ESANN 2010 proceedings, European Symposium on Artificial Neural Networks - Computational Intelligence and Machine Learning. Bruges (Belgium), 28-30 April 2010, d-side publi., ISBN 2-930307-10-2.

Reliability of dimension reduction visualizations of hierarchical structures

Elina Parviainen

Helsinki Univ. of Technology - Dept. of Biomedical Engineering and Computational Science - P.O.Box 2200, FI-02015 TKK - Finland

Abstract. Dimension reduction can produce visualizations of hierarchical structures, like those produced by cluster analysis. So far, reliability of such visualizations has only been assessed with rudimentary means. Here, a method for assessing reliability of such visualizations is developed. It measures how accurately the location of a data point in high-dimensional hierarchy tree can be inferred from a tree based on the low-dimensional visualization. The criterion can be used in point-wise fashion, allowing visual assessment of results, or as average values, for comparing visualizations. Use of the criterion is demonstrated on handwritten digits data, comparing visualizations by three dimension reduction methods.

1 Introduction

Hierarchical presentation, like that created by many clustering algorithms, can capture much of structure of multidimensional data. Hierarchies are often presented as trees, and different layouts, like arranging the leaves in a circle or at different levels, are used to fit more information in a single image. Humans do not need lines to see connections between groups, but can also infer group membership from spatial arrangements. A lot of space would be saved if no external cues about group memberships would be drawn. Therefore dimension reduction to 2D would be a good way to present hierarchies, if we could make sure the visualization shows the structure of the hierarchy correctly.

Many dimension reduction methods can use as inputs other metrics than euclidean. Specifically, cophenetic distances (a measure for closeness of two points in a hierarchy, see Sec. 2), can be used. This idea has been used with MDS [1] but reliability of the resulting visualizations has only been measured by correlation between real and visualized distances. Correlation is a very rough measure, saying little about what conclusions can reasonably be made based on visualizations, and whether all parts of the resulting image are equally reliable.

In this work, we develop a measure of visualization reliability for hierarchical structures. When used in point-wise fashion, it indicates unreliable areas in the visualization at different levels of granularity. Used as average values, it allows comparison of visualizations.

The main idea is explained in Sec. 3, and an algorithm for computing the criterion is developed in Sec. 4. Some limitations are discussed in Sec. 5, and Sec. 6 demonstrates the use of the criterion on handwritten digits data.

ESANN 2010 proceedings, European Symposium on Artificial Neural Networks - Computational Intelligence and Machine Learning. Bruges (Belgium), 28-30 April 2010, d-side publi., ISBN 2-930307-10-2.

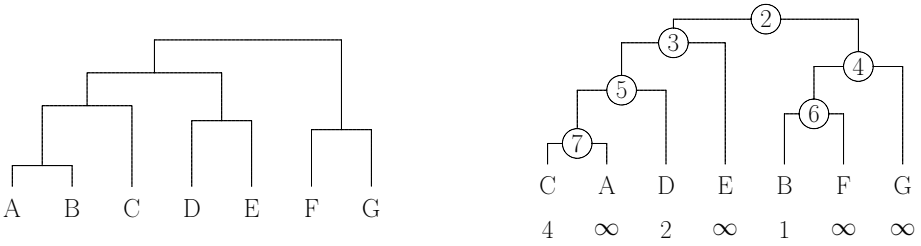


Fig. 1: An example of reliability levels of points. Target tree is on the left and 2D tree on the right. Cophenetic distance (drawn as height of the branch) determines the order of cuts if clusters are created based on this tree. The circles show how many clusters would result if the tree was cut here. Numbers in bottom row show the reliability levels. Point B should have been in the left branch, so it is only seen reliably if whole data in considered one cluster. D is in right place if two clusters are used, but using three would erroneously place it together with A and C. When bottom of the hierarchy is reached, A and C should not stay together, but it is not clear which one is the correctly placed one; their reliability levels could as well be exchanged.

2 Cophenetic distances

Hierarchical clustering arranges points into a binary tree. Each node has height, which equals the distance (as determined by the chosen linkage method) between its children. Cophenetic distance between two points is the height of the node where the points are first placed into the same cluster.

We can use matrix of cophenetic distances in any dimension reduction method which can be modified to use a non-euclidean distance between points. As the cophenetic distance is only defined between two data points, it cannot be used with methods which need to compute distances to other points (e.g. to codebook vectors in self-organizing maps).

3 Comparing hierarchies

We compare two trees, one resulting from clustering the original data (from now on, called the "target tree"), and the other created based on the visualization (the "2D tree"). Ideally, the hierarchical structure we infer from the visualization should match that of the target tree.

We are basically trying to measure whether what a user sees on the screen is what the clustering algorithm has seen in the high-dimensional space. Therefore, clustering in 2D space should use any linkage method whose results the user finds intuitive.

A visualization of clusters often has structure at several granularity levels. We can look for most prominent groupings only, or we can concentrate on sub-structure in a particular area. At each level, the question we are interested in

```
main:
    for all points, set reliability level to ∞
    rel_level(root of target tree, root of 2D tree)

function rel_level(N_T, N_2):
    P_T := pts(N_T), P_2 := pts(N_2), W := P_2 \ P_T
    Q := points of W with reliability level ∞
    set reliability level of points in Q to N_2.parent.cuts_into − 1
    if W = P_2 or is_leaf(N_T) or is_leaf(N_2), return, endif
    L = (pts(N_T.left) ∩ pts(N_2.left)) ∪ (pts(N_T.right) ∩ pts(N_2.right))
    R = (pts(N_T.left) ∩ pts(N_2.right)) ∪ (pts(N_T.right) ∩ pts(N_2.left))
    if |L| > |R|
      rel_level(N_T.left, N_2.left), rel_level(N_T.right, N_2.right)
    else
      rel_level(N_T.left, N_2.right), rel_level(N_T.right, N_2.left)
    endif

function pts(node): return all points in the subtree starting from node
```

Fig. 2: Algorithm for determining the reliability levels of points.

is: does this data point really belong to this cluster? At coarse level, it may be enough to have the data point in the correct quadrant of the image; a more detailed analysis might require a group of a dozen points to be correctly shown together.

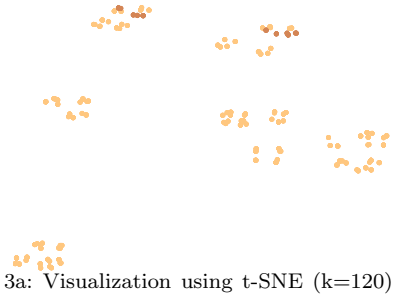
To capture quality at varying levels of detail, we can turn the question around and ask: to what level of detail must we go before we see this point in a wrong place? *Reliability level* of a point is defined to be the number of cluster into which the data can be divided before the point associates with a wrong cluster.

An example of reliability levels is shown in Fig. 1.

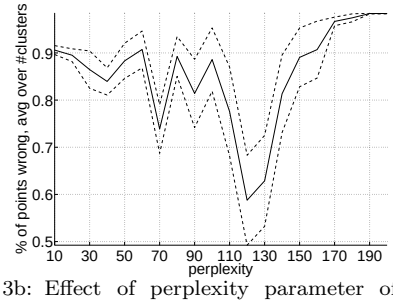
4 Algorithm

In case of hierarchical structures, number of clusters determines a location in the hierarchy tree. To form clusters, branches are cut in order of decreasing height. Each cut divides an existing cluster in two. Thus, each internal node of the hierarchy tree is associated with a number telling how many clusters are created by cutting the tree at this node.

We use the correspondence between number of clusters and tree nodes to develop an algorithm for determining reliability level. Pseudocode is shown in Fig. 2. We traverse both target tree and 2D tree at once. At each node, we find in 2D node those points which should not be there, according to the corresponding target node. If the wrong points have not yet been marked as unreliable, we conclude that cutting the tree at parent level made these points wrong, and update the reliability level accordingly. Correspondence between



3a: Visualization using t-SNE (k=120). For color scale, see Fig. 4.



3b: Effect of perplexity parameter of t-SNE on reliability levels (mean with its standard error, ten runs).

branches is established by checking which 2D branch has greater overlap with left (resp. right) branch of target tree, and recursion is continued until all points are considered wrong or a leaf in either tree is reached.

5 Limitations

Reliability levels are very clearly a tool for cluster analysis, and better suited for work in coarse levels in hierarchy than in small scale. Reliability level of a point tells us, when the point is first placed in a wrong cluster. If we were to continue cutting the tree, we would finally reach the level where points would again group with correct points only (at limit, being in the same cluster just with themselves). We argue that going to too much detail is not cluster analysis anymore. If we are mainly interested in fine detail, we should not use cophenetic distances at all, since they necessarily abstract away part of the local structure to better emphasize groups.

Trees to compare should be somewhat similar. Even in completely different trees, there will be some overlap between target branches and 2D branches. Because of this, the algorithm sees some points in "correct" clusters, even if the branches overlap just by chance. The same is likely to happen in lowest levels of any hierarchy trees (as was seen in the example in Fig. 1, where one of equally correct points was arbitrarily assigned higher reliability than the other). Visualizations can probably catch many of these situations by showing a mixture of correct and incorrect points in the same cluster, but care should be taken not to overinterpret results for dissimilar trees and at lowest hierarchy levels.

Visualizations with reliability levels concentrate on showing points which have ended in wrong place, but say nothing about points which are missing. It should be remembered, that a cluster marked as reliable in the visualization might not contain all points which belong to that cluster in the target tree.

Fig. 4: Reliable and unreliable areas of a visualization using Sammon mapping. Colors correspond to reliability levels. All points are reliably placed in 1–6 clusters; at the next granularity level, about 7–20 clusters, darkest points may be wrong. Medium dark means reliability problems start at scale of 21–50 clusters, and all points more reliable than that are drawn with lightest color.

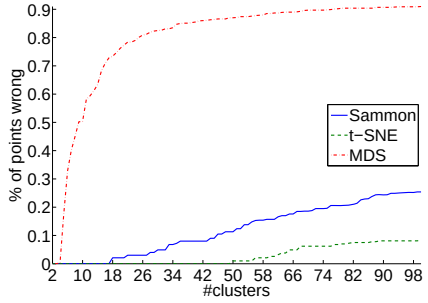


Fig. 5: Comparison of methods (to avoid local minima, best of ten runs for each method is used). 2D tree was built with single linkage for Sammon mapping and Ward's criterion for others (of linkage methods available, the one giving best results was chosen for each visualization).

6 Experiments

We tested the visualizations on a sample images of handwritten digits from US Postal Service data set. 100 samples from each of ten classes were used. Machine learning algorithms tend to see the images differently from humans, grouping the digits based on stroke directions, loops and other graphical features rather than their semantics. Therefore, clusters seen by an algorithm and classes of digits don't necessarily match. Running a clustering algorithm on this data produced a rather complicated structure, a good test for visualization abilities of different methods.

We built a tree of the high-dimensional data using hierarchical clustering with Ward's criterion, and used the corresponding cophenetic distances as inputs to MDS [2], Sammon mapping [3] and t-SNE [4].

In spite of repeated runs, MDS produced a degenerate solution, placing most points on top of each other in three clusters (not shown). Visualization created with t-SNE is shown in Fig. 3a, and that of Sammon mapping in Fig. 4.

Sammon mapping seems to have captured the structure well, showing a clear

hierarchical structure both at coarse and finer level. T-SNE shows highest levels of hierarchy clearly, but seeing detail is difficult without zooming into the image. In this sense, Sammon mapping visualizations are more readable. In average reliability, Sammon mapping falls second to t-SNE (see comparison in Fig. 5).

Cophenetic distances in t-SNE seem to require greater perplexity values than euclidean distance (k=120 was used in the visualization shown; k=30 produced good visualizations of raw data). The results seem to be sensitive to perplexity parameter (Fig. 3b). In certain range (around values 120..130) results are very good, but decay rapidly for smaller or large values.

7 Conclusions

We presented a novel method for assessing reliability of visualizations which show hierarchical structures of high-dimensional data as low-dimensional images. Comparison is based on how accurately a hierarchy tree derived from the low-dimensional image matches that derived from the high-dimensional data.

Reliability level of a point is the height of the low-dimensional tree node below which the point is seen in incorrect cluster. This point-wise criterion allows easy visualization of results, but can also be used as an average value for comparing visualizations. Important feature of this criterion is that it allows reliability analysis at different levels of granularity. That said, the criterion is mainly a tool for cluster analysis, and at very small scales, e.g. in analyzing preservation of neighborhoods in visualizations, other methods should be used.

We used the criterion to compare visualizations created with three dimension reduction methods (MDS, Sammon mapping, t-SNE), using cophenetic distances as input. Sammon mapping visualizations showed structure in both coarse and fine scales in very intuitive way. On the other hand, the reliability of the results was lower than that of t-SNE. Problem of t-SNE visualizations was that seeing any finer level structure required zooming into the image, which may be confusing for the user. Both Sammon mapping and t-SNE performed clearly better than MDS. Although cophenetic distances are mostly used together with MDS, our results show that MDS may not be the best choice for all data sets.

References

[1] Bart Alewijnse, John Nerbonne, Lolke J. van der Veen, and Franz Manni. A computational analysis of Gabon varieties. In *Proc. of RANLP workshop on computational phonology*, pages 3–12, 2007.

[2] W. S. Torgerson. Multidimensional scaling: I. theory and method. *Psychometrika*, 17:401–419, 1952.

[3] John W. Sammon. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, C-18(5):401–409, May 1969.

[4] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.