3b2 Version Number : 6.06e/W (Aug 31 2000)
File path : p:/Santype/Journals/Kluwer/Nepl/765/NEPL765.3d
Date and Time : 25/10/02 and 17:10

Neural Processing Letters **17:** 1–15, 2002.

Topic Identification in Dynamical Text by Complexity Pursuit

ELLA BINGHAM[★], ATA KABÁN[†] and MARK GIROLAMI[‡]
Neural Networks Research Centre, Helsinki University of Technology, P.O. Box 5400, FIN-02015 HUT, Finland

Abstract. The problem of analysing dynamically evolving textual data has arisen within the last few years. An example of such data is the discussion appearing in Internet chat lines. In this Letter a recently introduced source separation method, termed as *complexity pursuit*, is applied to the problem of finding topics in dynamical text and is compared against several blind separation algorithms for the problem considered. Complexity pursuit is a generalisation of projection pursuit to time series and it is able to use both higher-order statistical measures and temporal dependency information in separating the topics. 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.

Key words. chat line discussion, complexity pursuit, dynamical text, independent component analysis, time series

Abbreviations. ICA – Independent component analysis; LSI – Latent semantic indexing

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 [2]. The WEBSOM[1] is a document clustering and visualisation method [19]; its probabilistic counterpart has been presented e.g. in [16]. Another basic algorithm is Latent Semantic Indexing (LSI) [7] 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 [9] and Papadimitriou [27]. LSI is found to capture some of the underlying semantics of textual data, resolving problems of synonymy and polysemy.

In recent years, the use of higher-order statistics and information-theoretic measures has gained popularity in the data analysis community. LSI uses only

★ Corresponding author. e-mail: ella@iki.fi
† Current address of correspondance: Department of Information Systems, Eötvös Loránd University, Pázmány Péter sétány 1/C, Budapest, Hungary H-1117. e-mail: kaba-ci0@paisley.ac.uk
‡ Current address of correspondance: School of Information and Communications Technology, University of Paisley, Paisley PA1 2BE, Scotland, UK. e-mail: giro-ci0@paisley.ac.uk
[1] See http://websom.hut.fi/websom/

second-order moments of the data and neglects any higher order correlations, so a natural step forward is to apply more powerful methods. An important class of higher order statistical methods are independent component analysis (ICA)-type methods [6, 12, 14]. In ICA a set of multidimensional observations is presented as a (linear) combination of some underlying latent features that are more or less independent of each other.

First approaches of using ICA in the context of text data were presented by Isbell and Viola [13], Kolenda et al. [22] and Kabán and Girolami [15]. 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 [20, 21] employ Molgedey and Schuster's [23] 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 [17] have recently presented a Hidden Markov Model (HMM)-type algorithm for the topographic visualization of time-varying data.

In this Letter 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 [11], is a generalization of projection pursuit [8] to time series and it is able to exploit both higher-order 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, on the other hand, neglects any time-dependency information and defines interestingness as non-gaussianity. 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 the data and its preprocessing. 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, and shows comparisons between several algorithms that have been presented for separating time-correlated signals. Finally, some conclusions are drawn in Section 5.

2. Dynamical Textual Data: Chat Line Discussion

Often the characteristics of the textual data of interest change over time. Such dynamical data can be found e.g. in the online news services. Our example of dynamically evolving text is chat line data, and later also newsgroup data that shares some similarities to 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. (In splitting the text stream, the boundaries between comment lines are not taken into account, as this might result into windows of different lengths. Also, this kind of partitioning is not always possible in other dynamical text streams, and we do not wish to restrict our analysis to chat line discussions only.)

We employ the vector space model [28] 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 $\mathbf{X} = (\mathbf{x}(t))$, $t = 1, \ldots, N$.

As a preprocessing step we compute the LSI of the data matrix $\mathbf{X}$, that is, the singular value decomposition (SVD)

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

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 the K ($K \ll T$) largest singular values in the diagonal matrix $\mathbf{D}_K$:

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

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 complexity pursuit algorithm described in the following section. It is often advantageous to compute the LSI projection onto a somewhat larger dimensionality $K > M$ and then to find M minimum complexity projections.

To represent the estimated topics in the term space, the transpose of the original data is first projected onto the LSI term space by

$$\mathbf{Z}_{\text{term}} = \mathbf{D}_K^{-1}\mathbf{V}_K\mathbf{X}_K^T = \mathbf{U}_K^T \tag{3}$$

and then projected onto the directions $\mathbf{W}$ that were found earlier by feeding $\mathbf{Z}$ into the algorithm.

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 easily updated by folding-in [4] documents or terms: the LSI projection of a new document vector $\mathbf{x}_{\text{new}}$ (a new column in $\mathbf{X}$) is $\mathbf{z}_{\text{new}} = \mathbf{x}_{\text{new}}\mathbf{U}_K\mathbf{D}_K^{-1}$. Similarly, the projection of a new term vector $\mathbf{x}_{\text{new}}^{\text{term}}$ (a new row in $\mathbf{X}$) is $\mathbf{z}_{\text{new}}^{\text{term}} = \mathbf{x}_{\text{new}}^{\text{term}}\mathbf{V}_K\mathbf{D}_K^{-1}$.

3. The Complexity Pursuit Algorithm

Complexity pursuit [11] is a recently developed, computationally simple algorithm for separating interesting components from time series. It is an extension of projection pursuit [8] 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. [3, 23]); however the early algorithms as that proposed in [23] 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 [24]. However, complexity pursuit combines these criteria in a principled way by employing the information theoretical concept of Kolmogoroff complexity [25] 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 [30]; 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 [30] and [20]. Some other methods for detecting the semantics in a dynamical text stream are described e.g. in [29].

Our data model assumes that the observations $\mathbf{x}(t)$ are linear mixtures of some latent components:

$$\mathbf{x} = \mathbf{A}\mathbf{s} \tag{4}$$

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. In the context of complexity pursuit we do not put any special emphasis on the statistical independence of s_i, even though the data model (4) is similar to that of linear ICA.

A separate autoregressive model is assumed to model each component $s_i = \mathbf{w}_i^T\mathbf{x}$; as a simple special case of the algorithm presented in [11], we employ a first order autoregressive (AR) process $\hat{s}_i(t) = \alpha_i s_i(t - \tau)$ on each latent variable s_i. The approximate Kolmogoroff complexity of the residuals $\delta s(t) = s(t) - \hat{s}(t)$ (using the predictive coding of the components) [11]

$$\hat{K}(\delta(\mathbf{w}^T\mathbf{x}(t))) = E\Big\{G\Big(\frac{1}{\sigma_\delta(\mathbf{w})}\mathbf{w}^T(\mathbf{x}(t) - \alpha\mathbf{x}(t - \tau))\Big)\Big\} + \log\sigma_\delta(\mathbf{w}) \tag{5}$$

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 (5) 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 [11].

In our application the latent time-components s_i will model the evolving topics of the discussion. To find the minima of (5), 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 inverse of the mixing matrix for whitened data. 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 [11]

$$\hat{\alpha} = \mathbf{w}^T E\{\mathbf{z}(t)\mathbf{z}(t - \tau)\}\mathbf{w} \tag{6}$$

Then the gradient update of $\mathbf{w}$ that minimizes (5) is the following [11]:

$$\mathbf{w} \leftarrow \mathbf{w} - \mu E\{(\mathbf{z}(t) - \alpha(\mathbf{w})\mathbf{z}(t - \tau))g(\mathbf{w}^T(\mathbf{z}(t) - \alpha(\mathbf{w})\mathbf{z}(t - \tau)))\} \tag{7}$$

$$\mathbf{w} \leftarrow \mathbf{w}/||\mathbf{w}|| \tag{8}$$

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 (5). In practice, the choice of g is quite flexible. Choosing a linear g corresponds to neglecting the higher-order structure of the data, and

analyzing the time-correlations of the signals only. This kind of complexity minimization is discussed e.g. in [26]. In general, a nonlinear g should be preferred for the estimation of nongaussian latent variables or residuals.

To estimate several projections one can either use a deflation scheme, or estimate all projections simultaneously in a symmetric manner and use orthogonal decorrelation

$$\mathbf{W} \leftarrow \sqrt{(\mathbf{W}\mathbf{W}^T)^{-1}}\mathbf{W} \tag{9}$$

instead of (8). 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 [10].

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

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 hr 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); subsequent windows shared an overlap of 66%. From these windows a term histogram was generated using the Bow toolkit[3]. Stemming, stop-word removal and tf-idf (term frequency – inverse document frequency) term weighting were 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 – ith entry of a document vector was 1 if the ith 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

[2]http://www.cnn.com/chat/channel/cnn_newsroom
[3]http://www.cs.cmu.edu/~mccallum/bow/

algorithms – that would have destroyed the sparsity and resulted in severe computational difficulties in the LSI preprocessing stage.

The choice of the number of estimated topics M is somewhat arbitrary[4]. It has been proved in [27] 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. In the case of static data – e.g. ICA of functional magnetic resonance imaging (fMRI) and image recognition, or textual document analysis [15] – one can use both $\mathbf{X}$ and $\mathbf{X}^T$ for training (see [15] for derivation); this is called spatio-temporal ICA. In our case, the documents evolve dynamically but the terms have no time structure, and thus they will be employed in the visualization phase only.

It should also be noted that the projections $\mathbf{w}^T\mathbf{z}(t)$ that represent the latent topics of discussion are found by the complexity pursuit algorithm up to permutation, sign 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. (Katz [18] studies the distribution of words in phrases in more detail.) We must change the sign of the negatively skewed projections $\mathbf{w}^T\mathbf{z}(t)$ 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. AR models of order >1 did not bring substantial improvement in the results; also, estimating an AR(1) model is computationally much simpler than more complex AR models.

The choice of the nonlinearity g in Formula (7) is another issue. The best results were obtained when g was chosen as $g(u) = \tanh(u)$, corresponding to imposing a 'cosh' prior on the residuals $s(t) - \alpha s(t - \tau)$. We have also previously [5] had good results with the simple $g(u) = \text{sign}(u)$ nonlinearity that corresponds to a Laplace prior on the residuals. In the ICA of static text documents, a nonlinearity $g(u) = u^2$ has been found successful in e.g. [15], corresponding to the skewed distribution of terms in documents. For dynamical text data, $g(u) = \tanh(u)$ was

[4]In a recent paper, Kolenda et al. [21] give a Bayesian method for choosing the number of estimated topics. We became aware of their work during the review process of this paper.

nevertheless better. Also, choosing a linear g (which neglects the non-gaussian, higher-order structure of the data) did not prove successful in our experiments.

4.2. RESULTS ON CHAT LINE DATA

The LSI of order $K = 100$ was computed as a preprocessing step as described in (2). Smaller K would also suffice, as we will demonstrate on another data set in the next section. 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. The time span of Figure 1 is almost 24 hr; some topics are more or less persistent during the whole period and some will come up again after a few hours. The same fact can also be seen in the original text stream.

We now turn to analyze the projections $\mathbf{w}_i^T\mathbf{Z}_{\text{term}}$ of the terms onto minimum complexity directions. This information is complementary 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 $\mathbf{w}_i^T\mathbf{Z}_{\text{term}}$ we get a list of keywords for the i-th topic. The keywords are listed in Table I in the order of decreasing importance. 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

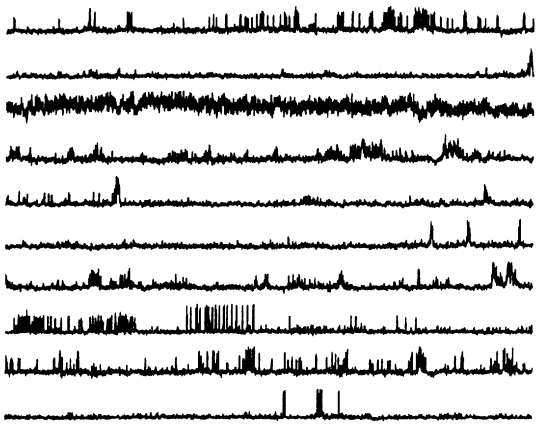

Figure 1. Activity of topics (vertical axis) in each chat window (horizontal axis). $g(u) = \tanh(u)$ and $\tau = 5$ were used in Formula (7). The uppermost time series corresponds to topic 1, the second to topic 2 etc.

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

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

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 is a short discussion dealing with the values of the politicians in the US.

One can compare the activities of the topic time series in Figure 1, and the term by document matrix frequencies of the first few keywords of each topic; the frequencies of the keywords nicely follow the activities of the time series.

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 I.

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.

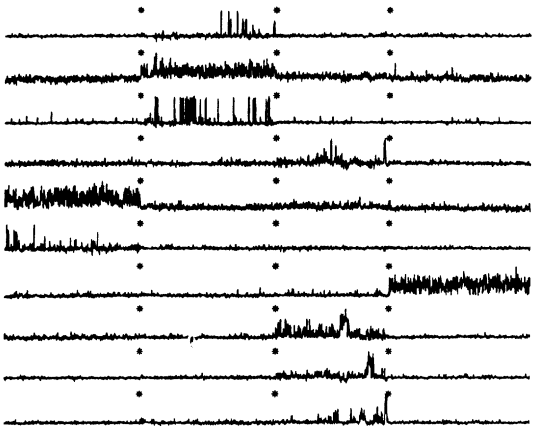

Figure 2. Activity of topics (vertical axis) in each newsgroup window (horizontal axis). $g(u) = \tanh(u)$ and $\tau = 5$ were used in Formula (7). The asterisks denote the newsgroup borders: sci.crypt, sci.med, sci.space and soc.religion.christian. The uppermost time series corresponds to topic 1, the second to topic 2 etc.

4.3. RESULTS ON NEWSGROUP DATA

In this section we present experimental results on newsgroup data where consecutive 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 representation of the documents was chosen 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.

LSI (2) of order $K = 50$ was computed as a preprocessing step. 6, 8 or 10 minimum-complexity directions $\mathbf{w}$ were estimated – discussion in a newsgroup can well be divided into subgroups, if more than one topic is dealt with. Figure 2 shows the topic time series $\mathbf{w}^T\boldsymbol{Z}$ in the case of 10 estimated topics. The asterisks in Figure 2 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 II, and they also nicely correspond to newsgroup labels: topics 1, 2 and 3 characterize different aspects discussed in sci.med, topics 4, 8, 9 and 10 in sci.space, topics 5 and 6 in sci.crypt and topic 7 is the only topic from soc.religion.christian.

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

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

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
human	problem	bank	design	kei
effect	diseas	skeptic	power	chip
food	scienc	intellect	station	govern
studi	medic	chastiti	control	encrypt
brain	studi	surrend	shuttl	secur
glutam	result	shame	orbit	clipper
review	food	won	option	public
level	effect	patient	human	system
singl	treatment	mar	provid	algorithm
paper	lot	medic	flight	david
diet	test	blood	engin	bit
industri	doctor	pittsburgh	modul	phone
blood	patient	comput	capabl	data
real	experi	practic	addition	nsa
high	medicin	migrain	system	escrow
Topic 6	Topic 7	Topic 8	Topic 9	Topic 10
phone	god	space	earth	matter
drug	christian	launch	venu	burst
commun	church	satellit	soviet	rememb
kei	christ	market	planet	star
life	sin	project	probe	black
dealer	jesu	commerci	mission	galaxi
assum	bibl	servic	surfac	red
crimin	approv	plan	mile	grb
discov	scriptur	orbit	kilomet	dark
hold	lord	cost	atmospher	gamma
motiv	homosexu	vehicl	venera	galact
terrorist	arami	note	lander	shift
compromis	faith	develop	orbit	object
system	love	fund	craft	show
polic	paul	nasa	balloon	energi

The classification error of the newsgroup documents is computed in the following way: The topic time series $\mathbf{w}_i^T \boldsymbol{Z}$ are first normalized to unit variance. Then a time series is mapped to the newsgroup whose documents have the highest sum of time series values in this particular time series. This is done at each time series separately. Now on the other hand, each document t is classified to that topic time series i in which the document projection $\mathbf{w}_i^T \boldsymbol{Z}(t)$ 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 Table III which shows average results over 20 trials with different initial values for $\mathbf{w}$. Complexity pursuit is compared to ordinary ICA (FastICA [10]; this corresponds to complexity pursuit without the autoregressive

Table III. Results of estimating 10, 8 or 6 topics on dynamical text document data (newsgroup data) using complexity pursuit (with $g = \tanh$), JADE$_{TD}$ [24], ordinary FastICA (with $g = \tanh$), delayed decorrelation [20] and temporal predictability maximization [30]. Average results over 20 independent trials with different initial values for $\mathbf{w}$.

Method	Error $M = 10$	Flops $\cdot 10^9$	Error $M = 8$	Flops $\cdot 10^9$	Error $M = 6$	Flops $\cdot 10^9$
Compl. purs. $\tau = 1$	0.1515	9.29	0.1230	8.48	0.1081	8.01
Compl. purs. $\tau = 5$	0.1495	8.33	0.1423	7.82	0.1922	7.57
Compl. purs. $\tau = 10$	0.1737	8.27	0.1933	8.05	0.2760	7.53
JADE$_{TD}$ $\tau = 1$	0.1774	0.69	0.2043	0.55	0.2204	0.37
JADE$_{TD}$ $\tau = 5$	0.1774	0.79	0.2043	0.55	0.2204	0.37
JADE$_{TD}$ $\tau = 10$	0.1774	0.69	0.2043	0.55	0.2204	0.39
FastICA	0.4905	7.40	0.5460	7.16	0.6083	6.92
Del. decorr. $\tau = 1$	0.6591	1.38	0.6603	1.08	0.6920	0.77
Del. decorr. $\tau = 5$	0.6356	1.40	0.6700	1.08	0.6709	0.78
Del. decorr. $\tau = 10$	0.6688	1.38	0.6675	1.10	0.6852	0.77
Temp. pred. maxim.	0.4843	6.82	0.5442	6.82	0.6116	6.81

modeling of $s(t)$), JADE$_{TD}$ [24], Kolenda's delayed decorrelation [20] and Stone's temporal predictability maximization [30]. All these methods except ordinary ICA and the temporal predictability maximization consider the data at the current time instant and at some time lag τ; we present here results on $\tau = 1$, 5 and 10. The temporal predictability maximization instead considers short-time and long-time fluctuations in the data simultaneously.

As seen in Table III, complexity pursuit yields the smallest error of classification. Ordinary ICA, delayed decorrelation and temporal predictability maximization are not as successful as complexity pursuit and JADE$_{TD}$, giving evidence that both the temporal structure and information-theoretic measures of the data need to be taken into account. In all methods except JADE$_{TD}$ and delayed decorrelation, the data matrix is first reduced to $K = 50$ dimensions using LSI (SVD) and then $M = 10$, 8 or 6 topics are estimated. In the cases of JADE$_{TD}$ and delayed decorrelation, the LSI of order $K = M$ was computed in the beginning. This makes these two methods computationally less demanding than the other methods, as seen in Table III where the number of Matlab's floating point operations is given.

A new paper by Kolenda et al. [21] gives a method for determining the optimal lag parameter τ; this method is not applied here. The values for τ found in [21] are somewhat larger (naturally, this is data dependent) than those used in Table III, but testing e.g. values of $\tau = 20$, 50 or 100 in the delayed decorrelation method did not give any improvements on the results.

Figure 3 is an example of a box plot of the results, showing the variation in the results between different runs of the algorithms. All methods except JADE$_{TD}$ are sensitive to the initial choice of the vectors $\mathbf{w}$.

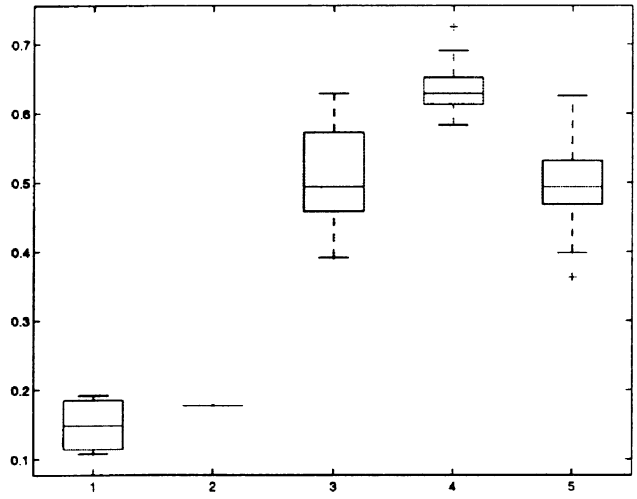

Figure 3. Box plot of the error in the case of $M = 10$ estimated topics and lag parameter $\tau = 5$. Methods from left to right: complexity pursuit, JADE$_{TD}$, ordinary FastICA, delayed decorrelation and temporal predictability maximization.

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 [11], 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$_{TD}$ [24], delayed decorrelation [20] and temporal predictability maximization [30]. 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 [1].

Acknowledgements

The authors are grateful to Thomas Kolenda for sharing his comments on the problem, and to Prof. Mikko Kurimo and the anonymous reviewers for giving valuable comments on the manuscript. E. Bingham has been partly supported by Ella and Georg Ehrnrooth Foundation and A. Kabán and M. Girolami have been partly supported by the Finnish National Technology Agency TEKES.

References

1. Allan, J., Carbonell, J., Doddington, G., Yamron, J. and Yang, Y.: Topic detection and tracking pilot study. Final report, In: *Proc. of DARPA Broadcast News Transcription and Understanding Workshop*, 1998, pp. 194–218.
2. Baeza-Yates, R. A. and Ribeiro-Neto, B.: *Modern Information Retrieval*, New York: ACM Press, 1999.
3. Belouchrani, A., Meraim, K. A., Cardoso, J.-F. and Moulines, E.: A blind source separation technique based on second order statistics, *IEEE Tr. on Signal Processing*, **45**(2) (1997), 434–444.
4. Berry, M. W., Dumais, S. T. and Letsche, T. A.: Computational methods for intelligent information access, In: *Proc. of Supercomputing '95*, San Diego, CA: USA, 1995.
5. Bingham, E., Kabán, A. and Girolami, M.: Finding topics in dynamical text: application to chat line discussions, In: *10th Int. World Wide Web Conf. Poster Proc.*, 2001, pp. 198–199.
6. Comon, P.: Independent component analysis—a new concept? *Signal Processing*, **36** (1994), 287–314.
7. Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K. and Harshman, R.: Indexing by latent semantic analysis, *Journal of the American Society for Information Science*, **41**(6) (1990), 391–407.
8. Friedman, J. H. and Tukey, J. W.: A projection pursuit algorithm for exploratory data analysis, *IEEE Tr. of Computers*, **c-23**(9) (1974), 881–890.
9. Hofmann, T.: Probabilistic Latent Semantic Analysis, In: *Proc. 15th Annual Conf. on Uncertainty in Artificial Intelligence (UAI'99)*, Sweden: Stockholm, 1999.
10. Hyvärinen, A.: Fast and robust fixed-point algorithms for independent component analysis, *IEEE Tr. on Neural Networks*, **10**(3) (1999), 626–634.
11. Hyvärinen, A.: Complexity pursuit: separating interesting components from time-series, *Neural Computation*, **13**(4) (2001), 883–898.
12. Hyvärinen, A., Karhunen, J. and Oja, E.: *Independent component analysis*, Wiley Interscience, 2001.
13. Isbell, C. L. and Viola, P.: Restucturing sparse high dimensional data for effective retrieval, In: *Advances in Neural Information Processing Systems 11*, 1998, pp. 480–486.
14. Jutten, C. and Herault, J.: Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture, *Signal Processing*, **24** (1991), 1–10.
15. Kabán, A. and Girolami, M.: Unsupervised topic separation and keyword identification in document collections: a projection approach, Technical Report 10, Dept. of Computing and Information Systems, Univ. of Paisley, 2000.
16. Kabán, A. and Girolami, M.: A combined latent class and trait model for the analysis and visualization of discrete data, *IEEE Tr. on Pattern Analysis*, **23**(8) (2001), 859–872.
17. Kabán, A. and Girolami, M.: A dynamic probabilistic model to visualize topic evolution in text streams, *Journal of Intelligent Information Systems, Special Issue on Automated Text Categorization*, **18**(2) (2002).
18. Katz, S.: Distribution of content words and phrases in text and language modeling, *Natural Language Engineering*, **2**(1) (1996), 15–59.
19. Kohonen, T., Kaski, S., Lagus, K., Salojärvi, J., Honkela, J., Paatero, V. and Saarela, A.: Self organization of a massive document collection, *IEEE Tr. on Neural Networks*, **11**(3) (2000) 574–585. Special Issue on Neural Networks for Data Mining and Knowledge Discovery.
20. Kolenda, T. and Hansen, L. K.: *Dynamical components of chat*, Technical report Technical University of Denmark, 2000.

21. Kolenda, T., Hansen, L. K. and Larsen, J.: Signal detection using ICA: application to chat room topic spotting, In: Lee and Jung and Makeig and Sejnowski (eds.): *Proc. of the Third International Conference on Independent Component Analysis and Signal Separation (ICA2001)*, San Diego, CA: USA pp. 540–545, 2001.
22. Kolenda, T., Hansen, L. K. and Sigurdsson, S.: Independent components in text, In: M. Girolami (ed.): *Advances in Independent Component Analysis*, Springer-Verlag, 2000, Chapt. 13, pp. 235–256.
23. Molgedey, L. and Schuster, H. G.: Separation of a mixture of independent signals using time delayed correlations, *Phys. Review Letters,* **72**(23) (1994), 3634–3637.
24. Müller, K.-R., Philips, P. and Ziehe, A.: 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)*, France: Aussois, 1999, pp. 87–92.
25. Pajunen, P.: Blind source separation using algorithmic information theory, *Neurocomputing*, **22** (1998), 35–48.
26. Pajunen, P.: Blind source separation of natural signals based on approximate complexity minimization, In: *Proc. Int. Workshop on Independent Component Analysis and Signal Separation (ICA'99)*, France: Aussois, 1999, pp. 267–270.
27. Papadimitriou, C., Raghavan, P., Tamaki, H. and Vempala, S.: Latent semantic indexing: a probabilistic analysis, In: *Proc. 17th ACM Symp. Principles of Database Systems*, Seattle, 1998, pp. 159–168.
28. Salton, G. and McGill, M.J.: *Introduction to modern information retrieval*, New York: McGraw-Hill, 1983.
29. Slaney, M. and Ponceleon, D.: Hierarchical segmentation: finding changes in a text signal, In: *Proc. of the SIAM Text Mining 2001 Workshop*, Chicago, IL: 2001, pp. 6–13.
30. Stone, J. V.: Blind source separation using temporal predictability, *Neural Computation*, **13**(4) (2001).