

# Events, Interest, Segmentation, Binding and Hierarchy

**Richard Rohwer**

SRI International, 9988 Hibert Street, Ste. 203, San Diego, CA 92131  
 richard.rohwer@sri.com

## Abstract

We advocate the position that unsupervised learning of rich representations requires careful consideration of an issue that usually receives only cursory attention: The definition of a statistical ‘event’, or ‘sample’. Data sets are presumed to have been generated by sampling from some probability distribution that is to be estimated, but there is no general canonical way to select a model for a given data set and define the correspondence between the various components of its joint random variable and particular subsets, or more generally, features, of the data. Any attempt to automate this choice must confront the fact that without a definition of ‘event’, this exercise cannot be formulated as a statistical learning problem. We introduce two supplementary criteria, *information at a distance* and *information contrast*, in order to clear this impasse, and show anecdotal results from using each. We argue that this issue also arises (whether recognized or not) in automated learning of feature hierarchies to form a rich representations, because distinct events are selected at one level of the hierarchy and bound together to form joint events at the next level.

## Imposing Randomness on Constant Data

In the Bayesian statistical approach to machine learning one attempts to interpret a data set as a collection of samples drawn from a random variable. One normally chooses a joint random variable, a conjunction of component variables. In deep models, this construction may be continued through multiple hierarchical levels. The form of this hierarchy is designed to reflect structure noticed or hypothesized in the data, so that structures derived from the data can be interpreted as samples from the joint variable.

The most basic example is tabular data. In this case one almost always chooses to interpret a row of the table as a sample of a joint random variable that has one component variable for each column. The data is segmented into the cells of the table, and each set of cells that forms a row is

bound together to form an event. A more complicated example is natural language text. Such data is usually presented with a natural segmentation into documents. These may be treated as events, but finer-scale features such as words or sentences may be treated as events instead. It is not obvious how to identify these events; how to decide which sequences of character instances should be bound together into patterns to be declared as “words”. The issue is compounded by phenomena such as multi-word units (e.g., “New York”) and spelling variations. Similarly, one might wish to treat all instances of a certain object category such as faces in imagery as events, but doing so requires segmenting those object instances from everything else. Similar remarks apply to treating simple actions expressed over several frames of video data as events. This last example illustrates that statistical events need not correspond to instants of physical time.

We see then, that event instances at one level of a feature hierarchy must be bound together to define event instances at the next level. It seems there must be better and worse ways to create these definitions, which suggests that some adaptive optimization approach would be helpful. However, we cannot formulate learning a definition of “event” as a statistics problem, because “event” would have to be defined first. So any attempt to formulate segmentation learning as a statistical problem must encounter this difficulty.

Segmentation involves identifying an instance of a model variable in terms of some part of the data that is regarded as generated by that event. We can focus on the issue of event definition without treating the full segmentation problem. Instead, we can consider defining an interest operator that can be scanned over every “location” in the data (which we assume is a well-defined notion). The operator is a Boolean-valued function of the data and a location in the data, and outputs “True” at locations that are interest points. Normally, at any particular interest point, the operator would be independent of data outside a region that is “nearby” in some appropriate sense. If we wish, we can regard this region,

approximately, as the segment belonging to the event. It is possible for such segments to overlap. This is appropriate, at least for some types of data such as audio speech. It is essentially impossible, for example, to distinguish the plosives ‘b’ and ‘d’ without looking into a neighboring vowel, so there is no sensible way to draw a boundary between them (Lander and Carmell 1997).

We introduce an objective function called *information at a distance*, and apply it to train an interest operator. This method produces a full (possibly overlapping) segmentation around each interest point. We then introduce a different criterion called *information contrast*, and apply it to define interest points directly, without involving an interest operator.

We present these examples in order to illustrate the introduction of supplementary criteria that guide learning of the concept of ‘event’ within a statistical learning problem. We do not claim that these are especially good criteria, but only that they illustrate two styles of approach to the issue.

## Information at a Distance

The *information at a distance* criterion was motivated from the technique of deriving semantic categories of words by maximizing the mutual information retained between words and their neighboring words (let us say “contexts”) after mapping each into clusters (Kapadia and Rohwer 2010), (Rohwer and Freitag 2004). To set up the problem, one needs a tokenizer to define and identify the instances of “words” and “contexts”, and to define which context instance is to be paired with any given word instance. The motivating question was whether the tokenizer could be learned adaptively in addition to the clusters.

We began by defining a large space of tokenizers. It was constructed out of several adaptable components called *character partitions*, *stop signs*, *move rules*, and *slot rules*. A *character partition* is simply a partitioning of the set of 256 ASCII byte codes into a given number of equivalence classes. In the experiments reported here, we used 2 partitions of 20 classes each, and stochastically adapted the membership of the classes. A *stop sign* is a particular class in a particular partition. We used 2 stop signs and stochastically adapted their definitions. A *move rule* is a sequence of up to 4 steps, each of which consists of a direction (left or right), a minimum and maximum number of characters to move, and a stop sign. A move rule is executed by moving the minimum number of characters in the specified direction and continuing until either the stop sign condition holds or the maximum distance is reached. We used 4 move rules with a maximum of 4 steps and a maximum distance of 50 characters, stochastically adapting the number of steps,

their direction, the minimum number of steps, and the stop signs. A *slot rule* consists of a set of move rules and a *readout* character partition. All the move rules are executed, and the characters from the leftmost to the rightmost positions reached form the slot. The content of the slot is not the raw sequence of characters, but that sequence mapped into the readout partition. We used 2 slot rules with 4 moves rules each and a single readout partition. The choice of move rules and readout partition were adapted stochastically. A tokenizer was defined by a pair of move rules, one for moving from one term to the next, and one for moving from a term to its context, and a pair of slot rules, one for terms and one for contexts.

To adapt the tokenizer, we wanted to capture the intuition that, as in term clustering, high word-context mutual information is good, but also that the same amount of mutual information between distantly separated words and contexts indicates a better tokenization than when present between nearby words and contexts. Therefore, for every candidate tokenizer we computed the co-occurrence statistics between words and contexts separately for every allowed amount of separation between word and context, obtaining a mutual information value for each distance. The corresponding distances and mutual information values were multiplied and summed, and this sum was multiplied by the ordinary word-context mutual information. Tokenizers were adapted by simulated annealing to maximize this *information at a distance* utility. We used the first 50 or 500 documents of our Pakistan News corpus, described elsewhere (Blume 2005), for text data, discarding any co-occurrence counts less than 5. Tokenizers that did not result in left-to-right next-word move rules were assigned zero utility, as were tokenizers that produced overlapping word-word or word-context slots.

No.	{variant}	No. tokens ...	No.	{variant}	No. tokens ...
451	{of}	450	83	{Chief Executive}	35
396	{to}	372 {so} 15 {sh} 4 {th} 3 {31} 1	82	{March 27 (PNS)}	10
336	{in}	274 {In} 56 {IR} 2	81	{March 30 (PNS)}	8
324	{an}	324	48	{Fervez Musharra}	20
278	{s}	146 {t} 121 {Q} 4 {G} 4 {3} 2	46	{Prime Minister}	22
199	{a}	198	41	{Executive General}	26
149	{on}	107 {no} 14 {21} 6 {11} 5 {ho} 5 {12} 3 {22} 3 {81} 2 {oo} 1	36	{Executive, General}	2
147	{n}	58 {o} 48 {j} 19 {h} 13 {R} 6 {1} 1 {2} 1 {3} 1 {%	36	{March 31 (PNS)}	8
146	{of the}	146	30	{March 16 (PNS)}	5
132	{ the}	131 {she} 1	29	{in the country}	17
128	{oo}	105 {ch} 21 {bo} 2	26	{into the country}	1
121	{is}	81 {it} 38 {83} 1	26	{electoral rolls}	20
112	{at}	64 {as} 47	26	{Hurriyet Conference}	18
108	{e}	104 {B} 4	26	{Muslim League}	19
102	{p}	51 {A} 22 {T} 17 {j} 6 {K} 3 {C} 3	25	{emazir Bh}	22
94	{be}	82 {ee} 9 {OX} 1	24	{ammu and Kashmir}	7
88	{hy}	88	22	{ the country}	8
			21	{second phase}	18
			20	{Munir A Sheikh}	15

Table 1. Sample output from two adaptive tokenization runs.

On the left, Table 1 shows the most frequent terms arrived at in a 5-hour simulated annealing run using 50 documents. The first column is the total number of tokens of a lexeme, and the second gives each variant and its number of occurrences. The list starts out with space-segmented prepositions and articles. Further down the list (not shown) one starts to see many longer words and some short phrases, but also many truncated words. On the right, Table 1 shows output from a 75-hour run on 500 documents. The results are very different but no less interesting, showing substantially correct segmentation of multi-word units, especially named entities. This pattern continues further down the list (not shown), though with increasing incidence of segmentations cutting through terms in seemingly inappropriate places.

## Information Contrast

Although it led to plausible segmentation from no more than a sequence of ASCII codes, no knowledge of white space, punctuation, capitalization etc., there are some unpleasantly ad hoc aspects to the *information at a distance* criterion. In particular, it bothered us that there is no obviously correct way to combine distance in characters with information in bits. This led us to define the *information contrast* criterion, which depends on distance only via a rank ordering. Furthermore, it completely dispenses with the interest operator. It depends only on the placement of an ensemble of interest points, and the data in the neighborhood of each point.

For any given placement of interest points, one can determine for each interest point its nearest neighbor and second nearest neighbor (given an arbitrary way to resolve ties). Given features defined in terms of data in the vicinity of an interest point, we can therefore assign a feature value to each interest point. Binding points by the nearest neighbor and second-nearest neighbor relationship, we can therefore perform feature value co-occurrence counts and obtain two mutual information measurements, one for each relationship. We define the *information contrast* of the interest point placement as the difference between the second-nearest neighbor mutual information and the nearest-neighbor mutual information. To maximize information contrast, the placement has to be such that features co-vary less with nearby points than with somewhat more distant points. The intuition is that second-nearest neighbors should lie within the same segment, while nearest neighbors lie in adjacent segments. This is an implausible state of affairs in the 1-dimensional geometry of text, but a plausible one for 2-dimensional images.

Using a set of 200 images of airplanes from the PASCAL data set, we alternated stochastic adaption of

interest point placement, using simulated annealing, with training a feature set based on those points by using K-means clustering on 8x8 pixel patches centered on these points. Figure 1 shows a typical result on a single image. The numbers are K-means centroid IDs. Red (dark) arrows show nearest neighbors and blue (light) arrows show second-nearest neighbors. Though not completely compelling, note that the blue arrows tend to lie within the grass, within the sky, within the white margin, and within the trees, while the red arrows tend to lie between these natural segments.

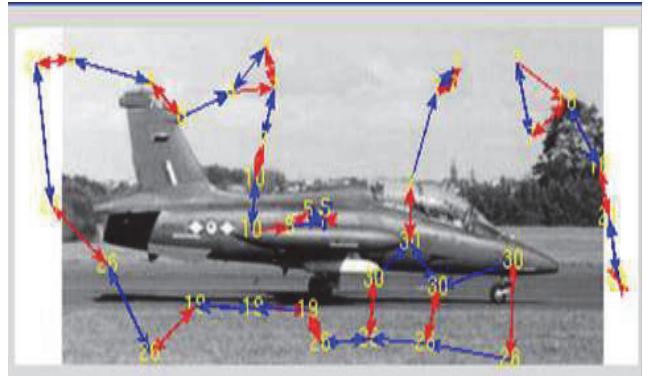


Figure 1. Interest point placement by maximization of information contrast.

## Conclusions

We have argued that unsupervised learning of adaptive hierarchical features involves learning definitions of statistical “events” that link one layer to the next, and that this places the problem outside the framework of statistical machine learning. We introduced additional criteria to make the problem well-posed and showed that sensible results can be obtained in this way, but clearly much more needs to be done to elucidate this issue and develop practical yet principled methods.

## References

- Lander, T., and Carmell, T. 1997. *Structure of Spoken Language: Spectrogram Reading*. <http://speech.bme.ogi.edu/tutordemos/SpectrogramReading/cse551html/cse551/cse551.html>
- Kapadia, S., and Rohwer, R. 2010. *Simmered Greedy Optimization for Co-clustering*. ITNG, Seventh Intl. Conf. on Information Technology, Las Vegas.
- Rohwer, R., and Freitag, D. 2004. *Towards Full Automation of Lexicon Construction*. Lexical Semantics Workshop, HLT/NAACL, Boston.
- Blume, M. 2005. *Automatic Entity Disambiguation: Benefits to NER, Relation Extraction, Link Analysis and Inference*. Proc. 2005 Intl. Conf. on Intelligence Analysis.