Using Graphs to Analyze High-Dimensional Classifiers

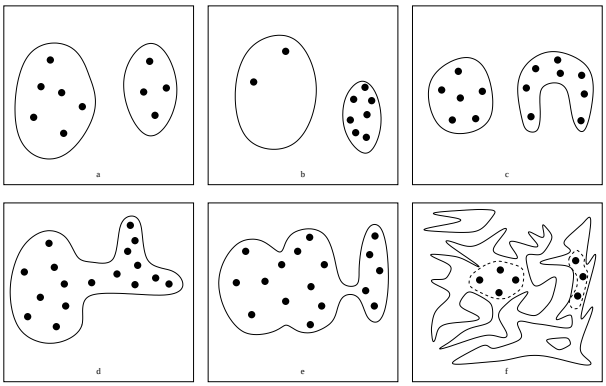
Ofer Melnik and Jordan Pollack, Volen Center for Complex Systems
Brandeis University, Waltham, MA, USA
melnik@cs.brandeis.edu pollack@cs.brandeis.edu

In this paper we present a method to extract qualitative information from any classification model that uses decision regions to generalize (e.g. neural nets, SVMs, graphical models etc) that is independent on the dimensionality of the data and model. The qualitative information can be directly used to analyze the classification strategies employed by a model, and also to directly compare strategies across different models. We apply the method to compare between two types of classifiers using real-world high-dimensional data.

It is very difficult to gauge the qualitative performance of high dimensional classification methods. The most common form of analysis usually consists of comparing raw performance scores. But, as a simple one-dimensional measure, it does not lend any insight as to what a model's advantages and shortcomings may be. This problem is exacerbated when we want to compare across different methods that solve the same problem, across a bank of different neural networks, different graphical model's, different SVMs etc....

The way that a model can form sets with an infinite number of points from a finite training sample is intrinsically tied in to how it can generalize. Many of the models used today for classification such as feed-forward neural networks, support vector machines, nearest neighbor classifiers, decision trees and many Bayesian networks generate classification sets that are manifolds or manifolds with boundaries. This is a strong local property of the sets, which implies the existence of neighborhoods. That is, most of the points in the set have a neighborhood surrounding them such that all points in the neighborhood are also part of the set. Thornton [5] demonstrated that many of the datasets in the UCI machine learning repository [2] contain data points that exhibit neighborhood properties, and as such are amenable to generalization by manifold type classifiers.

Given this commonality of generalization method between classifiers then what differentiates between them is how they individually partition the training points into decision regions. A classifier might only use separate convex decision regions to classify. Thus separating sample points explicitly by completely buffering them from each other in separate decision regions. However most interesting classifiers use more complex decision regions to organize the sample points. The points are organized into decision regions with concavity, thus creating a partitioning of the points without explicitly placing them in disconnected decision regions. In a sense this partitioning allows a finer grain of differentiation since points may be closely associated by being in a convex sub-component of the decision region or be distantly associated through a "network" of other convex sub-components. In figure 1 we see some of the variations possible in decision region structure:



a) The number of decision regions used. b) The number of sample points in a decision region. c) A decision region may be convex, such that any two sample points are within line of sight of each other. Or concave, some of the points inside it would share neighborhoods while others would be separated. d) Concave decision regions can be decomposed into convex subcomponents. e) The degree that the convex subcomponents are attached together in a concave decision region is indicative of how separated they are. The left pair are two strongly attached convex subcomponents, but the right pair are very weakly connected. f) Relevant part of decision regions.

Figure 1: Examples of some of the variations possible in decision region structure.

Our aim is to analyze the decision regions of classifiers. We would like a means to extract the different decision regions of classifiers and decompose them individually. However, dimensionality influences the

complexity of the models. For example, a neural network can have a number of decision regions that is exponential in the input dimension, where the complexity of the individual decision regions is also exponential in the input dimension [4].

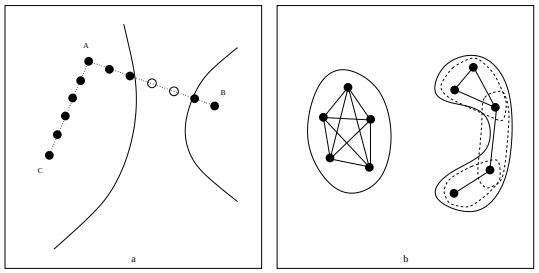
However, there is an intrinsic discrepancy between the potential complexity of the model, the complexity of the data and the relevant complexity of a trained model. In figure 1f we see an example of this. The model could be representing some highly complex decision regions. However, the actual data points only reside in a simple part of the of decision regions. And with respect to these data points, the model is basically enclosing them in two pseudo-decision regions, one convex and one slightly concave. Hence the additional complexity of the model is just artifactual.

Our analysis method tries to extract this relevant complexity, by elucidating the properties of the decision regions in the vicinity of the data points. This is done not by directly examining the decision regions, but rather by examining the effects that the decision regions have on the relationships between the data points. This is what makes the analysis method practically independent of the model type and dimensionality of the input space.

The rest of the paper is organized as follows: We first introduce the basic analysis method which uses graphs to describe the structure of decision regions. Following, we give an example of its application to a neural network that classifies points in a three-dimensional space. The fact that it is three-dimensional allows us to visually compare the graph with the actual network decision region. In the section after, we refine the graph analysis method, and explain how to automatically decompose the graph into the subgraphs which correspond to decision region subcomponents. We conclude with an example where we analyze two different types of classifiers on a high-dimensional problem. The graph analysis method allows us to clearly show that one of the classifiers discovered that one of its classes is really composed of two subclasses.

Low Level Analysis

The fundamental generalization quality of manifold type classifiers is the existence of common neighborhoods between points, which is what our analysis method tries to detect. As input we are given two things, the classifier we wish to analyze and relevant labeled sample points, possibly the training data. It is important that the sample points embody the part of the input space that is of interest, otherwise we would be analyzing the classifier's artifacts and not the relevant regions. From now on, when we refer to decision regions we will mean only the relevant portions of the decision regions.



a) The connectivity graph is generated by sampling between the sample points. In this case we see how sampling between points A and B detects a boundary, but points A and C share a neighborhood. b) A connectivity graph for two decision regions, one convex and one concave.

Figure 2: The analysis method

Figure 2a illustrates how the analysis method works. We take all pairs of points with the same classification label (in this case points A,B and C). Between each pair we extend a line segment in the input space. We then sample along this line with respect to the classifier. In other words, we find a series of points in the input space along the line and apply the classifier to them. What we look for is a break in the connectivity, a change in the classification label in one or more of the points. Such a change implies that between the two points there is a decision region boundary, and the two points do not share a common elliptic neighborhood.

With this connectivity information we construct a graph in the mathematical sense. In this graph each sample point is assigned a vertex, and the edges are the actual connectivity information. That is, if two points are connected in the actual input space with respect to the classifier then their vertices are connected in the graph.

This connectivity graph can tell us three basic pieces of information: What points reside in separate decision regions, if points are colocated in a convex decision region, or if points reside in a concave decision

region. Moreover, in the latter case we can decompose this concave decision region and find what points reside in its different convex subcomponents.

Figure 2b illustrates how the graph relates these three pieces of information. If decision regions are disconnected then the graphs of the points they enclose are also disconnected. In the figure we see this with respect to two decision regions, whose internal points form two disconnected graphs in the connectivity graph. When points are in a convex decision region then by definition they are fully connected and as such form a clique in the graph. We see this convexity property in the left decision region– it is convex and hence its graph is fully connected. The right decision region is not convex and so its graph is not fully connected. However cliques within its graph represent convex subregions of this concave decision region. In this example decision region there are three cliques, representing a decomposition into three convex subcomponents.

Analyzing a three-dimensional neural network

A 15 hidden-unit threshold neural network was trained to predict whether a thrown ball will hit a target. As input, it received a throwing angle, an initial velocity and a target distance (figure 3), only three inputs which makes it possible to visualize its decision regions. After iterations of back-propagation and hill-climbing it achieved an 87% success rate on the training data. This system can be easily solved analytically, and the analytic decision region is shown in figure 4 contrasted with the neural network decision region which was extracted using the DIBA algorithm [4].

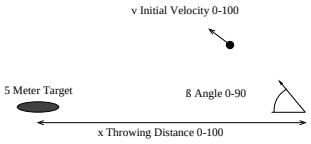


Figure 3: The classification task of the ball throwing network is to predict whether a ball thrown at a certain velocity and angle will hit a target at a given distance.

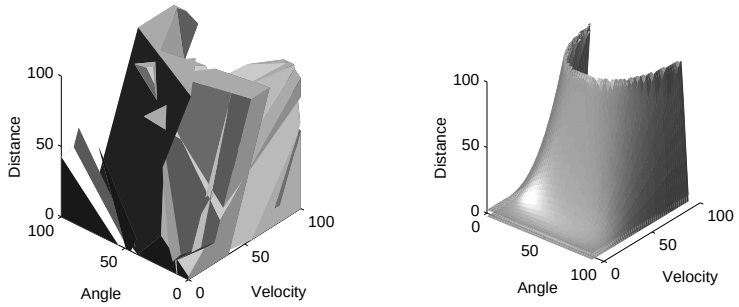


Figure 4: The decision region of the ball throwing network (left) contrasted with the analytic decision region (right).

Using the first 78 of the positive training points, a connectivity graph was generated, as seen in figure 5b. The graph was drawn using a spring-gravity type algorithm [3], where the edges are modeled as springs in a physical model. This drawing algorithm has the property of making highly interconnected vertices cluster together.

In the graph we can discern four different clusters that practically form cliques. We can label the vertices based on which cluster they belong to (if they belong to any cluster). Since this is a three dimensional classifier we can plot the position of the actual sample points corresponding to the labeled vertices inside the decision region (figure 5a). In this figure we can literally see that the points that make up each of these clusters correspond to different "slabs" or conspicuous convex subregions in the actual neural network concave decision region. Also notice how the connections between the clusters in the graph correspond to the relationships between the actual subregions.

Since we have separated the points into convex subregions we can also analyze their geometric properties. For example, by performing primary component analysis [1] (PCA) on each of these clusters of points, we can discern their dimensionality and also their orientation. Figure 6 shows the three eigenvalues for each of the clusters as well as for all the points combined. The eigenvalues of these clusters all have a practically negligible third eigenvalue. This indicates that they all form part of a decision region which takes up little volume in the input space, rather it is almost a two-dimensional embedding in a three-dimensional space. As seen in the graph, this is not a property we could have discerned by just performing a PCA of all the points since the eigenvalues of all the points have a sizeable magnitude in all three dimensions.

Higher Level Graph Analysis

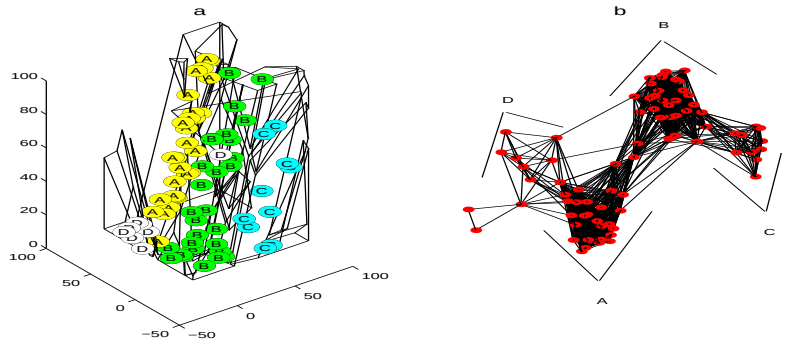


Figure 5: a) The labeled points extracted from the connectivity graph superimposed in their correct position within the decision region. b) The connectivity graph of the decision region in figure 4 using 78 internal points. The vertices are labeled by association to four different clusters.

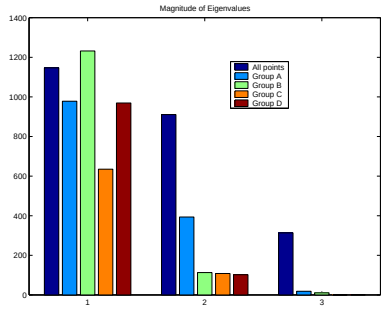


Figure 6: The eigenvalues of the PCA analysis for each of the groups contrasted with the eigenvalues of all the points taken together.

The higher level analysis method proposed extracts two pieces of information from the connectivity graph: First, which points are colocated in the same convex subregions. Second, how do the convex subregions combine to form the original decision region. The basic assumption of the method is that multiple points inhabit the convex subcomponents of the decision region. This is a fair assumption if we expect the classifier to have generalization properties, since in a manifold type classifier only by recognizing neighborhoods is generalization possible– which implies the aggregation of points together into convex subcomponents of decision regions. One point to note is that it is easy to test if a clique really corresponds to a convex subregion, since if we suspect that a group of points are located in a convex subregion then we can generate new points between them and test if they also form part of the clique.

In the first stage of the method we seek to group sample points with similar properties, to find points in similar locations with respect to the decision region. This is done as following: In the connectivity matrix associated with the graph, each row enumerates the edges of a vertex in the form of a binary vector. If the hamming distance between two such vectors is negligibly small then we group their respective vertices together. This means that all vertices in a group are mostly connected to the same vertices and disconnected from the same vertices. The logic of this selection mechanism is that if a group of points are in the same convex subregion then they should all be connected with each other, but by the geometry and topology of the decision region they should also all be connected to the same vertices outside their immediate clique. Therefore they should all have a similar connectivity pattern and that is what we look for in forming groups.

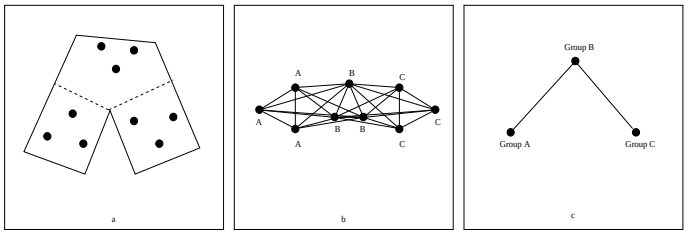


Figure 7: a) A concave decision region housing nine points in different convex subcomponents. b) The connectivity graph for the points in the decision region. c) The group graph associated with the labeling in the connectivity graph.

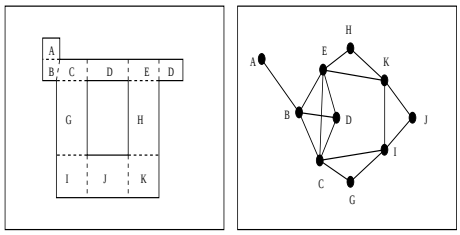


Figure 8: An example of a more complex concave decision regions and its group graph.

For example, consider the concave decision region in figure 7a and its connectivity graph in figure 7b. If we label the vertices based on similar connectivity, we end up with three groups of points as shown. Each group represents a convex subregion, as each group's vertices forms a clique. However, these groups are different with respect to their position in the decision region, which is seen by their intergroup connectivity.

In the second stage we wish to simplify the original connectivity graph, in order to gauge the relationships between the convex subregions. This is done as following: For each group, we take all its vertices and merge

them, thus transforming the group into one labeled vertex with the same intergroup connectivity as the group originally had. We are left with a much smaller and sparser graph that relates the relationships between the groups, their intergroup connectivity. In figure 7c we see this operation performed on the connectivity graph in figure 7b. This new graph shows us that with respect to the sample points the original decision region partitioned the space into three groups such that one of the groups (group B) is connected to both of the other groups, but that the other two groups are not directly connected. Notice how the graph is similar in structure to the actual decision region.

Figure 8 shows the group graph for a more complicated concave decision region. Cliques in the group graph represent subcomponents that may be combined to form larger convex subregions, for geometrical analysis of the points in the subregion. In the trivial case, every edge in the graph is a clique and represents a potential combined convex region. Another graph characteristic is loops of cliques. Loops of cliques in the group graph represent the existence of holes in the decision region. As illustrated the group graph relates the the actual partitioning between convex subregions, which subregions are directly connected, which are distantly connected and what the connection paths are.

Comparing two classifiers: A high-dimensional example

The UCI repository [2] contains a dataset contributed by Alpaydin and Kaynak of handwritten digits. There is a preprocessed version of the dataset, where the 32 by 32 images are shrunk to 8 by 8 by counting the number of pixels in each 4 by 4 of the original. This training set contains 3823 samples from 30 people.

Using the preprocessed dataset the following classification task was fabricated. The data corresponding to the numerals 3 and 4 were assigned to one class, while the remaining numerals were assigned to a second class. Thus the task consisted of classifying a 64 dimensional input into one of two classes.

Two classifiers were used, a sigmoidal feed-forward neural network with one hidden layer of 7 units and a K-nearest neighbor classifier with K set to 9 [1]. The network was trained using conjugate gradient [1] until it reached perfect classification on the test data.

In order to make the connectivity graph more presentable, only the first 63 cases of the 3-4 class were used to draw it. However in the additional levels of analysis 300 exemplars were used.

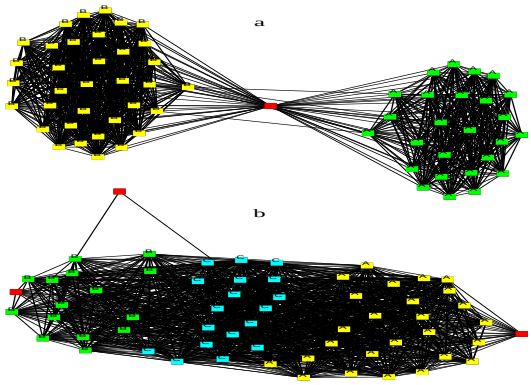


Figure 9: a) The connectivity graph of the neural network. b) The connectivity graph of the KNN classifier.

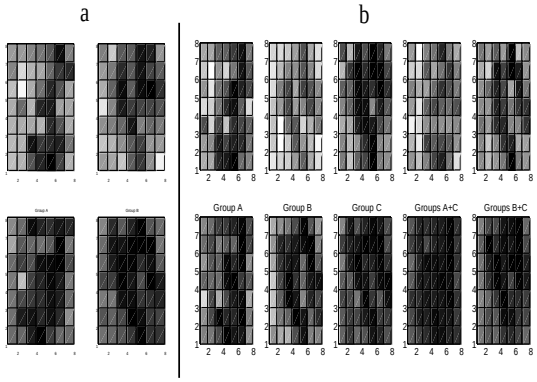


Figure 10: a) A PCA analysis of the convex subregions in the network connectivity graph. b) A PCA analysis of the convex subregions of the KNN connectivity graph.

Figure 9a shows the connectivity graph for the neural network. Since the graph is connected it consists of one decision region. However it is apparent that this graph is illustrating a concave decision region because the graph consists of two highly connected regions with only very sparse connectivity between them. The points were labeled using the labeling method described above at a 90% Hamming similarity, which labeled the points as expected into two classes corresponding to these practically clique subregions. In the connectivity graph the clusters are almost completely disconnected, therefore we can not draw a group graph, since a group graph is only constructed when the groups make up parts of larger convex subregions.

When we examine the actual numeral associated with the labeled points we realize that the points

associated with the first label all correspond to the threes! And all the points with the second label correspond to the fours! What this means is that the network discovered that the 3-4 class really consists of two subclasses and divided its decision region to clearly separate between them. Suppose that we didn't know that the class was decomposable and wanted to know what the subregions were that the neural network found. As before, since the subregions are convex we can analyze them using PCA. In figure 10a, for each group, we took the mean of the points, in the top half we added the first 20 eigenvectors of the PCA normalized by their standard deviation, and in the bottom half we subtracted the same values. This gives a coarse approximation of the scope of the decision region. As can be seen, the left images correspond to threes and the right to fours, so we can literally see that the two subregions correspond to two logically separate subclasses.

Figure 9b shows the connectivity graph for the K-Nearest Neighbor classifier. Again, it is a connected graph and hence has one decision region. This graph doesn't lend itself to a simple visual analysis, since it is more dense. However, when we apply the labeling method at 80% Hamming similarity we get three labeled classes. The group graph analysis of the three labeled sets shows that the vertices in group C are connected to both group A and B, but that there are very few connections between groups A and B directly. Therefore, the group graph is of the form we saw in the example in figure 7. In figure 10b we see the same form of PCA analysis as in the neural network example on the three different groups as well as on the two cliques of the group graph. As can be seen, group A corresponds to threes, groups B and C correspond to fours, as is the case with the composition of groups B and C, but the images of the composition of groups A and C are not interpretable. This goes with what we know about how the data is structured, a convex subregion consisting of both threes and fours would have to contain spurious data, and lead to a malformed classification set. When we examine the actual numerals associated with the labeled points, we see that the A labeled points do correspond to threes, and the B and C labeled points correspond to fours.

Both of the classifiers realized that the points making up the 3-4 class are not homogeneous. This is demonstrated by the fact that both classifiers used a concave decision region to house the class's points. However the discrepancy between them lies in how clearly they realized what the two subclasses are. The neural network made a very clean distinction, clearly dividing the space between the threes and fours. Where as the K-Nearest neighbor classifier divided some of the threes completely from some of the fours (groups A and B), it did not differentiate between the threes in group A and the fours in group C, hence we would expect potential misclassification in that region of the input space.

Conclusion

Many classifiers operate by constructing complex decision regions in the input space. These decision regions can be few or many, convex or concave, have large or small volumes etc. By focusing on the sample points enclosed in these regions we have demonstrated a method to extract these properties which is independent of the classifier type or the dimensionality of the input space. It thus allows us not only to analyze individual high-dimensional classifiers but to compare completely different classifier models on the same problems. We have demonstrated this by comparing a neural network and KNN classifier on a handwritten digit classification problem, and demonstrating fundamental differences in their generalization strategy.

We feel that this method is a significant contribution in helping to unite a field with many models and approaches by giving an analysis tool which addresses their greatest common denominator, their method of generalization, thus allowing the qualitative comparison of present and future high-dimensional classifiers.

[1] C.M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.

[2] C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998.

[3] I. Bruss and A. Frick. Fast interactive 3-d graph visualization. In *Proceedings of Graph Drawing '95*, pages 99–110. Springer-Verlag, 1995.

[4] O. Melnik and J. Pollack. Exact representations from feed-forward networks. Technical Report CS-99-205, Brandeis University, 1999.

[5] C. Thornton. Separability is a learner's best friend. In J.A. Bullinaria, D.W. Glasspool, and G. Houghton, editors, *Proceedings of the Fourth Neural Computation and Psychology Workshop: Connectionist Representations*, pages 40–47. Springer-Verlag, 1997.