and equation (7):

$$\Pr[c_i \mid d, s_n] = \frac{\Pr[d \mid c_i, s_n]\Pr[c_i, s_n]}{\Pr[d \mid s_n]\Pr[s_n]} = \frac{\Pr[d \mid c_i, s_n]\Pr[c_i]}{\sum_{c \in C} \Pr[d \mid c, s_n]\Pr[c]}. \tag{11}$$

3.2 Cartesian EM

The parameter set θ of the above Cartesian mixture model includes $\Pr[c_i]$ for all $c_i \in C$, $\Pr[s_j]$ for all $s_j \in S$, $\Pr[w_k \mid c_i]$ for all $w_k \in V$, $c_i \in C$, $\Pr[w_k \mid s_j]$ for all $w_k \in V$, $s_j \in S$, and λ. One difficulty to estimate θ is that we would not be able to get the values of λ, $\Pr[w_k \mid c_i]$ and $\Pr[w_k \mid s_j]$ by using maximum likelihood estimation in a straightforward manner, because for every observed word occurrence we do not know exactly whether the content model or the style model generated it. Furthermore, we need to take the test documents into consideration while estimating θ (at least $\Pr[w_k \mid s_j]$), but for every test document we do not know exactly which content class it comes from.

The Expectation-Maximization (EM) algorithm [3] is a general algorithm for maximum likelihood estimation when the data is "incomplete". In this paper, we propose a new EM based method named *Cartesian EM* that constructs the Cartesian mixture model and applies it to predict the content classes of the test documents in a new style. Cartesian EM actually belongs to the family of transductive learning [4], a special kind of semi-supervised learning that the learning algorithm can see the set of test examples and make use of them to improve the classification accuracy on them.

A common method for estimating the model θ is maximum likelihood estimation in which we choose a θ that maximizes its likelihood (or equivalently log-likelihood) for the observed data (in our case the set of documents D and their associated labels):

$$\hat{\theta} = \arg\max_\theta L(\theta) = \arg\max_\theta \log L(\theta) = \arg\max_\theta \log \Pr[D \mid \theta]. \quad (12)$$

Given the model θ, we have

$$\log \Pr[D \mid \theta] = \sum_{d \in D} \log \Pr[d, c(d), s(d)], \quad (13)$$

where $c(d)$ and $s(d)$ stand for the actual (possibly unknown) content label and the actual style label of document d respectively.

For a fully-labeled document d, we know that $\Pr[c_i \mid d]$ is 1 if d is in content class c_i or 0 otherwise and $\Pr[s_j \mid d]$ is 1 if d is in style type s_j or 0 otherwise, therefore we can calculate the log-likelihood of d given the model θ using Bayes's rule and equation (7):

$$\begin{aligned} \log \Pr[d, c(d), s(d)] &= \sum_{i=1}^{m} \sum_{j=1}^{n} \Pr[c_i \mid d] \Pr[s_j \mid d] \log \Pr[d \mid c_i, s_j] + \\ &\sum_{i=1}^{m} \sum_{j=1}^{n} \Pr[c_i \mid d] \Pr[s_j \mid d] \log \Pr[c_i] + \sum_{i=1}^{m} \sum_{j=1}^{n} \Pr[c_i \mid d] \Pr[s_j \mid d] \log \Pr[s_j] \end{aligned}. \quad (14)$$

Furthermore, using equation (6) and equation (9), we get

$$\log \Pr[d \mid c_i, s_j] = \sum_{p=1}^{|d|} \log\left(\lambda \Pr[o_{p,d} \mid c_i] + (1-\lambda) \Pr[o_{p,d} \mid s_j]\right). \quad (15)$$

Now we see that there are two obstacles to estimating θ: one is that the test documents are only half-labeled hence equation (14) is not applicable to their log-likelihood computation, the other is that equation (15) contains logarithms of sums hence hard to maximize.

The basic idea of the EM algorithm is to augment our "incomplete" observed data with some latent/hidden variables so that the "complete" data has a much simpler likelihood function to maximize [10]. In our case, we introduce a binary latent variable for each content class $c_i \in C$ and each (half-labeled) test document d to indicate whether document d is in the content class c_i, i.e.,

$$y(c_i, d) = \begin{cases} 1 & \text{if document } d \text{ is in content class } c_i \\ 0 & \text{otherwise} \end{cases}. \quad (16)$$

Moreover, we introduce a binary latent variable for each word occurrence $o_{p,d}$ to indicate whether the word has been generated from the content model or the style mode,

$$z(p, d) = \begin{cases} 1 & \text{if } o_{p,d} \text{ is from the content model} \\ 0 & \text{otherwise} \end{cases}. \quad (17)$$

If the two types of latent variables are observed, the data is complete and consequently the log-likelihood function becomes much easier to maximize. For a half-labeled test document d, we know that the latent variable $y(c_i, d)$ is 1 if d is in content class c_i or 0 otherwise and $\Pr[s_j \mid d]$ is 1 if d is in style type s_j or 0 otherwise, therefore as in equation (14) we can calculate its log-likelihood given the model θ and the latent variables $y(c_i, d)$:

$$\begin{aligned} &\log \Pr[d, c(d), s(d)] = \sum_{i=1}^{m} \sum_{j=1}^{n} y(c_i, d) \Pr[s_j \mid d] \log \Pr[d \mid c_i, s_j] + \\ &\sum_{i=1}^{m} \sum_{j=1}^{n} y(c_i, d) \Pr[s_j \mid d] \log \Pr[c_i] + \sum_{i=1}^{m} \sum_{j=1}^{n} y(c_i, d) \Pr[s_j \mid d] \log \Pr[s_j] \end{aligned}. \quad (18)$$

In addition, with the help of the latent variables $z(p, d)$, equation (15) for computing $\log \Pr[d \mid c_i, s_j]$ can be re-written as

$$\log \Pr[d \mid c_i, s_j] = \sum_{p=1}^{|d|} \begin{pmatrix} z(p, d) \log\left(\lambda \Pr[o_p(d) \mid c_i]\right) + \\ \left(1 - z(p, d)\right) \log\left((1 - \lambda) \Pr[o_p(d) \mid s_j]\right) \end{pmatrix}, \quad (19)$$

because we assume that we know which component model has been used to generate each word occurrence.

The EM algorithm starts with some initial guess of the model $\theta^{(0)}$, then iteratively alternates between two steps, called the "E-step" (expectation step) and the "M-step" (maximization step) respectively [10]. In the E-step, it computes the expected log-likelihood for the complete data, or the so-called "Q-function" denoted by $Q(\theta; \theta^{(t)})$, where the expectation is taken over the computed conditional distribution of the latent variables given the current setting of model $\theta^{(t)}$ and the observed data. In the M-step, it re-estimates the model to be $\theta^{(t+1)}$ by maximizing the Q-function. Once we have a new generation of model parameters, we repeat the E-step and the M-step. This process continues until the likelihood converges to a local maximum.

The major computation to be carried out in the E-step is to estimate the distributions of latent variables, in our case, $y(c_i, d)$ and $z(p, d)$. For a half-labeled test document d in style s_n, we have

$$\Pr[y(c_i,d)=1]=\Pr[c_i \mid d,s_n]=\frac{\Pr[d \mid c_i,s_n]\Pr[c_i]}{\sum_{c\in C}\Pr[d \mid c,s_n]\Pr[c]} \tag{20}$$

via using equation (11), and obviously $\Pr[y(c_i,d)=0] = 1-\Pr[y(c_i,d)=1]$. For a word occurrence $o_{p,d}=w_k$ in a document d with label-pair (c_i,s_j), we have

$$\Pr[z(p,d)=1]=\frac{\lambda \Pr[w_k \mid c_i]}{\lambda \Pr[w_k \mid c_i]+(1-\lambda)\Pr[w_k \mid s_j]}, \tag{21}$$

and $\Pr[z(p,d)=0] = 1-\Pr[z(p,d)=1]$. Since the value of $\Pr[z(p,d)=1]$ given by the above equation is same for every word occurrence $o_{p,d}=w_k \in V$ in (c_i,s_j) documents, we introduce a new notation z_{ijk} to represent it.

The M-step involves maximizing the Q-function,

$$\theta^{(t+1)} = \arg\max_{\theta} Q(\theta;\theta^{(t)}). \tag{22}$$

In our case, the Q function can be obtained by combining the equations (13), (14), (18), (19), (20) and (21), and taking expectation over latent variables:

$$\begin{aligned} Q(\theta;\theta^{(t)}) &= \sum_{d\in D}\sum_{i=1}^{m}\sum_{j=1}^{n}\Pr[c_i \mid d]\Pr[s_j \mid d]E(d,c_i,s_j) \\ &+\sum_{d\in D}\sum_{i=1}^{m}\sum_{j=1}^{n}\Pr[c_i \mid d]\Pr[s_j \mid d]\log\Pr[c_i] \\ &+\sum_{d\in D}\sum_{i=1}^{m}\sum_{j=1}^{n}\Pr[c_i \mid d]\Pr[s_j \mid d]\log\Pr[s_j] \end{aligned}, \tag{23}$$

where

$$E(d,c_i,s_j)=\sum_{w_k\in d}\Big(n(w_k,d)\big(E(w_k,c_i)+E(w_k,s_j)\big)\Big), \tag{24}$$

$$E(w_k,c_i)=z_{ijk}\log\big(\lambda \Pr[w_k \mid c_i]\big), \tag{25}$$

$$E(w_k,s_j)=(1-z_{ijk})\log\big((1-\lambda)\Pr[w_k \mid s_j]\big). \tag{26}$$

The next model estimation $\theta^{(t+1)}$ should maximize $Q(\theta;\theta^{(t)})$, meanwhile the model parameters need to obey some inherent constraints such as

$$\sum_{w\in V}\Pr[w \mid c_i]=1. \tag{27}$$

The M-step turns out to be a constrained optimization problem. Using the Lagrange multiplier method, we can get an analytical solution to this problem. The derived update rules for the M-step are as follows. The prior content class probabilities $\Pr[c_i]$ are updated using equation (1), while the prior style type probabilities $\Pr[s_j]$ are kept unchanged. The conditional word probabilities should be adjusted using the following equations:

$$\Pr[w_k \mid c_i]=\left(\eta+\sum_{d\in D}\sum_{s\in S}Z_1(w_k,d,c_i,s)\right)\Bigg/\left(\sum_{w\in V}\left(\eta+\sum_{d\in D}\sum_{s\in S}Z_1(w,d,c_i,s)\right)\right) \tag{28}$$

where

$$Z_1(w_k,d,c_i,s_j)=n(w_k,d)\Pr[c_i \mid d]\Pr[s_j \mid d]z_{ijk}, \tag{29}$$

and similarly,

$$\Pr[w_k \mid s_j] = \left(\eta + \sum_{d \in D} \sum_{c \in C} Z_0(w_k, d, c, s_j) \right) \Bigg/ \left(\sum_{w \in V} \left(\eta + \sum_{d \in D} \sum_{c \in C} Z_0(w, d, c, s_j) \right) \right) \quad (30)$$

where

$$Z_0(w_k, d, c_i, s_j) = n(w_k, d) \Pr[c_i \mid d] \Pr[s_j \mid d](1 - z_{ijk}) . \quad (31)$$

Besides, the weight parameter should be re-estimated by

$$\lambda = \left(\sum_{w \in V} \sum_{d \in D} \sum_{c \in C} \sum_{s \in S} Z_1(w, d, c, s) \right) \Bigg/ \left(\sum_{w \in V} \sum_{d \in D} \sum_{c \in C} \sum_{s \in S} \left(Z_1(w, d, c, s) + Z_0(w, d, c, s) \right) \right). \quad (32)$$

The EM algorithm is essentially a hill-climbing approach, thus it can only be guaranteed to reach a local maximum. When there are multiple local maximums, whether we will actually reach the global maximum depends on where we start: if we start at the "right hill", we will be able to find a global maximum. In our case, we ignore the style labels and use the 1D estimation equations (1) and (3) to initialize $\Pr[c_i]$ and $\Pr[w_k \mid c_i]$ from the training documents, just as in the standard NB algorithm. The initial values of $\Pr[s_j]$ and $\Pr[w_k \mid s_j]$ can be estimated similarly by ignoring the class labels, but from both the training documents and the test documents, using equation (8) and

$$\Pr[w_k \mid s_j] = \left(\eta + \sum_{d \in D} n(w_k, d) \Pr[s_j \mid d] \right) \Bigg/ \left(\sum_{w \in V} \left(\eta + \sum\nolimits_{d \in D} n(w, d) \Pr[s_j \mid d] \right) \right). \quad (33)$$

The initial value of the weighting parameter λ is simply set to $1/2$ that puts equal weights to the component models.

Initialization: set $\lambda = 1/2$, and estimate the probabilities $\Pr[c_i]$, $\Pr[w_k \mid c_i]$, $\Pr[s_j]$, $\Pr[w_k \mid s_j]$ just as in the 1D situation, using equations (1), (3), (8), (33).

while (θ has not converged) **{**

E-step: calculate $\Pr[c_i \mid d]$ for the half-labeled test documents using equation (20), and calculate z_{ijk} using equation (21), based on the current θ .

M-step: re-estimate θ using equations (28), (30), (32), with the help of the latent variables.

}

Classify the half-labeled test documents using equation (10).

Fig. 4. The Cartesian EM algorithm.

4 Experiments

We have conducted experiments on three real-world datasets to evaluate the effectiveness of the proposed Cartesian EM method for text classification in the 2D problem setting (stated in §1). Three methods, NB, 1D EM and Cartesian EM (C. EM), were compared in terms of classification accuracy. The Lidstone smoothing parameter was set to a good value $\eta = 0.1$ [6], and document frequency (df) based feature selection [11] were performed to let NB achieve optimal average performance on each dataset. The document frequency has been shown to have similar effect as information gain or chi-square test in feature selection for text classification.

The WebKB dataset (http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/) contains manually classified Web pages that were collected from the computer science departments of four universities ("Cornell", "Texas", "Washington" and "Wisconsin") and some other universities ("misc."). In our experiments, each category is considered as a content class, and each source university is considered as a style type because each university's pages have their own idiosyncrasies. Only pages in the top four largest classes were used, namely "student", "faculty", "course" and "project". Furthermore, unlike the popular setting, the pages from the "misc." universities were not used either, because we thought it made little sense to train a style model for them. All pages were pre-processed using the Rainbow toolkit [12] with the options "--skip-header --skip-html --lex-pipe-command=tag-digits --no-stoplist --prune-vocab-by-doc-count=3". The experimental results on this dataset are shown in Table 1. The weighting parameter values found by Cartesian EM were consistently around 0.75.

The SRAA dataset (http://www.cs.umass.edu/~mccallum/code-data.html) is a collection of 73,218 articles from four newsgroups (simulated-auto, simulated-aviation, real-auto and real-aviation). In our experiments, "auto" and "aviation" are considered as content classes and "simulated" and "real" are considered as style types. All articles were pre-processed using the Rainbow toolkit [12] with the option "--skip-header --skip-html --prune-vocab-by-doc-count=200". The experimental results on this dataset are shown in Table 1. The weighting parameter values found by Cartesian EM were consistently around 0.95. It is also possible to take "simulated" and "real" as classes while "auto" and "aviation" as styles, but in our experiments, Cartesian EM did not work well in that situation. We think the reason is that the actual weight of "auto" and "aviation" models are overwhelming (around 0.95) so that Cartesian EM would drift away and consequently fail to get good discrimination between "simulated" and "real". This is a known pitfall of the EM algorithm. Therefore Cartesian EM may be not suitable when the text content is significantly outweighed by text style. It may be possible to detect this situation by observing λ and back off to a simpler method like NB when λ is small.

The 20NG dataset (http://people.csail.mit.edu/people/jrennie/20Newsgroups) is a collection of approximately 20,000 articles that were collected from 20 different newsgroups [13]. The "bydate" version of this dataset along with its train/test split was used. In our experiments, each newsgroup is considered as a content class, and the time period before and after the split point are considered as two style types. It is realistic and common to train a classifier on documents before a time point and then

test it on documents thereafter. All articles were pre-processed using the Rainbow toolkit [12] with the option "--prune-vocab-by-occur-count=2 --prune-vocab-by-doc-count=0". The experimental results on this dataset are shown in Table 1. The weighting parameter value found by Cartesian EM was about 0.95.

To sum up, Cartesian EM compared favorably with NB and 1D EM in our experiments. In the case where Cartesian EM improved performance, the improvements over the standard 1D-EM and Naive Bayes on these three datasets are all statistically significant (P-value < 0.05), using the McNemar's test. It is able to yield better understanding (such as the weight of content/style) as well as increase classification accuracy.

Table 1. Experimental results.

Dataset	Test Style	NB	1D EM	C. EM
WebKB	Cornell	81.42%	**84.96%**	**84.96%**
	Texas	81.75%	82.94%	**83.33%**
	Washington	77.25%	79.22%	**80.00%**
	Wisconsin	82.79%	81.17%	**84.42%**
SRAA	simulated	80.11%	91.85%	**94.39%**
	real	91.99%	87.00%	**92.68%**
20NG	new articles	79.55%	80.92%	**81.78%**

5 Related Works

The study of separating content and style has a long history in computational cognitive science and pattern recognition. One typical work in this area is [2]. Our work exploits the characteristic of text data, and the proposed Cartesian EM method is relatively more efficient than the existing methods.

Various mixture models have been used in different text classification problems. However, most of them assume that a document is generated by only one component model [14], while our Cartesian mixture model assumes that a document is generated by two multinomial component models. In [15], a mixture model is proposed for multi-label text classification, where each document is assumed to be generated by multiple multinomial component models, one per document label. In [16], a mixture model is proposed for relevance feedback, where the feedback documents are assumed to be generated by two multinomial component models, one known background model and one unknown topic model, combined via a fixed weight. The Cartesian mixture model is different with these existing works as it is designed for documents with 2D labels and it works in the context of transductive learning.

One straightforward way to extend 1D EM to 2D scenario is to simply regard each label-pair (c, s) as a pseudo 1D label. In other words, each cell in the Cartesian product label matrix (as shown in Figure 1) is associated with a multinomial model. That method named EM2D has been proposed for the problem of learning to integrate

web taxonomies [17]. However, EM2D is not able to generalize content classifiers to new styles thus not applicable to our problem. In contrast, Cartesian EM assumes that each row or column in the Cartesian product label matrix corresponds to a multinomial model and the observations (cells) are generated by the interaction between content models (rows) and style models (columns). This is to simulate the situation where some words in the document are used for style purposes while other words in the same document are used to describe the content.

6 Conclusion

The ability of human being to separate content and style is amazing. This paper focuses on the problem of *style-independent text content classification*, and presents an EM based approach, Cartesian EM, that has been shown to be effective by experiments on real-world datasets.

References

1. Sebastiani, F.: Machine Learning in Automated Text Categorization. ACM Computing Surveys **34** (2002) 1-47
2. Tenenbaum, J.B., Freeman, W.T.: Separating Style and Content with Bilinear Models. Neural Computation **12** (2000) 1247-1283
3. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum Likelihood from Incomplete Data via the EM Algorithm. Journal of the Royal Statistical Society, Series B **39** (1977) 1-38
4. Vapnik, V.N.: Statistical Learning Theory. Wiley (1998)
5. Mitchell, T.: Machine Learning. McGraw Hill (1997)
6. Agrawal, R., Bayardo, R., Srikant, R.: Athena: Mining-based Interactive Management of Text Databases. Proceedings of the 7th International Conference on Extending Database Technology (EDBT), Konstanz, Germany (2000) 365-379
7. McCallum, A., Nigam, K.: A Comparison of Event Models for Naive Bayes Text Classification. AAAI-98 Workshop on Learning for Text Categorization, Madison, WI (1998) 41-48
8. Nigam, K., McCallum, A., Thrun, S., Mitchell, T.: Learning to Classify Text from Labeled and Unlabeled Documents. Proceedings of the 15th Conference of the American Association for Artificial Intelligence (AAAI), Madison, WI (1998) 792-799
9. Nigam, K., McCallum, A., Thrun, S., Mitchell, T.: Text Classification from Labeled and Unlabeled Documents using EM. Machine Learning **39** (2000) 103-134
10. Zhai, C.: A Note on the Expectation-Maximization (EM) Algorithm. (2004)
11. Yang, Y., Pedersen, J.O.: A Comparative Study on Feature Selection in Text Categorization. Proceedings of the 14th International Conference on Machine Learning (ICML), Nashville, TN (1997) 412-420
12. McCallum, A.: Bow: A Toolkit for Statistical Language Modeling, Text Retrieval, Classification and Clustering. (1996)
13. Lang, K.: NewsWeeder: Learning to Filter Netnews. Proceedings of the 12th International Conference on Machine Learning (ICML), Tahoe City, CA (1995) 331-339
14. Pavlov, D., Popescul, A., Pennock, D.M., Ungar, L.H.: Mixtures of Conditional Maximum Entropy Models. Proceedings of the 20th International Conference on Machine Learning (ICML), Washington DC, USA (2003) 584-591

15.McCallum, A.: Multi-Label Text Classification with a Mixture Model Trained by EM. AAAI'99 Workshop on Text Learning (1999)
16.Zhai, C., Lafferty, J.D.: Model-based Feedback in the Language Modeling Approach to Information Retrieval. Proceedings of the 10th ACM International Conference on Information and Knowledge Management (CIKM), Atlanta, GA (2001) 403-410
17.Sarawagi, S., Chakrabarti, S., Godbole, S.: Cross-Training: Learning Probabilistic Mappings between Topics. Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), Washington DC, USA (2003) 177-186