Local Space-Time Smoothing for Version Controlled Documents

Seungyeon Kim
Georgia Institute of Technology

Guy Lebanon
Georgia Institute of Technology

Abstract

Unlike static documents, version controlled documents are continuously edited by one or more authors. Such collaborative revision process makes traditional modeling and visualization techniques inappropriate. In this paper we propose a new representation based on local space-time smoothing that captures important revision patterns. We demonstrate the applicability of our framework using experiments on synthetic and real-world data.

1 Introduction

Most computational linguistics studies concentrate on modeling or analyzing documents as sequences of words. In this paper we consider modeling and visualizing version controlled documents which is the authoring process leading to the final word sequence. In particular, we focus on documents whose authoring process naturally segments into consecutive versions. The revisions, as the differences between consecutive versions are often called, may be authored by a single author or by multiple authors working collaboratively.

One popular way to keep track of version controlled documents is using a version control system such as CVS or Subversion (SVN). This is often the case with books or with large computer code projects. In other cases, more specialized computational infrastructure may be available, as is the case with the authoring API of Wikipedia.org, Slashdot.com, and Google Wave. Accessing such API provides information about what each revision contains, when was it submitted, and who edited it. In any case, we formally consider a version controlled document as a sequence of documents $d_1, \ldots, d_l$ indexed by their revision number where d_i typically contains some locally concentrated additions or deletions, as compared to d_{i-1}.

In this paper we develop a continuous representation of version controlled documents that generalizes the locally weighted bag of words representation (Lebanon et al., 2007). The representation smooths the sequence of version controlled documents across two axes-time t and space s. The time axis t represents the revision and the space axis s represents document position. The smoothing results in a continuous map from a space-time domain to the simplex of term frequency vectors

$$\gamma : \Omega \to \mathbb{P}_V \quad \text{where} \quad \Omega \subset \mathbb{R}^2, \quad \text{and} \qquad (1)$$
$$\mathbb{P}_V = \left\{ w \in \mathbb{R}^{|V|} : w_i \geq 0, \ \sum_{i=1}^{|V|} w_i = 1 \right\}.$$

The mapping above (V is the vocabulary) captures the variation in the local distribution of word content across time and space. Thus $[\gamma(s,t)]_w$ is the (smoothed) probability of observing word w in space s (document position) and time t (version). Geometrically, γ realizes a divergence-free vector field (since $\sum_w [\gamma(s,t)]_w = 1$, γ has zero divergence) over the space-time domain Ω.

We consider the following four version controlled document analysis tasks. The first task is visualizing word-content changes with respect to space (how quickly the document changes its content), time (how much does the current version differs from the previous one), or mixed space-time. The second task is detecting sharp transitions or edges in word content. The third task is concerned with segmenting the space-time domain into a finite partition reflecting word content. The fourth task is predicting future revisions. Our main tool in addressing tasks 1-4 above is to analyze the values of the vector field γ and its first order derivatives fields

$$\nabla\gamma = (\dot{\gamma}_s, \dot{\gamma}_t)\,. \tag{2}$$

2 Space-Time Smoothing for Version Controlled Documents

With no loss of generality we identify the vocabulary V with positive integers $\{1,\ldots,V\}$ and represent a word $w \in V$ by a unit vector[1] (all zero except for 1 at the w-component)

$$e(w) = (0,\ldots,0,1,0,\ldots,0)^{\top} \quad w \in V. \tag{3}$$

We extend this definition to word sequences thus representing documents $\langle w_1,\ldots,w_N\rangle$ ($w_i \in V$) as sequences of V-dimensional vectors $\langle e(w_1),\ldots,e(w_N)\rangle$. Similarly, a version controlled document is sequence of documents $d^{(1)},\ldots,d^{(l)}$ of potentially different lengths $d^{(j)} = \langle w_1^{(j)},\ldots,w_{N(j)}^{(j)}\rangle$. Using (3) we represent a version controlled document as the array

$$\begin{array}{ccc} e(w_1^{(1)}), & \ldots, & e(w_{N(1)}^{(1)}) \\ \vdots & \ddots & \vdots \\ e(w_1^{(l)}), & \ldots, & e(w_{N(l)}^{(l)}) \end{array} \tag{4}$$

where columns and rows correspond to space (document position) and time (versions).

The array (4) of high dimensional vectors represents the version controlled document without any loss of information. Nevertheless the high dimensionality of V suggests we smooth the vectors in (4) with neighboring vectors in order to better capture the local word content. Specifically we convolve each component of (4) with a 2-D smoothing kernel K_h to obtain a smooth vector field γ over space-time (Wand and Jones, 1995) e.g.,

$$\gamma(s,t) = \sum_{s'}\sum_{t'} K_h(s-s',t-t')e(w_{s'}^{(t')})$$
$$K_h(x,y) \propto \exp\left(-(x^2+y^2)/(2h^2)\right). \tag{5}$$

Thus as (s,t) vary over a continuous domain $\Omega \subset \mathbb{R}^2$, $\gamma(s,t)$, which is a weighted combination of neighboring unit vectors, traces a continuous surface in $\mathbb{P}_V \subset \mathbb{R}^V$. Assuming that the kernel K_h is a normalized density it can be shown that $\gamma(s,t)$ is a non-negative normalized vector i.e., $\gamma(s,t) \in \mathbb{P}_V$ (see (1) for a definition of $\mathbb{P}_V$) measuring the local distribution of words around the space-time location (s,t). It thus extends the concept of lowbow (locally weighted bag of words) introduced in (Lebanon et al., 2007) from single documents to version controlled documents.

One difficulty with the above scheme is that the document versions $d_1,\ldots,d_l$ may be of different lengths. We consider two ways to resolve this issue. The first pads shorter document versions with zero vectors as needed. We refer to the resulting representation γ as the non-normalized representation. The second approach normalizes all document versions to a common length, say $\prod_{j=1}^{l} N(j)$. That is each word in the first document is expanded into $\prod_{j\neq 1} N(j)$ words, each word in the second document is expanded into $\prod_{j\neq 2} N(j)$ words etc. We refer to the resulting representation γ as the normalized representation.

The non-normalized representation has the advantage of conveying absolute lengths. For example, it makes it possible to track how different portions of the document grow or shrink (in terms of number of words) with the version number. The normalized representation has the advantage of conveying lengths relative to the document length. For example, it makes it possible to track how different portions of the document grow or shrink with the version number relative to the total document length. In either case, the space-time domain Ω on which γ is defined (5) is a two dimensional rectangular domain $\Omega = [0,I]\times[0,J]$.

Before proceeding to examine how γ may be used in the four tasks described in Section 1 we demonstrate our framework with a simple low dimensional example. Assuming a vocabulary of two words $V = \{1,2\}$ we can visualize γ by displaying its first component as a grayscale image (since $[\gamma(s,t)]_2 = 1 - [\gamma(s,t)]_1$ the second component is redundant). Specifically, we created a version controlled document with three contiguous segments whose $\{1,2\}$ words were sampled from Bernoulli distributions with parameters 0.3 (first segment), 0.7 (second segment), and 0.5 (third segment). That is, the probability of getting 1 is highest for the second segment, equal for the third and lowest for the first segment. The initial lengths of the segments were

[1] Note the slight abuse of notation as V represents both a set of words and an integer $V = \{1,\ldots,V\}$ with $V = |V|$.

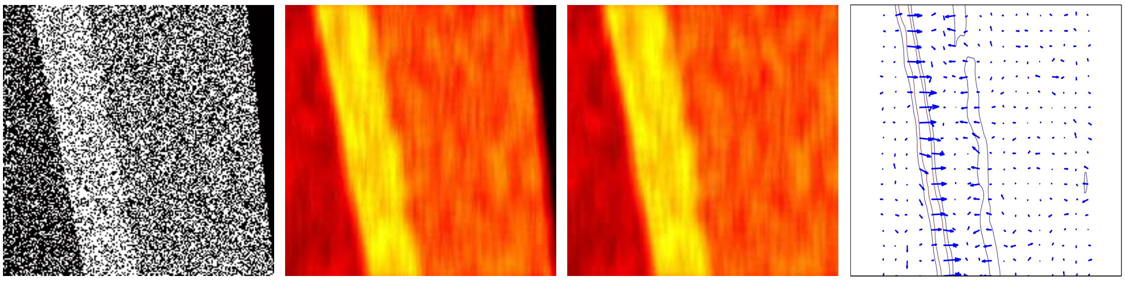

Figure 1: Four space-time representations of a simple synthetic version controlled document over $V = \{1, 2\}$ (see text for more details). The left panel displays the first component of (4) (non-smoothed array of unit vectors corresponding to words). The second and third panels display $[\gamma(s,t)]_1$ for the non-normalized and normalized representations respectively. The fourth panel displays the gradient vector field $(\dot\gamma_s(s,t), \dot\gamma_t(s,t))$ (contour levels represent the gradient magnitude). The black portions of the first two panels correspond to zero padding due to unequal lengths of the different versions.

30, 40 and 120 words with the first segment increasing and the third segment decreasing at half the rate of the first segment with each revision. The length of the second segment was constant across the different versions. Figure 1 displays the nonsmoothed ragged array (4) (left), the non-normalized $[\gamma(s,t)]_1$ (middle left) and the normalized $[\gamma(s,t)]_1$ (middle right).

While the left panel doesn't distinguish much between the second and third segment the two smoothed representations display a nice segmentation of the space-time domain into three segments, each with roughly uniform values. The non-normalized representation (middle left) makes it easy to see that the total length of the version controlled document is increasing but it is not easy to judge what happens to the relative sizes of the three segments. The normalized representation (middle right) makes it easy to see that the first segment increases in size, the second is constant, and the third decreases in size. It is also possible to notice that the growth rate of the first segment is higher than the decay rate of the third.

3 Visualizing Change in Space-Time

We apply the space-time representation to four tasks. The first task, visualizing change, is described in this section. The remaining three tasks are described in the next three section.

The space-time domain Ω represents the union of all document versions and all document positions. Some parts of Ω are more homogeneous and some are less in terms of their local word distribution. Locations in Ω where the local word distribution substantially diverges from its neighbors correspond to sharp content transitions. On the other hand, locations whose word distribution is more or less constant correspond to slow content variation.

We distinguish between three different types of changes. The first occurs when the word content changes substantially between neighboring document positions within a certain document version. As an example consider a document location whose content shifts from high level introductory motivation to a detailed technical description. Such change is represented by

$$\|\dot\gamma_s(s,t)\|^2 = \sum_{w=1}^{V} \left(\frac{\partial[\gamma(s,t)]_w}{\partial s}\right)^2. \qquad (6)$$

A second type of change occurs when a certain document position undergoes substantial change in local word distribution across neighboring versions. An example is erroneous content in one version being heavily revised in the next version. Such change along the time axis corresponds to the magnitude of

$$\|\dot\gamma_t(s,t)\|^2 = \sum_{w=1}^{V} \left(\frac{\partial[\gamma(s,t)]_w}{\partial t}\right)^2. \qquad (7)$$

Expression (6) may be used to measure the instantaneous rate of change in the local word distribution. Alternatively, integrating (6) provides a global measure of change

$$h(s) = \int \|\dot\gamma_s(s,t)\|^2\,dt, \quad g(t) = \int \|\dot\gamma_t(s,t)\|^2\,ds$$

with $h(s)$ describing the total amount of spatial change across all revisions and $g(t)$ describing

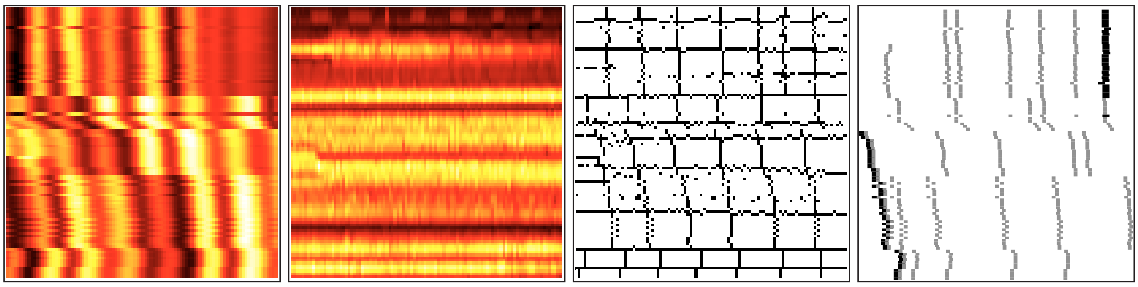

Figure 2: Gradient and edges for a portion of the version controlled Wikipedia Religion article. The left panel displays $\|\dot\gamma_s(s,t)\|^2$ (amount of change across document locations for different versions). The second panel displays $\|\dot\gamma_t(s,t)\|^2$ (amount of change across versions for different document positions). The third panel displays the local maxima of $\|\dot\gamma_s(s,t)\|^2 + \|\dot\gamma_t(s,t)\|^2$ which correspond to potential edges, either vertical lines (section and subsection boundaries) or horizontal lines (between substantial revisions). The fourth panel displays boundaries of sections and subsections as black and gray lines respectively.

the total amount of version change across different document positions. $h(s)$ may be used to detect document regions undergoing repeated substantial content revisions and $g(t)$ may be used to detect revisions in which substantial content has been modified across the entire document.

We conclude with the integrated directional derivative

$$\int_0^1 \|\dot\alpha_s(r)\dot\gamma_s(\alpha(r)) + \dot\alpha_t(r)\dot\gamma_t(\alpha(r))\|^2\,dr \quad (8)$$

where $\alpha : [0,1] \to \Omega$ is a parameterized curve in the space-time and $\dot\alpha$ its tangent vector. Expression (8) may be used to measure change along a dynamically moving document anchor such as the boundary between two book chapters. The space coordinate of such anchor shifts with the version number (due to the addition and removal of content across versions) and so integrating the gradient across one of the two axis as in (7) is not appropriate. Defining $\alpha(r)$ to be a parameterized curve in space-time realizing the anchor positions $(s,t) \in \Omega$ across multiple revisions, (8) measures the amount of change at the anchor point.

3.1 Experiments

The right panel of Figure 1 shows the gradient vector field corresponding to the synthetic version controlled document described in the previous section. As expected, it tends to be orthogonal to the segment boundaries. Its magnitude is displayed by the contour lines which show highest magnitudes around segment boundaries.

Figure 2 shows the norm $\|\dot\gamma_s(s,t)\|^2$ (left), $\|\dot\gamma_t(s,t)\|^2$ (middle left) and the local maxima of $\|\dot\gamma_s(s,t)\|^2 + \|\dot\gamma_t(s,t)\|^2$ (middle right) for a portion of the version controlled Wikipedia Religion article. The first panel shows the amount of change in local word distribution within documents. High values correspond to boundaries between sections, topics or other document segments. The second panel shows the amount of change as one version is replaced with another. It shows which revisions change the word distributions substantially and which result in a relatively minor change. The third panel shows only the local maxima which correspond to edges between topics or segments (vertical lines) or revisions (horizontal lines).

4 Edge Detection

In many cases documents may be divided to semantically coherent segments. Examples of text segments include individual news stories in streaming broadcast news transcription, sections in article or books, and individual messages in a discussion board or an email trail. For non-version controlled documents finding the text segments is equivalent to finding the boundaries or edges between consecutive segments. See (Hearst, 1997; Beeferman et al., 1999; McCallum et al., 2000) for several recent studies in this area.

Things get a bit more complicated in the case of version controlled documents. Segments, and their boundaries exist in each version. As in case of image processing, we may view segment boundaries as edges in the space-time domain Ω. These boundaries separate the segments from each other, much like borders separate countries

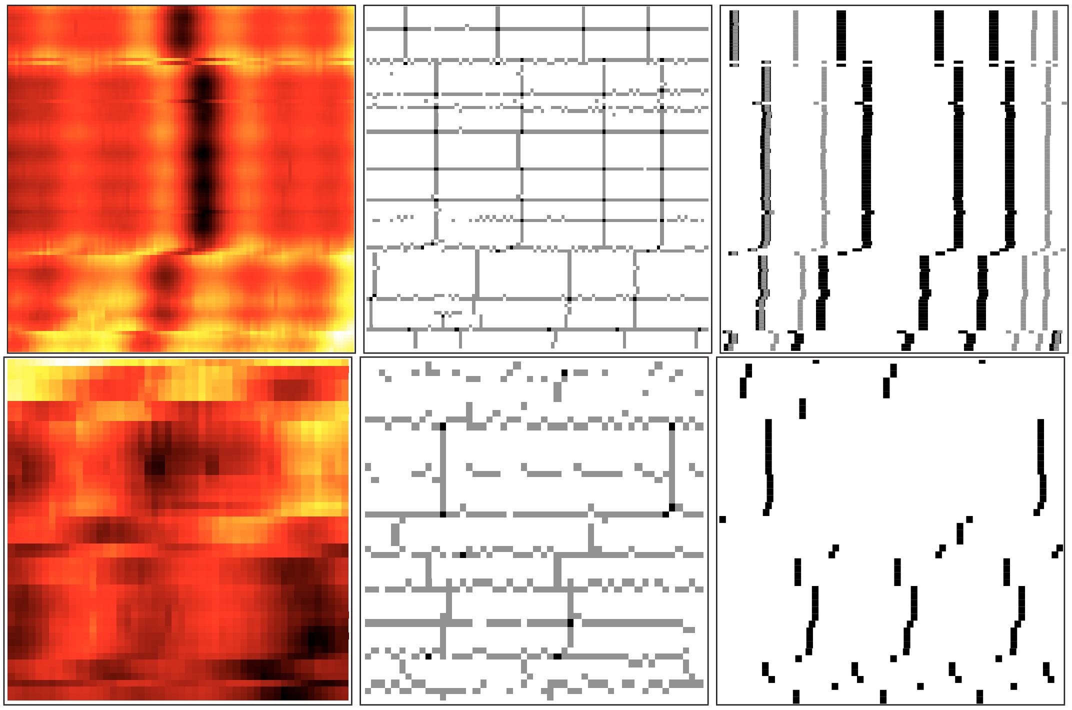

Figure 3: Gradient and edges of a portion of the version controlled Atlanta Wikipedia article (top row) and the Google Wave Amazon Kindle FAQ (bottom row). The left column displays the magnitude of the gradient in both space and time $\|\dot{\gamma}_s(s,t)\|^2 + \|\dot{\gamma}_t(s,t)\|$. The middle column displays the local maxima of the gradient magnitude (left column). The right column displays the actual segment boundaries as vertical lines (section headings for Wikipedia and author change in Google Wave). The gradient maxima corresponding to vertical lines in the middle column matches nicely the Wikipedia section boundaries. The gradient maxima corresponding to horizontal lines in the middle column correspond nicely to major revisions indicated by a discontinuities in the location of the section boundaries.

in a two dimensional geographical map.

Assuming all edges are correctly identified, we can easily identify the segments as the interior points of the closed boundaries. In general, however, attempts to identify segment boundaries or edges will only be partially successful. As a result predicted edges in practice are not closed and do not lead to interior segments. We consider now the task of predicting segment boundaries or edges in Ω and postpone the task of predicting a segmentation to the next section.

Edges, or transitions between segments, correspond to abrupt changes in the local word distribution. We thus characterize them as points in Ω having high gradient value. In particular, we distinguish between vertical edges (transitions across document positions), horizontal edges (transitions across versions), and diagonal edges (transitions across both document position and version). These three types of edges may be diagnosed based on the magnitudes of $\dot{\gamma}_s$, $\dot{\gamma}_t$, and $\dot{\alpha}_1\gamma_s + \dot{\alpha}_2\gamma_t$ respectively.

4.1 Experiments

Besides the synthetic data results in Figure 2, we conducted edge detection experiments on six different real world datasets. Five datasets are Wikipedia.com articles: Atlanta, Religion, Language, European Union, and Beijing. Religion and European Union are version controlled documents with relatively frequent updates, while Atlanta, language, and Beijing have less frequent changes. The sixth dataset is the Google Wave Amazon Kindle FAQ which is a less structured version controlled document.

Preprocessing included removing html tags and pictures, word stemming, stop-word removal, and removing any non alphabetic characters (numbers and punctuations). The section heading information of Wikipedia and the information of author of each posting in Google Wave is used as ground truth for segment boundaries. This information was separated from the dataset and was used for training and evaluation (on testing set).

Figure 3 displays a gradient information, local maxima, and ground truth segment boundaries for

Article	**Rev.**	**Voc. Size**	$p(y)$	**Error Rate**			**F1 Measure**		
				a	b	c	a	b	c
Atlanta	2000	3078	0.401	0.401	0.424	0.339	0.000	0.467	0.504
Religion	2000	2880	0.403	0.404	0.432	0.357	0.000	0.470	0.552
Language	2000	3727	0.292	0.292	0.450	0.298	0.000	0.379	0.091
European Union	2000	2382	0.534	0.467	0.544	0.435	0.696	0.397	0.663
Beijing	2000	3857	0.543	0.456	0.474	0.391	0.704	0.512	0.682
Amazon Kindle FAQ	100	573	0.339	0.338	0.522	0.313	0.000	0.436	0.558

Figure 4: Test set error rate and F1 measure for edge prediction (section boundaries in Wikipedia articles and author change in Google Wave). The space-time domain Ω was divided to a grid with each cell labeled edge ($y = 1$) or no edge ($y = 0$) depending on whether it contained any edges. Method a corresponds to a predictor that always selects the majority class. Method b corresponds to the TextTiling test segmentation algorithm (Hearst, 1997) without paragraph boundaries information. Method c corresponds to a logistic regression classifier whose feature set is composed of statistical summaries (mean, median, max, min) of $\dot{\gamma}_s(s,t)$ within the grid cell in question as well as neighboring cells.

the version controlled Wikipedia articles Religion and Atlanta. The local gradient maxima nicely match the segment boundaries which lead us to consider training a logistic regression classifier on a feature set composed of gradient value statistics (min, max, mean, median of $\|\dot{\gamma}_s(s,t)\|$ in the appropriate location as well as its neighbors (the space-time domain Ω was divided into a finite grid where each cell either contained an edge ($y = 1$) or did not ($y = 0$)). The table in Figure 4 displays the test set accuracy and F1 measure of three predictors: our logistic regression (method c) as well as two baselines: predicting edge/no-edge based on the marginal $p(y)$ distribution (method a) and TextTiling (method b) (Hearst, 1997) which is a popular text segmentation algorithm. Since we do not assume paragraph information in our experiment we ignored this component and considered the document as a sequence with $w = 20$ and 29 minimum depth gaps parameters (see (Hearst, 1997)). We conclude from the figure that the gradient information leads to better prediction than TextTiling (on both accuracy and F1 measure).

5 Segmentation

As mentioned in the previous section, predicting edges may not result in closed boundaries. It is possible to analyze the location and direction of the predicted edges and aggregate them into a sequence of closed boundaries surrounding the segments. We take a different approach and partition points in Ω to k distinct values or segments based on local word content and space-time proximity.

For two points $(s_1, t_2), (s_2, t_2) \in \Omega$ to be in the same segment we expect $\gamma(s_1, t_1)$ to be similar to $\gamma(s_2, t_2)$ and for (s_1, t_1) to be close to (s_2, t_2). The first condition asserts that the two locations discuss the same topic. The second condition asserts that the two locations are not too far from each other in the space time domain. More specifically, we propose to segment Ω by clustering its points based on the following geometry

$$d((s_1,t_1),(s_2,t_2)) = d_H(\gamma(s_1,t_1),\gamma(s_2,t_2)) + \sqrt{c_1(s_1-s_2)^2 + c_2(t_1-t_2)^2} \quad (9)$$

where $d_H : \mathbb{P}_V \times \mathbb{P}_V \to \mathbb{R}$ is Hellinger distance

$$d_H^2(u,v) = \sum_{i=1}^{V} (\sqrt{u_i} - \sqrt{v_i})^2. \quad (10)$$

The weights c_1, c_2 are used to balance the contributions of word content similarity with the similarity in time and space.

5.1 Experiments

Figure 5 displays the ground truth segment boundaries and the segmentation results obtained by applying k-means clustering ($k = 11$) to the metric (9). The figure shows that the predicted segments largely match actual edges in the documents even though no edge or gradient information was used in the segmentation process.

6 Predicting Future Operations

The fourth and final task is predicting a future revision d_{l+1} based on the smoothed representation of the present and past versions $d_1, \ldots, d_l$. In

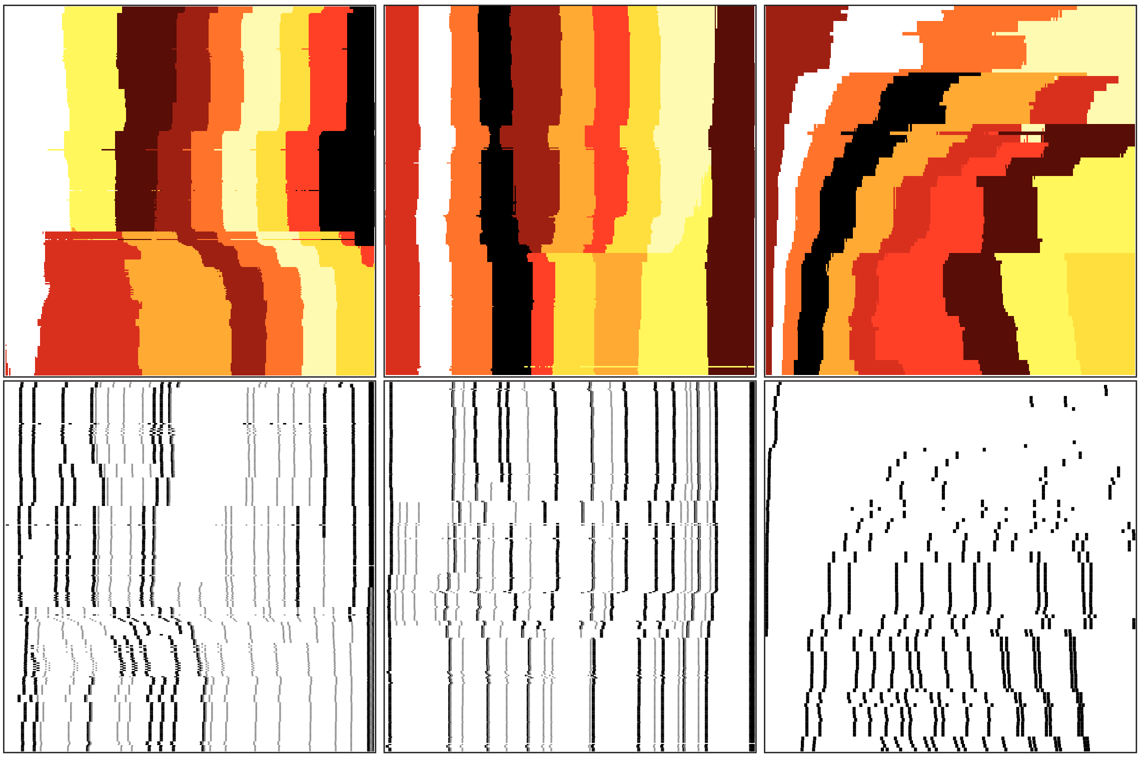

Figure 5: Predicted segmentation (top) and ground truth segment boundaries (bottom) of portions of the version controlled Wikipedia articles Religion (left), Atlanta (middle) and the Google Wave Amazon Kindle FAQ(right). The predicted segments match the ground truth segment boundaries. Note that the first 100 revisions are used in Google Wave result. The proportion of the segments that appeared in the beginning is keep decreasing while the revisions increases and new segments appears.

terms of Ω, this means predicting features associated with $\gamma(s,t), t \geq t'$ based on $\gamma(s,t), t < t'$.

6.1 Experiments

We concentrate on predicting whether Wikipedia edits are reversed in the next revision. This action, marked by a label UNDO or REVERT in the Wikipedia API, is important for preventing content abuse or removing immature content (by predicting ahead of time suspicious revisions).

We predict whether a version will undergo UNDO in the next version using a support vector machine based on statistical summaries (mean, median, min, max) of the following feature set $\|\dot{\gamma}_s(s,t)\|$, $\|\ddot{\gamma}_s(s,t)\|$, $\|\dot{\gamma}_t(s,t)\|)$, $\|\dot{\gamma}_t(s,t)\|$, $g(h)$, and $h(s)$. Figure 6 shows the test set error and F1 measure for the logistic regression based on the smoothed space-time representation (method c), as well as two baselines. The first baseline (method a) predicts the majority class and the second baseline (method b) is a logistic regression based on the term frequency content of the current test version. Using the derivatives of γ, we obtain a prediction that is better than choosing majority class or logistic regression based on word content. We thus conclude that the derivatives above provide more useful information (resulting in lower error and higher F1) for predicting future operations than word content features.

7 Related Work

While document analysis is a very active research area, there has been relatively little work on examining version controlled documents. Our approach is the first to consider version controlled documents as continuous mappings from a space-time domain to the space of local word distributions. It extends the ideas in (Lebanon et al., 2007) of using kernel smoothing to create a continuous representation of documents. In fact, our framework generalizes (Lebanon et al., 2007) as it reverts to it in the case of a single revision.

Other approaches to sequential analysis of documents concentrate on discrete spaces and discrete models, with the possible extension of (Wang et al., 2009). Related papers on segmentation and sequential document analysis are (Hearst,

Article	**Rev.**	**Voc. Size**	$p(y)$	**Error Rate**			**F1 Measure**		
				a	b	c	a	b	c
Atlanta	2000	3078	0.218	0.219	0.313	0.212	0.000	0.320	0.477
Religion	2000	2880	0.123	0.122	0.223	0.125	0.000	0.294	0.281
Language	2000	3727	0.189	0.189	0.259	0.187	0.000	0.334	0.455
European Union	2000	2382	0.213	0.208	0.331	0.209	0.000	0.275	0.410
Beijing	2000	3857	0.137	0.137	0.219	0.136	0.000	0.247	0.284

Figure 6: Error rate and F1 measure over held out test set of predicting future UNDO operation in Wikipedia articles. Method a corresponds to a predictor that always selects the majority class. Method b corresponds to a logistic regression based on the term frequency vector of the current version. Method c corresponds a logistic regression that uses summaries (mean, median, max, min) of $\|\dot{\gamma}_s(s,t)\|$, $\|\dot{\gamma}_s(s,t)\|$, $g(t)$, and $h(s)$.

1997; Beeferman et al., 1999; McCallum et al., 2000) with (Hearst, 1997) being the closest in spirit to our approach. An influential model for topic modeling within and across documents is latent Dirichlet allocation (Blei et al., 2003; Blei and Lafferty, 2006). Our approach differs in being fully non-parametric and in that it does not require iterative parametric estimation or integration. The interpretation of local word smoothing as a non-parametric statistical estimator (Lebanon et al., 2007) may be extended to our paper in a straightforward manner.

Several attempts have been made to visualize themes and topics in documents, either by keeping track of the word distribution or by dimensionality reduction techniques e.g., (Fortuna et al., 2005; Havre et al., 2002; Spoerri, 1993; Thomas and Cook, 2005). Such studies tend to visualize a corpus of unrelated documents as opposed to ordered collections of revisions which we explore.

8 Summary and Discussion

The task of analyzing and visualizing version controlled document is an important one. It allows external control and monitoring of collaboratively authored resources such as Wikipedia, Google Wave, and CVS or SVN documents. Our framework is the first to develop analysis and visualization tools in this setting. It presents a new representation for version controlled documents that uses local smoothing to map a space-time domain Ω to the simplex of tf vectors $\mathbb{P}_V$. We demonstrate the applicability of the representation for four tasks: visualizing change, predicting edges, segmentation, and predicting future revision operations.

Visualizing changes may highlight significant structural changes for the benefit of users and help the collaborative authoring process. Improved edge prediction and text segmentation may assist in discovering structural or semantic changes and their evolution with the authoring process. Predicting future operation may assist authors as well as prevent abuse in coauthoring projects such as Wikipedia.

The experiments described in this paper were conducted on synthetic, Wikipedia and Google Wave articles. They show that the proposed formalism achieves good performance both qualitatively and quantitatively as compared to standard baseline algorithms.

It is intriguing to consider the similarity between our representation and image processing. Predicting segment boundaries are similar to edge detection in images. Segmenting version controlled documents may be reduced to image segmentation. Predicting future operations is similar to completing image parts based on the remaining pixels and a statistical model. Due to its long and successful history, image processing is a good candidate for providing useful tools for version controlled document analysis. Our framework facilitates this analogy and we believe is likely to result in novel models and analysis tools inspired by current image processing paradigms. A few potential examples are wavelet filtering, image compression, and statistical models such as Markov random fields.

Acknowledgements

The research described in this paper was funded in part by NSF grant IIS-0746853.

References

Beeferman, D., A. Berger, and J. D. Lafferty. 1999. Statistical models for text segmentation. *Machine Learning*, 34(1-3):177–210.

Blei, D. and J. Lafferty. 2006. Dynamic topic models. In *Proc. of the International Conference on Machine Learning*.

Blei, D., A. Ng, , and M. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Fortuna, B., M. Grobelnik, and D. Mladenic. 2005. Visualization of text document corpus. *Informatica*, 29:497–502.

Havre, S., E. Hetzler, P. Whitney, and L. Nowell. 2002. Themeriver: Visualizing thematic changes in large document collections. *IEEE Transactions on Visualization and Computer Graphics*, 8(1).

Hearst, M. A. 1997. Texttiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64.

Lebanon, G., Y. Mao, and J. Dillon. 2007. The locally weighted bag of words framework for documents. *Journal of Machine Learning Research*, 8:2405–2441, October.

McCallum, A., D. Freitag, and F. Pereira. 2000. Maximum entropy Markov models for information extraction and segmentation. In *Proc. of the International Conference on Machine Learning*.

Spoerri, A. 1993. InfoCrystal: A visual tool for information retrieval. In *Proc. of IEEE Visualization*.

Thomas, J. J. and K. A. Cook, editors. 2005. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE Computer Society.

Wand, M. P. and M. C. Jones. 1995. *Kernel Smoothing*. Chapman and Hall/CRC.

Wang, C., D. Blei, and D. Heckerman. 2009. Continuous time dynamic topic models. In *Proc. of Uncertainty in Artificial Intelligence*.