TOPIC IDENTIFICATION IN DYNAMICAL TEXT BY EXTRACTING MINIMUM COMPLEXITY TIME COMPONENTS

Ella Bingham

Neural Networks Research Centre, Helsinki University of Technology
P.O. Box 5400, FIN-02015 HUT, Finland, tel. +358 9 451 5282, fax +358 9 451 3277
ella@iki.fi, http://www.cis.hut.fi/projects/ica/

ABSTRACT

The problem of analysing dynamically evolving textual data has recently arisen. An example of such data is the discussion appearing in Internet chat lines. In this paper a recently introduced source separation method, termed as *complexity pursuit*, is applied to the problem. The method is a generalisation of projection pursuit to time series and it is able to use both spatial and temporal dependency information in separating the topics of the discussion. Experimental results on chat line and newsgroup data demonstrate that the minimum complexity time series indeed do correspond to meaningful topics inherent in the dynamical text data, and also suggest the applicability of the method to query-based retrieval from a temporally changing text stream. The complexity pursuit method is compared to several ICA-type algorithms for time series.

1. INTRODUCTION

In times of huge information flow especially in the Internet, there is a strong need for automatic textual data analysis tools. There are a number of algorithms and methods developed for text mining from static text collections [1]. The WEBSOM[1] is a document clustering and visualisation method [2]; its probabilistic counterpart has been presented e.g. in [3]. Another basic algorithm is Latent Semantic Indexing (LSI) [4] in which the data is projected onto a subspace spanned by the most important singular vectors of the data matrix; its probabilistic counterparts have been presented by Hofmann [5] and Papadimitriou [6]. LSI uses only second-order moments of the data and neglects any higher order correlations, so independent component analysis (ICA) -type algorithms are in this sense a possible step forward. First approaches of using ICA in the context of text data were presented by Isbell and Viola [7], Kolenda et al. [8] and Kabán and Girolami [9]. In these approaches, the textual data is not a dynamic time series but rather an instantaneous mixture of independent topics. The underlying assumption which we also adopt is that the textual data consists of some more or less independent topics. In the text retrieval parlance, a *topic* is a probability distribution on the universe of terms; it is typically concentrated on terms that might be used when discussing a particular subject. In this paper, the word “topic” also refers to a hidden, more or less independent random variable with time structure. Thus we can analyze the “independent components” of text both by the terms they concentrate on, and by their activity in time.

Recently the issue of analyzing *dynamically evolving* textual data has arisen, and investigating appropriate tools for this task is of practical importance. An example of a dynamically evolving discussion is found in the Internet relay chat rooms. In these chat rooms daily news topics are discussed and the topic of interest changes according to participants’ contributions. The online text stream can thus be seen as a time series, and methods of time series processing may be used to extract the underlying characteristics — here the topics — of the discussion. Kolenda and Hansen [10] employ Molgedey and Schuster’s [11] ICA algorithm for the identification of the dynamically evolving topics. Molgedey and Schuster’s algorithm is an early separation algorithm which uses temporal information and does not require any higher order moments for the source separation problem. Kabán and Girolami [12] have recently presented an HMM-type algorithm for the topographic visualization of time-varying data.

In this paper a recently introduced powerful separating method is applied to the problem of extracting the topics of a dynamically evolving discussion. The method presented by Hyvärinen, termed as complexity pursuit [13], is a generalization of projection pursuit [14] to time series and it is able to exploit both spatial and temporal dependency information in separating the topics. Complexity pursuit is a method for finding interesting projections of time series, the interestingness being measured as a short coding length of the projection. Projection pursuit neglects any time-dependency information and defines interestingness as nongaussianity. Complexity pursuit uses both information-theoretic measures and time-correlations of the data, which makes it more powerful and motivates its use in the task approached in this paper.

This paper is organized as follows. Section 2 describes how the chat line data is generated and preprocessed. Section 3 provides an introduction to complexity pursuit. Section 4 presents experimental results on using the complexity pursuit algorithm on chat line and newsgroup data. Finally, some conclusions are drawn in Section 5.

2. CHAT LINE DATA

Often the characteristics of the textual data of interest change over time. Such dynamical data dan be found e.g. in the online news services. Our example of a dynamically evolving text is chat line data.

The discussion found in chat lines on the Internet is an ongoing stream of text generated by the chat participants and the chat line moderator. To analyze it using data mining methods a convenient technique is to split the stream into windows that may be overlapping if desired. Each such window can now be viewed as one document.

[1] See http://websom.hut.fi/websom/

We employ the vector space model [15] for representing the documents, although other models can be considered. In the vector space model, each document forms one T-dimensional vector where T is the number of distinct terms in the vocabulary. The i-th element of the vector indicates (some function of) the frequency of the i-th vocabulary term in the document. The data matrix $\mathbf{X}$, also called the term by document matrix, contains the document vectors as its columns and is of size $T \times N$ where N is the number of documents. We will write $\mathbf{X}$ when referring to the whole set of data vectors and $\mathbf{x}$ when referring to one of them; thus $\mathbf{X} = (\mathbf{x}(t))$, $t = 1, \ldots, N$. Similarly, a vector $\mathbf{w}$ is a direction onto which the data may be projected, and the matrix $\mathbf{W} = (\mathbf{w}_i)$ contains these directions as column vectors.

As a preprocessing step we compute the LSI of the data matrix $\mathbf{X}$, that is, the singular value decomposition (SVD) $\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T$ where orthogonal matrices $\mathbf{U}$ and $\mathbf{V}$ contain the left and right singular vectors of $\mathbf{X}$, respectively, and the pseudodiagonal matrix $\mathbf{D}$ contains the singular values of $\mathbf{X}$. The term by document matrix — which may be of very high dimension — is then projected onto a smaller dimensional subspace spanned by K left singular vectors in $\mathbf{U}_K$ corresponding to K largest singular values in the diagonal matrix $\mathbf{D}_K$:

$$\mathbf{Z} = \mathbf{D}_K^{-1}\mathbf{U}_K^T\mathbf{X}_K = \mathbf{V}_K^T \tag{1}$$

where $\mathbf{X}_K = \mathbf{U}_K\mathbf{D}_K\mathbf{V}_K^T$ is an approximation of $\mathbf{X}$. Thus the observations in $\mathbf{X}$ are represented as linear combinations of some orthogonal latent features. The new data matrix $\mathbf{Z} = \mathbf{V}_K^T$ and its columns $\mathbf{z}(t)$, $t = 1, \ldots, N$ are now the inputs for the algorithm that will be described in Section 3. The time-structure of the topics of the discussion or the minimum complexity projections can be found by projecting $\mathbf{Z}$ onto the directions $\mathbf{W} = (\mathbf{w}_1 \cdots \mathbf{w}_M)$ given by the algorithm. It is often advantageous to compute the LSI projection onto a larger dimensionality $K > M$ and then to find M minimum complexity projections. To represent the estimated topics in the term space, the original data is first projected onto the LSI term space by

$$\mathbf{Z}_{term} = \mathbf{D}_K^{-1}\mathbf{V}_K\mathbf{X}_K^T = \mathbf{U}_K^T \tag{2}$$

and then projected onto the directions $\mathbf{W}$ found earlier.

The LSI (SVD) preprocessing is computationally the most demanding part of the problem, of order $O(NTc)$ for a sparse $T \times N$ data matrix with c nonzero entries per column (here, c is the number of vocabulary terms present in one document). If new data is obtained after the LSI has been computed, the decomposition can be updated by folding-in documents or terms: the LSI projection of a new document vector $\mathbf{x}_{new}$ is $\mathbf{z}_{new} = \mathbf{x}_{new}\mathbf{U}_K\mathbf{D}_K^{-1}$. Similarly, the projection of a new term vector $\mathbf{x}_{new}^{term}$ (a new row in $\mathbf{X}$) is $\mathbf{z}_{new}^{term} = \mathbf{x}_{new}^{term}\mathbf{V}_K\mathbf{D}_K^{-1}$ [4].

3. THE COMPLEXITY PURSUIT ALGORITHM

Complexity pursuit [13] is a recently developed, computationally simple algorithm for separating interesting components from time series. It is an extension of projection pursuit [14] to time series data and also closely related to ICA. Projection pursuit seeks for directions in which the data has an interesting, structured distribution, the interestingness being understood as nongaussianity — neglecting any time-dependency information that may exist in the data. ICA, on the other hand, finds statistically independent directions. It is to be noted that under some restrictions, it is also possible to estimate the independent components using the time dependency information alone (see e.g. [16], [11]); however the early algorithms as that proposed in [11] do not utilize the distribution of the data in obtaining the separation. A heuristic way of combining both of these estimation criteria (nongaussianity and time-correlations) has been proposed in the JADE$_{TD}$ algorithm [17]. However, complexity pursuit combines these criteria in a principled way by employing the information theoretical concept of Kolmogoroff complexity [18] and developing a simple approximation of it. In complexity pursuit the structure of the projected time series is measured as the coding complexity. Time series which have the lowest coding complexity are considered the most interesting. Another method of separating independent sources in time series has recently been presented by Stone [19]; in his approach, it is assumed that the source signals are more predictable than any linear mixture of them. In Section 4 we shall present experimental results on using complexity pursuit, JADE$_{TD}$, ordinary ICA and the methods presented in [19] and [10].

The model assumes that the observations $\mathbf{x}(t)$ are linear mixtures of some latent components: $\mathbf{x} = \mathbf{A}\mathbf{s}$ where $\mathbf{x} = (x_1, \ldots, x_T)$ is the vector of observed random variables, $\mathbf{s} = (s_1, \ldots, s_M)$ is the vector of independently predictable latent components, and $\mathbf{A}$ is an unknown constant mixing matrix. A separate autoregressive model is assumed to model each component $s_i = \mathbf{w}_i^T\mathbf{z}$ (where $\mathbf{w}_i$ corresponds to an estimate of a row of $\mathbf{A}^{-1}$); as a simple special case of the algorithm presented in [13], we employ a first order autoregressive (AR) process $\hat{s}(t) = \alpha s(t - \tau)$, each s_i having its own parameter α. The approximate Kolmogoroff complexity of the residuals $\delta s(t) = s(t) - \hat{s}(t)$ (using the predictive coding of the components) [13]

$$\begin{aligned}\hat{K}\big(\delta(\mathbf{w}^T\mathbf{x}(t))\big) =& E\big\{G\big(\frac{1}{\sigma_\delta(\mathbf{w})}\mathbf{w}^T(\mathbf{x}(t) - \alpha\mathbf{x}(t-\tau))\big)\big\} \\ &+ \log\sigma_\delta(\mathbf{w})\end{aligned} \tag{3}$$

is then minimized, where G is the negative log-density of the residuals. In the above formula it is emphasized that the values of α and the residual standard deviation σ_δ depend on the projection vector $\mathbf{w}$ only. An additional constraint $E\{(\mathbf{w}^T\mathbf{x}(t))^2\} = 1$ is also required to fix the scale of the projection. In the right hand side of Formula (3) the first term measures the contribution of the nongaussianity, and the second term the contribution of the variance to the entropy of the residual. Minimizing the first term would find the direction of maximal nongaussianity of the residual, and minimizing the second term the direction of maximum autocovariances, i.e. maximum time-dependencies [13].

In our application the latent time-components s_i will model the evolving topics of the discussion. To find the minima of (3), the data is first whitened by LSI as described in the previous section. We denote by $\mathbf{z}(t)$ this preprocessed data, and $\mathbf{w}$ now corresponds to an estimate of a row of the mixing matrix $\mathbf{B}$ for whitened data: $\mathbf{z} = \mathbf{B}\mathbf{s}$. At every step of the algorithm, the autoregressive constant $\alpha(\mathbf{w})$ for the time series given by $\mathbf{w}^T\mathbf{z}(t)$ is first found using [13]

$$\hat{\alpha} = \mathbf{w}^T E\{\mathbf{z}(t)\mathbf{z}(t-\tau)\}\mathbf{w} \tag{4}$$

Then the gradient update of $\mathbf{w}$ that minimizes (3) is the following [13]:

$$\begin{aligned}\mathbf{w} \leftarrow& \mathbf{w} - \mu E\big\{(\mathbf{z}(t) - \alpha(\mathbf{w})\mathbf{z}(t-\tau))\cdot \\ & \qquad g\big(\mathbf{w}^T(\mathbf{z}(t) - \alpha(\mathbf{w})\mathbf{z}(t-\tau))\big)\big\}\end{aligned} \tag{5}$$

$$\mathbf{w} \leftarrow \mathbf{w}/||\mathbf{w}|| \tag{6}$$

The function g is chosen according to the probability distribution of the residual: to be exact, g should be the negative score function p'/p of the density of the residual, as g is the derivative of G in (3).

To estimate several projections one can either use a deflation scheme, or estimate all projections simultaneously in a symmetric manner and use orthogonal decorrelation $\mathbf{W} \leftarrow \sqrt{(\mathbf{W}\mathbf{W}^T)^{-1}}\mathbf{W}$ instead of (6). In the deflationary approach, after the estimation of p projections, we run the algorithm for $\mathbf{w}_{p+1}$ and after every iteration step subtract from $\mathbf{w}_{p+1}$ the projections of the previously estimated p vectors, and then renormalize $\mathbf{w}_{p+1}$. This kind of Gram-Schmidt decorrelation is presented e.g. in [20].

The algorithm scales as $O(NK^2M)$ on preprocessed data; this is linear in the number of observations N as typically $K << N$ and $M \leq K$.

4. EXPERIMENTAL RESULTS

4.1. Experimental setting

The chat line data used in our experiments was collected from the CNN Newsroom chat line[2]. A contiguous stream of almost 24 hours of discussion of 3200 chat participants, contributing 25 000 comment lines, was recorded on January 18th, 2001. The data was cleaned by omitting all user names and non-user generated text. The remaining text stream was split into windows of 12 rows (about 130 words)[3]; subsequent windows shared an overlap of 66%. From these windows a term histogram was generated using the Bow toolkit[4], stemming, stop-word removal and tf-idf (term frequency – inverse document frequency) term weighting being part of the process. This resulted in a term by document matrix $\mathbf{X}$ of size $T \times N = 5000 \times 7430$.

The binary valued coding of the term by document matrix — i-th entry of a document vector was 1 if the i-th vocabulary term was present in the document, and 0 otherwise — was used in the experiments. Binary coding avoids serious outliers in the data and is computationally simple; also, it may be suitable for short documents where the size of the vocabulary is large, such as short windows of chat line discussion.

The text document data is typically very sparse; in our chat line data, on the average, each document had about 40 vocabulary terms and only 0.65% of the entries of the data matrix $\mathbf{X}$ were nonzero. Sparsity gives additional computational savings, so we did not make the data zero mean as is often done in the context of ICA-type algorithms — that would have destroyed the sparsity.

The LSI of order $K = 100$ was computed as a preprocessing step as described in Section 2. The choice of the number of topics M is somewhat arbitrary. It has been proved in [6] that if the data has a clear clustered structure, it is enough to choose M equal to the number of clusters. In our application the case is somewhat more complex, because more than one topic may be discussed at any one time, and real-life data may not have clear clusters. The identified topics lend themselves easily to human evaluation if they are presented in the term space as described in the end of Section 2 and the most representative words associated with each $\mathbf{w}_i$, $i = 1, \ldots, M$ are listed. Similar complementing architectures are used in the ICA of functional magnetic resonance imaging (fMRI) and image recognition, and in the context of textual document analysis [9]. Note that while in the static case the projections of both $\mathbf{X}$ and $\mathbf{X}^T$ could be used for training in the ICA algorithm (see [9] for derivation), in our case the terms have no time structure and they will be employed in the visualization phase only.

[2] `http://www.cnn.com/chat/channel/cnn_newsroom`

[3] The complexity pursuit method does not seem to be very sensitive to the window length.

[4] `http://www.cs.cmu.edu/~mccallum/bow/`

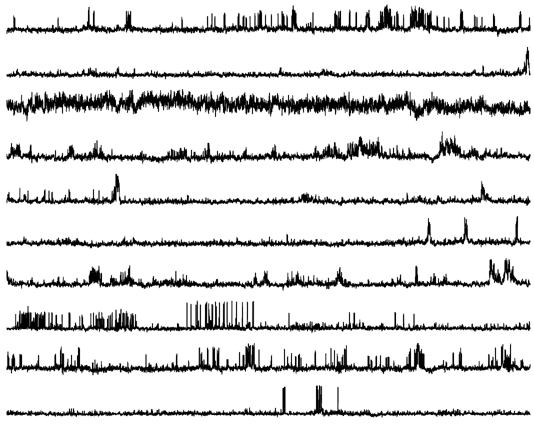

Fig. 1. Activity of topics (vertical axis) in each chat window (horizontal axis). $g(u) = \tanh(u)$ and $\tau = 5$ were used in Formula (5). The uppermost time series corresponds to topic 1, the second to topic 2 etc.

It should also be noted that the projections $\mathbf{w}^T\mathbf{Z}$ that represent the latent topics of discussion are found by the complexity pursuit algorithm up to permutation and scaling, as is always the case in the context of ICA-type algorithms. Therefore some prior knowledge based post-processing is necessary for interpreting the results. We know that the terms belonging to each topic should have a positively skewed distribution — there are often only a few terms that occur very frequently and correspondingly a large number of seldom occurring terms. We must change the sign of the negatively skewed projections $\mathbf{w}^T\mathbf{Z}$ so that their distribution becomes positively skewed.

Our experiments showed that choosing a first order AR model $\hat{s}(t) = \alpha s(t - \tau)$ was successful and that lags of e.g. $\tau = 1$ and $\tau = 5$ were the most suitable — in a typical discussion in a chat line, the participants' on-line contributions only depend on a few previous comments which in our data are recorded in the preceding text windows. The best results were obtained when the nonlinearity g in Formula (5) was chosen as $g(u) = \tanh(u)$, corresponding to imposing a "cosh" prior on the residuals $\mathbf{z}(t) - \alpha\mathbf{z}(t - \tau)$. We have also previously [21] had good results with the simple $g(u) = \text{sign}(u)$ nonlinearity that corresponds to a Laplace prior on the residuals.

4.2. Results on chat line data

We estimated $M = 10$ topics of chat line discussion simultaneously, using the orthogonal decorrelation presented in the end of Section 3. Figure 1 shows how different topic time series $\mathbf{w}_i^T\mathbf{Z}$, $i = 1, \ldots, M$ are activated at different times. We can see that the topics clearly are autocorrelated in time.

We now turn to analyze the projections $\mathbf{w}_i^T\mathbf{Z}_{term}$ of the terms onto minimum complexity directions. This information is com-

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8	Topic 9	Topic 10
jackson	site	bush	religion	violenc	flag	california	join	tax	free
sharpton	web	ashcroft	god	report	move	power	discuss	cut	liber
child	net	vote	jesu	youth	citi	electr	est	exempt	opinion
stori	word	kennedi	bibl	children	ncaa	energi	tonight	monei	religion
drudg	parent	presid	religi	gun	offici	blackout	room	gop	form
rainbow	nanni	cnn	life	point	atlanta	state	studio	hous	polit
monei	internet	time	follow	home	count	deregul	cnn	congress	conserv
mistress	block	gore	read	drug	game	compani	conserv	pay	birth
coalition	kid	question	stori	famili	night	crisi	american	interest	philosophi
tonight	system	elect	univers	satcher	georgia	price	nea	recess	establish
pregnant	access	god	exist	health	chang	plant	union	payer	narrow
affair	child	senat	faith	risk	lose	util	keen	secur	restrict
black	base	power	man	factor	confeder	order	type	henri	independ
chenei	chat	thing	book	surgeon	hehe	home	chat	hypocrit	orthodox
jessi	page	fact	earth	prevent	chenei	cost	newsroom	hyde	bound

Table 1. Keywords of chat line discussion topics related to the time series in Figure 1.

plementary to that revealed by analyzing the document projections $\mathbf{w}_i^T\mathbf{Z}$, and offers an informative way of visualizing the results. By listing the terms corresponding to the highest values of $\mathbf{w}_i^T\mathbf{Z}_{term}$ we get a list of keywords for the i-th topic. The keywords are listed in Table 1 and it is seen that each keyword list indeed characterizes one distinct topic quite clearly. Due to polysemy, the same word may appear in more than one topic. Topic 1 deals with Jesse Jackson and his illegitimate child, topic 2 is about parental control over children's web usage and topic 3 is a general discussion about G.W. Bush. Topic 4 is a religious discussion, topic 5 deals with problems of the youth such as violence and drug abuse, and topic 6 is about the controversial flag of the state of Georgia, US, due to which the NCAA basketball games risked cancellation in Atlanta. Topic 7 involves the energy shortage in California, topic 8 corresponds to comments given by the chat line moderator, topic 9 is about taxation and topic 10 deals with the values of the politicians.

The choice of the number of estimated topics is somewhat flexible. For example, estimating $M = 6$ topics would have given keyword lists similar to topics 2, 3, 4, 5, 6 and 7 in Table 1.

The evaluation of the results based on the keywords is rather subjective. Numerical measures are hard to find as the chat line discussion data is not labeled. For this reason we present results on labeled data in the next section.

4.3. Results on newsgroup data

In this section we present experimental results on newsgroup data where consequtive newsgroup articles are divided into overlapping windows similarly to what was done with the chat line data. Newsgroup data is often similar to chat line data in the sense that subsequent articles share a vague topic and the topic changes in time. The newsgroup data is labeled (as articles are from distinct newsgroups) and so we are able to quantitatively assess the separation results obtained by our algorithm and some other methods. The data is from four newsgroups of the 20 Newsgroup corpus[5]: sci.crypt, sci.med, sci.space and soc.religion.christian. The newsgroup articles, about 1000 from each group, were split to windows of 20 rows (excluding the headers) with 50% overlap between neighboring windows. Again, a binary term histogram was generated but this time no stemming was used as newsgroup language tends to be quite precise, in contrast to chat line discussions. The size of the data matrix $\mathbf{X}$ was 5000 terms by 4695 documents.

[5] http://www.cs.cmu.edu/~textlearning

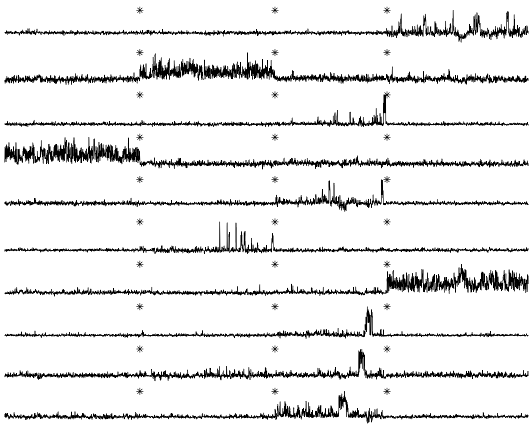

Fig. 2. Activity of topics (vertical axis) in each newsgroup window (horizontal axis). $g(u) = \tanh(u)$ and $\tau = 1$ were used in Formula (5). The asterisks denote the newsgroup borders: sci.crypt, sci.med, sci.space and soc.religion.christian. The uppermost time series corresponds to topic 1, the second to topic 2 etc.

Figure 2 shows the topic time series $\mathbf{w}^T\mathbf{Z}$ found using the complexity pursuit algorithm. 10 minimum-complexity directions $\mathbf{w}$ were estimated. The asterisks denote the borders between different newsgroups. It can be seen that each estimated topic time series corresponds to one of the newsgroups, or part of it. The keywords are seen in Table 2, and they also nicely correspond to newsgroup labels: topics 1 and 7 characterize different aspects discussed in soc.religion.christian, topics 2 and 6 in sci.med, topics 3, 5, 8, 9 and 10 in sci.space and topic 4 is the only topic from sci.crypt.

The classification error of the newsgroup documents is computed in the following way: The topic time series $\mathbf{w}_i^T\mathbf{Z}$ are first normalized to unit variance. Then each time series is considered

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8	Topic 9	Topic 10
homosexu	medic	burst	kei	orbit	food	god	venu	theori	launch
paul	diseas	rememb	chip	station	human	christ	soviet	physic	space
christian	treatment	matter	encrypt	shuttl	effect	christian	planet	book	satellit
sexual	bank	grb	clipper	power	studi	church	probe	univers	commerci
discuss	patient	red	govern	option	brain	jesu	mission	larson	market
sex	scienc	black	secur	flight	glutam	sin	mile	gener	servic
evid	problem	galaxi	law	space	review	lord	surfac	physicist	project
activ	doctor	dark	phone	design	singl	faith	kilomet	unifi	plan
cor	medicin	star	algorithm	engin	diet	bibl	earth	relat	orbit
refer	skeptic	shift	system	modul	industri	love	venera	comprehens	cost
vers	intellect	galact	public	control	paper	scriptur	atmospher	motion	note
issu	chastiti	halo	bit	manag	level	life	lander	natur	vehicl
male	result	absorpt	escrow	capabl	check	psalm	craft	develop	develop
passag	food	isotrop	nsa	present	sensit	prayer	balloon	result	technologi
interpret	effect	gamma	secret	team	blood	john	vega	light	provid

Table 2. Keywords of newsgroup topics related to the time series in Figure 2.

to represent one of the four newsgroups as follows. The time series values of the documents coming from each newsgroup are added together, and the newsgroup having the largest of these sums is chosen. Now each document is classified to that topic time series i in which the document projection $\mathbf{w}_i^T\mathbf{Z}(t)$ attains the maximum value. If the document is classified to a time series representing a different newsgroup than where the document was taken from, we consider the document misclassified. The total error is the percentage of misclassifications.

The results are seen in Tables 3 and 4. Complexity pursuit is compared to ordinary ICA (this corresponds to complexity pursuit without the autoregressive modeling of $s(t)$), JADE$_{TD}$ [17], Kolenda's delayed decorrelation [10] and Stone's temporal predictability maximization [19]. Complexity pursuit yields the smallest error of classification. All methods except the temporal predictability maximization consider the data at the current time instant and at some time lag τ; we present here results on $\tau = 1$ and $\tau = 5$. The temporal predictability maximization instead considers short-time and long-time fluctuations in the data simultaneously. Ordinary ICA is not as successful as complexity pursuit, giving evidence that the temporal structure of the data needs to be taken into account. One explanation for the poor performance of the delayed decorrelation and temporal predictability maximization methods might be that they are sensitive to the mean removal of the data; we did not remove the mean as that would have destroyed the sparsity of the data and resulted in more computational load. In all methods except JADE$_{TD}$, the data matrix is first reduced to $K = 100$ dimensions using LSI (SVD) and then $M = 10$ (Table 3) or $M = 6$ (Table 4) topics are estimated. Running JADE$_{TD}$ on a 100-dimensional data matrix was too heavy for Matlab and instead the LSI of order $K = M$ was computed in the beginning. This makes JADE$_{TD}$ computationally less demanding than the other methods, as seen in the rightmost column in Tables 3 and 4 where the number of Matlab's floating point operations is given. The delayed decorrelation method is computationally the heaviest, as the SVD of both the data matrix and the delayed data matrix need to be computed. Actually, the LSI of order less than $K = 100$ would have often been enough; e.g. LSI preprocessing with $K = 30$ would have given an error of 0.1384 on complexity pursuit in the case of $M = 6$ topics.

Method	Error	Flops
Compl. purs. $g = \tanh$, $\tau = 1$	0.103	$2.49 \cdot 10^{10}$
Compl. purs. $g = \tanh$, $\tau = 5$	0.108	$2.31 \cdot 10^{10}$
JADE$_{TD}$ $\tau = 1$	0.177	$7.55 \cdot 10^{8}$
JADE$_{TD}$ $\tau = 5$	0.177	$7.55 \cdot 10^{8}$
ICA $g = \tanh$, $\tau = 1$	0.571	$2.22 \cdot 10^{10}$
ICA $g = \tanh$, $\tau = 5$	0.557	$2.14 \cdot 10^{10}$
Del. decorr. $\tau = 1$	0.652	$4.10 \cdot 10^{10}$
Del. decorr. $\tau = 5$	0.691	$4.11 \cdot 10^{10}$
Temp. pred. maxim.	0.530	$2.11 \cdot 10^{10}$

Table 3. Results of estimating 10 topics on dynamical text document data using complexity pursuit, JADE$_{TD}$ [17], ordinary ICA, delayed decorrelation [10] and temporal predictability maximization [19].

5. CONCLUSIONS

In this paper we have shown experimental results on how independent minimum complexity projections of a dynamic textual data identify some underlying latent or hidden topics in a dynamically evolving text stream. As an example of such dynamically evolving data we used chat line discussions. The method we used for finding the latent topics, complexity pursuit [13], is a generalization of projection pursuit to time series and amounts to estimating projections of the data whose approximative Kolmogoroff complexity is minimized. In our experiments the complexity pursuit algorithm was able to find distinct and meaningful topics of the discussion. We compared the complexity pursuit method to ordinary ICA and to ICA-type methods for time-dependent data: JADE$_{TD}$ [17], delayed decorrelation [10] and temporal predictability maximization [19]. In order to obtain numerical results we used labeled dynamical newsgroup data; complexity pursuit was the most successful in recognizing topically different newsgroup articles. Our results suggest that the method could serve in queries on temporally changing text streams, e.g. complementing other topic segmentation and tracking methods [22].

Method	Error	Flops
Compl. purs. $g = \tanh, \tau = 1$	0.088	$2.37 \cdot 10^{10}$
Compl. purs. $g = \tanh, \tau = 5$	0.156	$2.50 \cdot 10^{10}$
JADE$_{TD}$ $\tau = 1$	0.220	$3.87 \cdot 10^{8}$
JADE$_{TD}$ $\tau = 5$	0.220	$3.87 \cdot 10^{8}$
ICA $g = \tanh, \tau = 1$	0.616	$2.10 \cdot 10^{10}$
ICA $g = \tanh, \tau = 5$	0.628	$2.10 \cdot 10^{10}$
Del. decorr. $\tau = 1$	0.633	$4.28 \cdot 10^{10}$
Del. decorr. $\tau = 5$	0.544	$4.10 \cdot 10^{10}$
Temp. pred. maxim.	0.600	$2.11 \cdot 10^{10}$

Table 4. Results of estimating 6 topics on dynamical text document data using complexity pursuit, JADE$_{TD}$ [17], ordinary ICA, delayed decorrelation [10] and temporal predictability maximization [19].

6. ACKNOWLEDGEMENTS

The author is grateful to Ata Kabán and Prof. Mark Girolami for their advice and help in the early phases of this work. Prof. Mikko Kurimo has given valuable comments on the manuscript. The author is partially supported by Tekniikan edistämissäätiö.

7. REFERENCES

[1] R. A. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*, ACM Press, New York, 1999.

[2] T. Kohonen, S. Kaski, K. Lagus, J. Salojärvi, J. Honkela, V. Paatero, and A. Saarela, "Self organization of a massive document collection," *IEEE Tr. on Neural Networks*, vol. 11, no. 3, pp. 574–585, May 2000, Special Issue on Neural Networks for Data Mining and Knowledge Discovery.

[3] A. Kabán and M. Girolami, "A combined latent class and trait model for the analysis and visualization of discrete data," *IEEE Tr. on Pattern Analysis*, vol. 23, July 2001, In press.

[4] S. Deerwester, S.T. Dumais, G.W. Furnas, and T.K. Landauer, "Indexing by latent semantic analysis," *Journal of the Am. Soc. for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[5] T. Hofmann, "Probabilistic latent semantic analysis," in *Proc. 15th Annual Conf. on Uncertainty in Artificial Intelligence (UAI'99)*, Stockholm, Sweden, 1999.

[6] C.H. Papadimitriou, P. Raghavan, H. Tamaki, and S. Vempala, "Latent semantic indexing: a probabilistic analysis," in *Proc. 17th ACM Symp. Principles of Database Systems*, Seattle, 1998, pp. 159–168.

[7] C. L. Isbell and P. Viola, "Restucturing sparse high dimensional data for effective retrieval," in *Advances in Neural Information Processing Systems 11*, 1998, pp. 480–486.

[8] T. Kolenda, L. K. Hansen, and S. Sigurdsson, "Independent components in text," in *Advances in Independent Component Analysis*, Mark Girolami, Ed., chapter 13, pp. 235–256. Springer-Verlag, 2000.

[9] A. Kabán and M. Girolami, "Unsupervised topic separation and keyword identification in document collections: a projection approach," Tech. Rep. 10, Dept. of Computing and Information Systems, Univ. of Paisley, August 2000.

[10] T. Kolenda and L. K. Hansen, "Dynamical components of chat," Tech. Rep., Technical University of Denmark, 2000.

[11] L. Molgedey and H. G. Schuster, "Separation of a mixture of independent signals using time delayed correlations," *Physical Review Letters*, vol. 72, no. 23, pp. 3634–3637, June 1994.

[12] A. Kabán and M. Girolami, "A dynamic probabilistic model to visualize topic evolution in text streams," *Journal of Intelligent Information Systems, Special Issue on Automated Text Categorization*, Conditionally accepted.

[13] A. Hyvärinen, "Complexity pursuit: separating interesting components from time-series," *Neural Computation*, vol. 13, no. 4, pp. 883–898, April 2001.

[14] J. H. Friedman and J. W. Tukey, "A projection pursuit algorithm for exploratory data analysis," *IEEE Tr. of Computers*, vol. c-23, no. 9, pp. 881–890, 1974.

[15] G. Salton and M.J. McGill, *Introduction to modern information retrieval*, McGraw-Hill, New York, 1983.

[16] A. Belouchrani, K. A. Meraim, J.-F. Cardoso, and E. Moulines, "A blind source separation technique based on second order statistics," *IEEE Tr. on Signal Processing*, vol. 45, no. 2, pp. 434–444, February 1997.

[17] K.-R. Müller, P. Philips, and A. Ziehe, "JADE$_{TD}$: Combining higher-order statistics and temporal information for blind source separation (with noise)," in *Proc. Int. Workshop on Independent Component Analysis and Signal Separation (ICA'99)*, Aussois, France, 1999, pp. 87–92.

[18] P. Pajunen, "Blind source separation using algorithmic information theory," *Neurocomputing*, vol. 22, pp. 35–48, 1998.

[19] J. V. Stone, "Blind source separation using temporal predictability," *Neural Computation*, 2001.

[20] A. Hyvärinen, "Fast and robust fixed-point algorithms for independent component analysis," *IEEE Tr. on Neural Networks*, vol. 10, no. 3, pp. 626–634, May 1999.

[21] E. Bingham, A. Kabán, and M. Girolami, "Finding topics in dynamical text: application to chat line discussions," in *10th Int. World Wide Web Conf. Poster Proc.*, 2001, pp. 198–199.

[22] J. Allan, J. Carbonell, G. Doddington, J. Yamron, and Y. Yang, "Topic detection and tracking pilot study. final report," in *Proc. of DARPA Broadcast News Transcription and Understanding Workshop*, 1998, pp. 194–218.