Feature Extractions for Data Mining Problems

Our problems are instances of unsupervised clustering, or "clustering without training". More specifically, we are looking at two types of data sets:

1) Static data, such as computer files. We represent computer files as "messages" over some alphabet (ex: bytes over a 256-long alphabet). We are trying to locate clusters of files of the same type, without knowing the number of clusters. Some files may not belong to any cluster. In this case, we assume that we can't rely on obvious information, such as file extension or some keyword(s), to identify the type of files.

2) Time-dependent data, such as the behaviours of users on a given computer network. For each user, we collect some "features" at regular intervals, thus forming a sequence of random vectors {X_t}. Users that perform similar tasks will should tend to form clusters, while an intruder will likely behave in a different fashion. In this case, we are therefore interested in locating "outliers".

In both cases, in view of the typically huge amount of data to process, we are looking for automated solutions for the date mining problems at hand, and the extraction of good features is central. We already did some preliminary study for the static case, and came up with a 4-step process. We considered computer files as a sequence of bytes.
The steps are:

(a) Feature extraction: we used mainly n-grams, and a grammar-building algorithm called SEQUITUR [SEQ].

(b) Correspondence analysis [CA] is used to map our feature vectors into Euclidean space, prune the features and reduce dimensionality.

(c) Hierarchical clustering is then applied to the feature vectors (there are many references for this topic, such as chap. 12 of [JW]).

(d) Extracting cluster(s) from the output of hierarchical clustering (the dendogram). We used a straightforward function of the number of elements and the relative dissimilarity.

The questions of interest to us are:

- Can we improve this "automated" feature extraction in the static case?
- Can we detect and prune "orthogonal features" early in this process?
- Can a similar 4-step process be used in the dynamic case, and how do we select the features? How do we go about "clustering" in view of the dynamic behaviour of this system?
- Can we improve the identification of cluster(s) (point (d))?
- Can we use some feedback mechanism, fixing steps (b)-(c)-(d), to re-visit our selection of features in (a)?

[CA] Greenacre, M.J., "Theory and Applications of Correspondance Analysis", Academic Press, 1984.

[RIP] Ripley, B.D., "Pattern Recognition and Neural Networks", Cambridge University Press, 2001.

[SEQ] C.G. Nevill-Manning, and I.H Witten, "Identifying Hierarchical Structure in Sequences: A linear-time algorithm" Journal of Artificial Intelligence Research, 1997.

[SP] Johnson, R.A. and Wichern, D.W., "Applied Multivariate Statistical Analysis", 4th edition, Prentice Hall, 1999.