Context Driven Exploratory Projection Pursuit

Mohit Dayal [*][†]

January 17, 2013

Abstract

In this paper, we propose a new algorithm for exploratory projection pursuit. The basis of the algorithm is the insight that previous approaches used fairly narrow definitions of interestingness / non interestingness. We argue that allowing these definitions to depend on the problem / data at hand is a more natural approach in an exploratory technique. This also allows our technique much greater scope of applicability than the approaches extant in the literature. Complementing this insight, we propose a class of projection indices based on the spatial distribution function that can make use of such information.

Finally, with the help of real datasets, we demonstrate how a range of multivariate exploratory tasks can be addressed with our algorithm. The examples further demonstrate that the proposed technique is quite capable of focussing on the interesting structure in the data, even when this structure is otherwise hard to detect or arises from very subtle patterns.

Keywords: Multivariate Data Exploration, Spatial Distribution Function, Test of Distribution, Dimension Reduction

1 Introduction

The importance of exploring data, before embarking on a formal analysis - especially through visualizations - has long been recognized in the statistics community. However, the application of

[*] Researcher, Applied Statistics and Computing Lab, Indian School of Business, Hyderabad (mohitdayal2000@gmail.com).

[†] The author would like to thank Professor Probal Chaudhuri at the Theoretical Statistics and Mathematics Unit, Indian Statistical Institute, Kolkata for his invaluable comments and suggestions on an early draft of this paper and Professor Galit Shmueli, SRITNE Chaired Professor of Data Analytics and Associate Professor of Statistics and Information Systems at the Indian Schoxol of Business, discussions with whom greatly helped the exposition of this paper and also opened up directions for future research.

this general principle to a multivariate setting is neither easy nor straightforward, since most of the common visualizations like scatter plots or histograms are intrinsically 1, 2 or 3-dimensional. If one wishes to use these familiar tools for multidimensional data, then some dimension reduction must be affected.

This reduction may be accomplished either linearly or non-linearly. The simplest linear dimension reducer is the projection. While they may appear simplistic, linear projections have important advantages over their non-linear cousins. The most important is their ability to easily ignore any arbitrary set of variables in the multivariate dataset. Thus they are not easily led astray by noisy variables that contribute little to structure. Apart from this, linear projections are quick to compute and easily interpretable.

Linear projections behave as smoothing operators - in that they can only obscure structure, but never enhance it. In other words, "Any structure seen in a projection is a shadow of an actual (usually sharper) structure in full dimensionality" (Friedman, 1987). Thus, every projection of multivariate data reveals something about it. Then, if we adopt no further criteria, we have no means of preferring any one projection over the other. This insight underlies the grand tour (Asimov, 1985), where one generates a space filling, dense sequence of projection matrices onto which the data is projected. The procedure thus comes closest to the ideal of treating all projections equally. Viewing a complete such sequence, however, can take a long time and be tiring.

An alternative is provided by the "Projection Pursuit" methodology pioneered by Friedman and Tukey (1974). The objective here is to locate the most "interesting" projections of the data via a computer-aided search. Thus the burden of viewing possibly thousands of different projections of the data is moved from the analyst to the computer.

The central idea in projection pursuit is of "interesting" projections. But what makes a projection "interesting"? Obviously, the terminology is rather vague. There is thus a need to somehow formulate the idea before we can make use of it in a statistical framework. All existing projection pursuit procedures are in fact, based on a single formulation of this idea. We have put down this formulation as algorithm (Algorithm 1 on the following page).

Several instances of constructions on the lines of this algorithm can be found in the literature. They differ principally in the choice of structure S. Depending on how S is interpreted - either as

Algorithm 1 : The Basic Projection Pursuit Algorithm

Step 0: Given a k-dimensional dataset $\mathbf{X}_{n_1 \times k}$.

Step 1: A target dimension $d, d < k$ is chosen in which the data will be projected.

Step 2: A d-dimensional structure S is chosen, by which either interestingness, or its absence will be judged for the projections.

Step 3: A scalar function, the projection index $p(\mathbf{X}, S, \mathbf{A})$ is defined to quantify the extent of structure S in all d-dimensional linear projections $\mathbf{A}_{k \times d}$ of the dataset $\mathbf{X}$.

Step 4: Local maxima of the projection index are located either numerically or analytically.

being of interest or not - we may identify two classes of projection pursuit indices.

The first class of such constructions may be termed the "interestingness indices". In this class, we may place all indices that explicitly seek out a particular structure in projections of the data. The procedure is as follows. A structure S is (pre-)defined to be of interest. For example, Friedman and Tukey (1974) considered a projection exhibiting localized regions of high density interspersed with regions of low density - a kind of clustering - as an interesting structure. Similarly, Posse (1992) and Lee et al. (2005) pursued projections exhibiting class separation, while Eslava and Marriott (1994) and Bolton and Krzanowski (2003) define projections exhibiting clustering structures as of interest. Once the structure of interest is defined, a function - the projection index - is designed to quantify adherence of projected data to this structure S. This index is then maximized to obtain solution projections.

The other class of indices may be termed the "non-interestingness indices". These differ from the above only in the interepretation that is attached to structure S. Rather than interepreting S as a structure that is interesting and needs to be sought out, they interepret it as a structure that is *not* interesting and needs to be avoided. The most prominent model for non-interestingness has been multivariate normality (Huber, 1985; Friedman, 1987; Jones and Sibson, 1987), but see Nason (2001); Naito (1997) for alternatives. The projection index, as before, is designed to quantify adherence of projected data to this structure S. However, in keeping with the interpretation of S, it is now minimized to obtain solution projections.

But is the algorithm above the only possible formulation of the idea of "interesting" projections? This was the question that motivated our research. However, before we go on to describe the alternate construction, let us motivate it by an example.

We recall here the Randu dataset, a singularly impenetrable problem that finds frequent mention in the literature cited above. 1200 numbers are generated consecutively from the congruence, $x_{i+1} = ((2^{16} + 3)x_i) \bmod 2^{31}$. These are arranged in 3 columns, so that the first three numbers form the first row, the next three the second, and so on. The three dimensional data so generated displays a characteristic structure consisting of parallel planes as seen in figure 1. This structure is visible only when viewed at certain angles, that is, only in certain 2-D projections of the data.

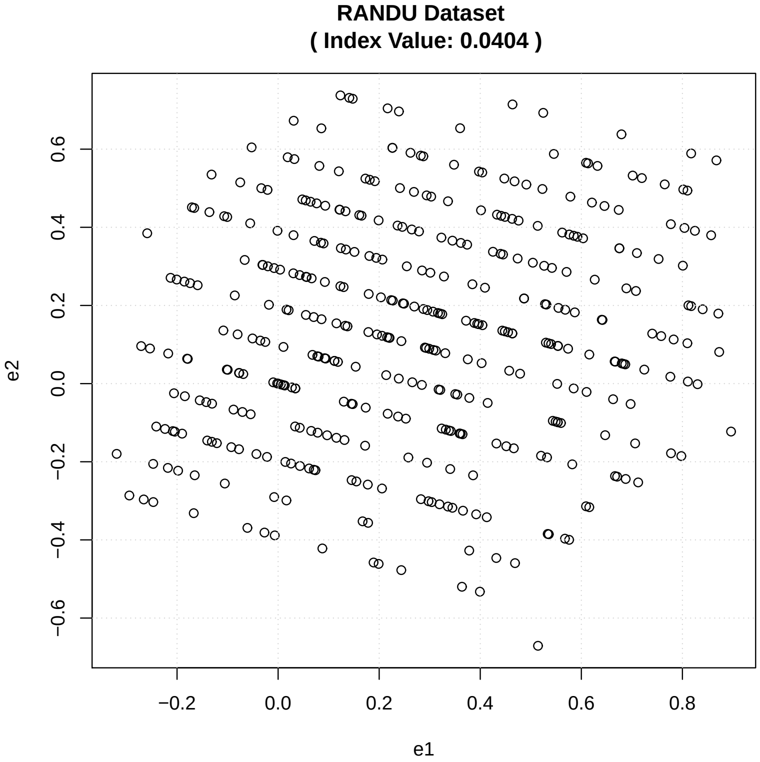

Figure 1: Projection of RANDU dataset showing the characteristic parallel planes structure.

Projections obtained from the principal components miss this structure entirely. Unfortunately, all PP procedures mentioned above fail at this task as well.

The class of "interestingness" PP indices are not even applicable here since while we do possess indices for class separation as well as clustering, no index has as yet been proposed for the unusual structure that we see here.

The performance of the normality based indices is only marginally better. Perisic and Posse

(2005) ran an extensive simulation study of such indices on a variety of structures. The Randu structure was also included. Every one of the pre-existing normality indices considered by them failed in locating this structure, and of the 4 new normality-based indices that they proposed, only 2 succeeded in detecting it. More pointedly, one can see from their Table 3, that the Randu structure is assigned either the lowest index value or the next to lowest value by 9 out of 10 projection indices considered.

Tukey said, “The greatest value of a picture is when it forces us to notice what we never expected to see”. The Randu structure is probably one of the best examples of such a picture. The fact that existing PP procedures cannot locate it easily, seems to us a portentous failure - what else indeed are we missing in our exploratory analyses?

The rest of this paper is organized as follows. In section 2, we take a deeper look at the basic projection pursuit algorithm and how it has been implemented. In particular, we point out two fundamental flaws that affect the basic projection pursuit algorithm. These we call (a) “Straitjacketing” and (b) “Tangling” . These flaws are in fact, closely related since they arise as a consequence of the same basic fact - that there is an infinite variety of configurations that points can take on in multidimensional space. While this may seem obvious, designing an exploratory technique that can account for it is non-trivial. Next, in section 3, we propose an alternate formulation of the idea of interestingness, to address the problems mentioned above, especially that of straitjacketing. In section 4, we design our projection index. The design of the index solves the second problem of “tangling”. In section 5, we treat 3 real datasets with our methodology. Section 6 concludes.

2 What ails projection pursuit?

As we saw in the last section, most PP procedures fail to locate the structure present in the Randu data. Why does this happen?

For the first class of indices, the point of failure is easy to spot. The Randu structure is highly unusual and does not fit any of the usual idioms used to describe multivariate point configurations like clustering or class separation. However these are the only structures for which we possess projection pursuit indices!

For the class of normality based indices, the problem is a little harder to see. The Randu data has almost a uniform distribution within the unit cube, so predictably, most 2-D projections of that data resemble either a square or a rectangle. That is, almost every projection of the data is quite non-normal. The parallel planes projection is thus distinguished among other projections of the data not by the fact of its non-normality, but by virtue of its non-uniformity. This is why the normality indices generally fail to locate this structure.

Having identified these points of failure, the next question is : why do they arise? On close examination, it shall be seen that these failures are symptomatic of weaknesses in the projection pursuit algorithm itself.

First, Step 2 in that algorithm requires the specification of a structure that shall be deemed to be either of interest or devoid of it. Given, however, that we are *exploring* the data, how reasonable is it to place such a demand on the analyst?

The usual way out of this chicken-and-egg problem has been to pre-specify a structure. This itself rasies two issues. One, that the pre-specification is done without reference to the data at hand, and thus takes away from the exploratory flavor of the technique. Two, that only the cases of a clustering structure, class separation and normality have been considered in the literature - a list that is far from exhaustive. The point is not that all structures have not been considered, but that we *can't* consider all of them! There is an infinite variety of configurations that points can take on in multidimensional space - for only a few of which we have verbal or mathematical descriptions beforehand.

This attempt to fit a concept as diffuse as "interestingness" into rigid categories, like class separation or clustering, or equally badly, set it up in opposition to strict parametric forms like normality is what we term "straitjacketing". We can safely conclude that Step 2 of existing algorithm is flawed and impossible to implement in the spirit of data exploration.

Second, is the problem that we term "tangling". To see how it arises, note how in the projection pursuit index of step 3, viz. $p(\mathbf{X}, S, \mathbf{A})$, the structure of interest S enters only as an argument. This suggests the use of a single index regardless of the type of structure chosen. In other words, the projection index $p(.)$ should be able to take as input any structure S and quantify its extent in projections of the data. The suggestion is of obvious practical utility. It saves the analyst the trouble of building a function for each particular structure - a daunting task indeed!

However, implementing this suggestion is also completely impractical in the present scheme of things. We saw how step 2 in the algorithm requires us to specify a structure. If for instance, we specify a clustering structure therein, the projection index, of necessity, must be designed around this choice. This is so, since a function meant to quantify (say) a clustering structure is next to useless if we wish to locate (say) non-normal structures.

But designing projection indices around structures voids the suggestion in Step 3 of the algorithm. Rather, it has the effect of introducing an additional level of dependence between the choice of structure S and the projection index $p(.)$. We can put down this dependence as an intermediate step in the basic algorithm between steps 2 and 3.

Step 2.5 A meta-projection index $P(S)$ is used to choose the projection index $p(\mathbf{X}, S, \mathbf{A})$ on the basis of structure S.

Even though introduction of this step is necessary if the basic algorithm is to be put into practice, it has the effect of tangling up the second and third steps of the projection pursuit algorithm. Its presence implies that one must now define a new projection index for every structure that one is interested in exploring. This is generally beyond the technical skills of most *users* of statistical analysis. Nor indeed is it possible to define the meta-projection index $P(S)$ for every S beforehand - given that we are dealing with the configuration of points in multidimensional space, where an infinite variety of structures is possible!

It is not hard to see that the problems of straitjacketing and tangling are closely related. In fact, each of these arise because a fundamental issue is inadequately addressed in the PP literature. Which is the sheer variety of structures (point configurations) possible in multidimensional space. Being an exploratory technique, aiming to uncover the unexpected, this is serious flaw indeed. What we need is a PP algorithm that can deal with this variety sensibly.

3 Resolving the problems

In this section, we concentrate attention on the problem of “straitjacketing”. This is in fact, the central problem in projection pursuit - to define and formulate the interestingness of multidimensional structures - in as general a way as possible. In other words, the definition of interestingness should be such that it can accommodate the infinite variety of point configurations possible in multidimensional space.

How does one do this? To build our solution, we try to mimic within our formulation what a human analyst does intuitively.

Given projections of a multivariate dataset, a human analyst is more often than not easily able to point out those that seem "interesting" to him. Here the term "interesting" is not being used in the technical sense of projection pursuit, but rather in the English language sense. How can he do it?

This is possible since data and its analysis seldom occur in a vacuum - rather they take place in a certain context. For example for the RANDU dataset, we "know" that the numbers lie uniformly within a cube. This is the context, by which we judge the extreme regularity of the parallel planes structure to be unusual and thus "interesting". Seen this way, we realize that interestingness really arises from the context. This also gives us a satisfactory explanation for why a structure that is considered interesting in one data analysis is not considered to be so in another - it is the context that changes and the change in concepts of interestingness follows. Thus, on a purely philosophical level, one way of defining what is an interesting structure and what is not is suggested if we can somehow make use of the context.

To make this insight practical, we need a formulation of it. This has two parts: (a) Given a context, a way to formulate it and, (b) Given such formulation, a way to find "interesting" projections of the data. We shall suggest a way to do each of these in the following. Note however, that whatever formulations we may propose kick-in only *after* the context has already been identified. Thus the context needs to be supplied by the analyst based upon his understanding of the data analysis exercise. However, in the latter part of this section, we shall (c) suggest ways in which context may be provided.

Let us deal first with (a). Our suggestion is to formulate context via what we call "auxilliary datasets". Thus, these are datasets that embody the context of the data analysis exercise.

For instance, for the Randu data, the information that points lie uniformly within a unit cube is what provides context to its analysis. To obtain an auxilliary dataset for this data means to obtain *another* set of data that reflects this property of the randu data. That is, we need another data that has the same property. Thus we may choose any other 3-dimensional dataset whose points are also uniformly distributed within the unit cube.

Note here that the context formulation did *not* imply or require information about the interesting structures in the data. In fact, at this stage, we do not even know what these structures will turn out to be! In general, specifying an auxilliary dataset in itself does not tell us about the interesting structures about the data. Rather, they are only stand-ins for the state of information possessed by the analyst. This is in contrast to the existing algorithm that requires the specification of interesting structures at the very outset.

Another way to think about out formulation is provided by the comment of Tukey quoted in the introduction. We want to locate structure that is unexpected, so as a first step, we have to formulate that which is expected.

Turning now to (b), we shall for the sake of convenience use the same framework as that of algorithm 1 to express our formulation. Thus our context driven formulation of interestingness has the effect of replacing the second step of that algorithm with,

Step 2(ALT) A k-dimensional dataset $\mathbf{Y}_{n_2 \times k}$ is chosen, which embodies the context of the data analysis by which either interestingness, or its absence will be judged for the projections of the data.

Specification of this step completes one part of (b). The rest of it shall be dealt with in the next section via the projection index.

Finally, we turn our attention to (c), namely how context can be arrived at. The context of a data analysis may be said to emerge from the sum total of one or more of several elements.

The first of these can broadly be termed subject matter knowledge. It was this element that was exploited by us in the randu example. Subject matter knowledge may arise in several other ways as well. For instance, data on all the variables may be available from a previous study. This data is then not only the context, but also the most natural auxilliary dataset to represent it. Another instance is when theory or convention dictates how the data *should* look like. An example would be data resulting from certain standard industrial processes. Similarly, certain physical experiments yield highly multivariate data and the focus is on detecting anomalies. A good idea of what is "normal" may be known from previous experiments performed under conditions where the anomaly is known not to arise. Similar problems come up in the change analysis of images (Radke et al., 2005). In all such cases, the previous study can be said to provide the context by which interestingness of structures may be judged.

Sometimes, certain variables in the the data itself may be used to provide auxiliary datasets. For example, it is common to record class (factor) variables like sex, location, etc. along with the measurement variables. A perennial question in data exploration is the investigation of the effect, if any, that these labels have. One can then partition the data by class and use either partition to provide the auxiliary dataset - the context - to the other. Note that this situation is slightly different from those foregoing - we are now equally interested in the "data" as well as the "auxiliary".

In fact even when these elements - subject matter knowledge and factor variables - are not available - we can still construct an auxilliary dataset. The idea behind them is the following. In projection pursuit, we are interested in exploring multivariate structures. Any and all such structures must necessarily arise from the joint distribution of all variables, and not solely from their marginals. Thus the joint distribution is what provides context to the exercise. Consider then a multivariate dataset each of whose marginal distributions is the same but with a different joint distribution. If such a dataset can be obtained, it can be said to provide some kind of "anti-context" to the data analysis in that it is expected to show no structure present in the original set.

In fact, it is not hard to construct such datasets. Denote the multivariate random variable underlying the data by $\mathbf{X}$ with components $X_i, i = 1, 2, \ldots, d$, each of whose distribution is $F_i, i = 1, 2, \ldots, d$. The distribution of $\mathbf{X}$ can always be written as $\mathcal{C}(F_1, F_2, \ldots, F_d)$ where $\mathcal{C}$ is some copula, representing the dependence structure between the components of $\mathbf{X}$. Under the hypothesis that the data indeed exhibits some geometric structure of a multivariate nature, it must necessarily derive from $\mathcal{C}$. One way then, to explore the geometric structure in $\mathbf{X}$ is to compare it to another random variable $\mathbf{Y}$ that is in some sense closest to $\mathbf{X}$, but yet does not exhibit said structure. We propose to construct such a $\mathbf{Y}$ by replacing the dependence structure implied by $\mathcal{C}$ by some other copula $\mathcal{C}'$. The simplest such $\mathcal{C}'$ is the copula of independence, that is, by obtaining the distribution of $\mathbf{Y}$ as $\mathcal{C}'(F_1 \times F_2 \times \ldots \times F_d)$.

s.

4 Projection Index

In this section, we turn to the previously mentioned problem of "tangling". That is, when the structure S in step 2 of the basic algorithm is specified parametrically, there is no choice but to design the projection index, $p(\mathbf{X}, S, \mathbf{A})$ around this choice. This is a limitation since for each structure S of interest, a new projection index needs to be proposed.

On the other hand, there is obvious utility in a projection index that can deal with any arbitrary structure. Specifying S via a dataset $\mathbf{Y}_{n_2 \times k}$ is meant to allow exactly this. By placing all contextual information into the benchmark dataset, we are able to segregate it from the projection index.

Effectively, we now need to specify the projection index only once. This design allows the notion of what is interesting or not to change from problem to problem, at the user's discretion, without necessarily requiring a change of projection index. Thus subject matter or other inputs can be meaningfully incorporated without being overwhelming for the user.

The required change is in Step 3 of the algorithm.

Step 3(ALT) A scalar function, the projection index $p(\mathbf{X}, \mathbf{Y}, \mathbf{A})$ is defined to quantify the extent of difference / similarity in datasets $\mathbf{X}$ and $\mathbf{Y}$ over their d-dimensional linear projections on $\mathbf{A}$, viz. $\mathbf{XA}$ and $\mathbf{YA}$.

We have chosen to base our projection index on the spatial distribution function, sometimes also called the M-distribution function (Koltchinskii, 1997; Serfling, 2002). This is nothing but the inverse function of the better-known spatial quantiles (Chaudhuri, 1996). Since this statisitc is somewhat unfamiliar, we describe it and projection index based on it briefly. The reason for the unconventional choice are the several advantages that this index has over other test statistics.

4.1 The Index

For a vector-valued random variable $\mathbf{X}$ having an absolutely continuous distribution in $\Re^d$ and $\mathbf{t} \in \Re^d$, the spatial distribution function $\mathbf{G}_{\mathbf{X}}(\mathbf{t})$ is defined as

$$\mathbf{G}_{\mathbf{X}}(\mathbf{t}) = E\frac{\mathbf{X}-\mathbf{t}}{\|\mathbf{X}-\mathbf{t}\|} \tag{1}$$

$\mathbf{G}_{\mathbf{X}}(\mathbf{t})$ can thought of as a generalization of the cumulative distribution function to multidimension. Just like the cumulative distribution function, $\mathbf{G}_{\mathbf{X}}(\mathbf{t})$ also characterizes the distribution of $\mathbf{X}$. However, it has the advantage of being a continuous function of its argument $\mathbf{t}$ even in samples, making it easier to handle.

Test statistics for differences in distribution can be easily constructed using $\mathbf{G}_{\mathbf{X}}(\mathbf{t})$. For the the two-sample case that is of interest, we propose as projection index, use of the function,

$$\int_{\mathbf{t}\in S} ||\mathbf{G}_{\mathbf{X}}(\mathbf{t}) - \mathbf{G}_{\mathbf{Y}}(\mathbf{t})||d\mathbf{t} \tag{2}$$

The estimation of the projection index has been put down as algorithm 2.

Steps 4 and 5 in this algorithm deserve further comment.

In step 4, we need to choose a center $\boldsymbol{\mu}$, and a radius r. For an estimate of $\boldsymbol{\mu}$, we use the spatial median since it fits naturally into the framework, being the point $\mathbf{t}$ at which $E\|\mathbf{X}-\mathbf{t}\|$ is minimized. For the radius parameter r, we took the distance of the farthest data point from $\boldsymbol{\mu}$. Thus, the multiplier k is the only parameter that needs to be set by the user. In our experience, values in the range (0.5,3) work well, and the index is not too sensitive to the choice.

In step 5, theoretically, the choice of points $\mathbf{t}_j$ do not affect the estimated value of the index via equation (5). In practice however, the choice is material, especially if m is small. This is important Whisince if the evaluation points for the integrand are generated randomly, the estimated value of the integral will vary between evaluations over the same plane! In that case, there may arise a certain amount of unpredictability in the optimization.

To remedy this, we chose to generate the evaluation points via Sobol sequences (Dutang and Savicky, 2010), which are quasi-random random numbers. Not only do they provide greater

Algorithm 2 Estimating the Index

Step 0: Given: two k-dimensional samples $\mathbf{X}_{n_1 \times k}$ and $\mathbf{Y}_{n_2 \times k}$.

Step 1: Given a projection matrix $\mathbf{A}_{kxd}$.

Step 2: Obtain the projected samples, $\mathbf{XA}_{n_1 \times d}$ and $\mathbf{YA}_{n_2 \times d}$.

Step 3: At any point $\mathbf{t}_{d \times 1}$, an estimate of $\mathbf{G_{XA}}(\mathbf{t})$ can be obtained as,

$$\hat{\mathbf{G}}_{(\mathbf{XA})}(\mathbf{t}) = \frac{1}{n_1} \sum_{i=1}^{i=n_1} \frac{(\mathbf{XA})_\mathbf{i} - \mathbf{t}}{\|(\mathbf{XA})_\mathbf{i} - \mathbf{t}\|} \tag{3}$$

and likewise for $\mathbf{G}_{(\mathbf{YA})}(\mathbf{t})$.

Step 4: The integral in (2) needs to be computed in a finite region S. Specify S as,

$$S(k) = \{\mathbf{t} : |\boldsymbol{\mu} - \mathbf{t}| < kr\} \tag{4}$$

where $\boldsymbol{\mu}$ is any estimate of location for the two datasets combined, $r > 0$ is a radius parameter and $k > 0$ serves as a multiplier. Thus we compute the integral inside a circle of radius kr with centre at $\boldsymbol{\mu}$.

Step 5: Generate m points, $\mathbf{t}_1, \ldots, \mathbf{t}_m$ inside the region S at which the integrand will be evaluated.

Step 6: Estimate $\mathbf{G}_{(\mathbf{XA})}(\mathbf{t})$ and $\mathbf{G}_{(\mathbf{YA})}(\mathbf{t})$ at these m-points to obtain $\hat{\mathbf{G}}_{(\mathbf{XA})}(\mathbf{t_j})$ and $\hat{\mathbf{G}}_{(\mathbf{XA})}(\mathbf{t_j})$, $j = 1, \ldots, m$.

Step 7: Obtain an m-point monte-carlo approximation of the integral (2) as,

$$\hat{p}(\mathbf{X}, \mathbf{Y}, \mathbf{A}) = \frac{1}{m} \sum_{j=1}^{j=m} (\hat{\mathbf{G}}_{(\mathbf{XA})}(\mathbf{t_j}) - \hat{\mathbf{G}}_{(\mathbf{XA})}(\mathbf{t_j})) \tag{5}$$

accuracy for a fixed number of evaluations, but also provide a systematic way to scale up the accuracy of the approximation via simply increasing the points. Algorithm 3 desribes how to generate points in the unit circle.

Algorithm 3 Generating Evaluation Points

```
Step 0: Required: $m$ points in $d$-dimensional space, $\mathbf{t'}_1, \ldots, \mathbf{t'}_m$
Step 1: Obtain the first m terms $\mathbf{s}_1, \ldots, \mathbf{s}_m$ in the d-dimensional sobol sequence. Each of these is
        an $m$-dimensional point, that is, $\mathbf{s}_j = (u_{j1}, u_{j2}, \ldots, u_{jd})$ where each $u_{jk}$ can be considered
        to be uniformly distributed in (0,1).
Step 2: Assume that each $\mathbf{s}_j$ describes the coordinates of a $d$-dimensional vector in hyperspherical
        coordinates. Convert each of them then to usual Cartesian coordinates. This gives us
        our $m$ evaluation points in the unit circle. Store these.
Step 3: Given a candidate plane $\mathbf{A}$, with associated $\boldsymbol{\mu}$ and $r$, change the origin of points
        $\mathbf{t'}_1, \ldots, \mathbf{t'}_m$ to $\boldsymbol{\mu}$ and scale by $kr$ to obtain the final evaluation points, $\mathbf{t}_1, \ldots, \mathbf{t}_m$.
```

4.2 Advantages of the chosen index

To guide our choice for the projection index $p(\mathbf{X}, \mathbf{Y}, \mathbf{A})$, we looked at the previous PP literature. It is interesting to note that literature has almost always chosen to base projection indices on test statistics for difference in distributions. The reason is not hard to see. In an exploratory analysis, it would be prudent to ensure that the function chosen respond to any and all kinds of differences. As such, an omnibus test is always preferable to tests that concentrate on specific features, like skewness. This factor is even more important to our construction which is intended to be much more general than those foregoing.

Thus, we must choose a test statistic for a two-sample multivariate distributional test to form the basis of our projection index. Even this, however, still leaves us with substantial choice. The best known of such statistics is of course the χ^2. However it is also the one most unsuited to the task, because of the requirement of a cell partition scheme. The power of the statistic depends critically on the particular scheme used; yet as the projections change, the appearance of the point cloud can change dramatically, causing problems for any cell partition scheme.

Other test statistics available in the class are the multivariate generalizations of the Wald

Wolfowitz and Kolmogorov Smirnov statistics suggested by Friedman and Rafsky (1979), the multivariate versions of the Kolmogorov Smirnov (Peacock, 1983; Fasano and Franceschini, 1987; Justel, Peña, and Zamar, 1997) and Cramer von Mises statistics and a recent test proposed by Baringhaus and Franz (Baringhaus and Franz, 2004).

Our projection index is preferable to the other statistics, primarily on grounds of fast computation. Since the projection index is computed repeatedly over hundreds of candidate planes, an index that takes time to compute can slow down the whole process to a crawl. Not only is our projection index faster to compute, but its computation is easily parallelized, since the evaluations in step 6 of algorithm (2) are all independent of each other. We can further speed up the computation by keeping the number of evaluation points low during the early search. At any stage, a more exact value can be calculated, by using a greater number of evaluation points.

The other statistics like those of Friedman and Rafsky (1979) which requires the computation of a minimal spanning tree or the KS-type statistics of (Fasano and Franceschini, 1987; Justel, Peña, and Zamar, 1997) all involve discrete functions. Thus a complete evaluation of the test statistic is required each time. They all also involve some kind of sorting-like operation which are not as easily parallelized. As such, they are generally slower than our index.

Our index possesses two other nice properties as well. The first is rotational invariance. In projection pursuit, we are generally only interested in the configuration of points, while the orientation of axes play only a minor role at best (Morton, 1990). In our case, as long as the rotation is applied equally to the dataset of interest and the auxiliary one, the index value does not change.

Finally, our index is unmatched in how easily it is generalized to any arbitrary dimension. The estimation of the function $\mathbf{G}_{\mathbf{X}}(\mathbf{t})$ poses no challenge at all in any dimension. The Monte-Carlo evaluation of integral poses no theoretical challenges, though with increasing dimension, the number of evaluation points will also need to go up.

4.3 The Optimization

Optimization of the index is how we locate our solution projections. Two choices are available. The more traditional way is to use simulated annealing. Full details of that procedure may be found in Lee et al. (2005).

Alternately, we may exploit the special structure of our problem. We aim to optimize our projection index $p(\mathbf{A})$ over the space $\Re^{n\times d}$ with orthogonality constraint $\mathbf{A}^T\mathbf{A} = I$ and rotations of $\mathbf{A}$ having no effect on the index. A theoretical treatment of optimization under such conditions can be found in Edelman et al. (1998). Buja et al. (2005) explicitly deal with the case of projection pursuit. Functions for the same are available in the `tourr` package (Wickham et al., 2011) from which we adapted them for our own package.

5 Applications

5.1 Randu

The Randu problem was already encountered in the introduction. Let us suppose that the analyst knows that the numbers lie within the unit cube. This forms the context of the analysis. Next, we need to obtain an auxiliary dataset to represent this information. In general, numbers generated from any random number generator have the same geometric property of lying within a cube.

To start with, we chose the Park-Miller MINSTD generator. We generated 1200 numbers from it, and arranged them in the same 400×3 arrangement as the RANDU set. That is, in the auxiliary set, the first three numbers generated form the first row, the next three the second row, and so on. Only one such auxiliary set was generated.

Next, 20 projection matrices are generated randomly, to serve as starting points for the optimization. From each of these, the index was optimized using simulated annealing with 200 iterations being allowed from each start. The behaviour of the optimized index values can be gauged from figure 2 on the following page.

What if we use a different auxiliary dataset? To test this, we generated 1200 numbers from the Mersenne-Twister (MST) random number generator, and arranged them in the same way as above. From the same 20 projection matrices, a search was again initialized. It was noticed that the structure was more likely to be found with a smaller (50-80) iterations than with a larger number of iterations when starting from the same projection. This is probably because the MST generator is so different from RANDU, that the parallel planes projection is not the biggest

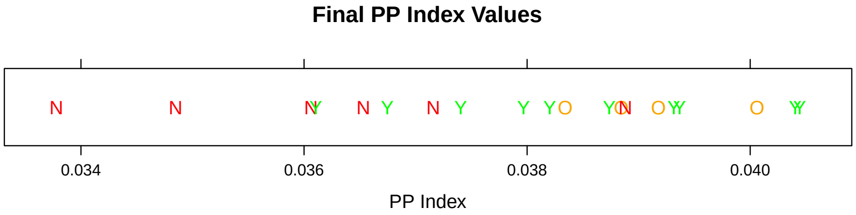

Figure 2: 20 random projections were generated, and from each a simulated annealing was initialized with maximum iterations set to 200. The strip chart shows the final index values achieved. Red points (plot character 'N') indicate solution projections where the parallel planes structures was not visible, while green points ('Y') indicate projections where the structure was visible. Orange points ('O') are those solution projections where the structure was partially visible. The spread of the values is typical of the results obtained when simulated annealing is used. There is very little spread and "good" and "bad" solution projections are mixed together. It is to be noted that these index values are also dependent on the auxiliary dataset chosen, and so are not comparable across such datasets.

difference between the two. With the iteration limit set to 75, 4 of the 20 solution projections, showed the parallel planes structure.

5.2 Colon Cancer

The data relates to colon cancer and first appeared in **?**. There are 2000 columns, each corresponding to a gene, and 62 rows, each corresponding to a tissue sample. 40 of these tissue samples were cancerous, while 22 of them were normal. The measurements themselves, correspond to intensity of expression. The data is in fact quite high dimensional.

In aplying our PP procedure to this data we shall assume that the class label (cancerous / non cancerous) are known. The rest is rather simple. Either of the two groups may be called the data, and the other the auxiliary set. Since the copmarisons are always symmetric, this does not really matter. At any rate, we are interested in both the datasets, and further we shall plot both on the same graph. This is of course unusual, but serves to demostrate the proposed procedure's versatility.

Solution projections found are shown in figures **??** and **??**. Separation between the two kinds of tissue samples is achieved. Further, it appears from some solution projections that there may two groups in the tumour samples, one closer to the normal tissues than the other. This is the kind of insight that a visual analysis can provide over an analytical one. For instance, it seems to have gone unnoticed in the original study.

5.3 Olives Data

This dataset relates to olive oils produced from different regions of Italy and the percentage of 8 fatty acids contained therein. One hopes to identify the geographical origin of the different oils by studying their fatty acid composition. The identification of the geographical origin itself is hierarchical : first by the broad region in Italy (North, South and the island of Sardinia) and next by the particular collection area. Three oils come from the north: East and West Liguria and Umbria, four from the south: Calabria, North and South Apulia and Sicily, and two from the island of Sardinia : the Coast and Inland.

We shall use this dataset to demonstrate the use of an *objective* benchmark, that is, one obtained via breaking the dependence structure in the data. Thus, we do not make use of any external information or the available class labels to obtain the solution projections. To construct our benchmark (of non interestingness), we sample without replacement from each marginal.

Qualitatively, we obtained two kinds of solution projections for this data, with most projections being either similar to one of these or intermediate between them. Representative projections are shown in figures (**??**) and (**??**). Index values ranged between 20 and 30, with a tendency for the higher values to occur for structures similar to (**??**).

In fact there is good reason for these distinct structures to appear in the solution projections. Recall that the geographical identity is hierarchical, with (finer) collection areas embedded within (broader) regions. The projections similar to figure (**??**) are actually picking up on the (broader) region information, while those similar to figure (**??**) reveal the finer structure. Interestingly, while in the former, the fine structure is not so easy to see, in the latter, there is considerable confounding of the northern and southern oils. Further, if one uses the (broader) region information to plot the corresponding points in separate figures, one can see that northern oils and those from the island of Sardinia are well separated within themselves by area of collection, while for the southern oils, only those from the North and South Apulia can be made out distinctly. This is true of both kinds of projections.

It should be noted here that the projection in figure (??) is very similar to that obtained from the first two principal components, while no combination of principal components yields figure (??).

While the (finer) collection area clusters could be identified using the principal components as well, the multiplicity provided by the projection pursuit offers us a way to visually validate the clusters as well. For this, we recommend a three step strategy. First, one obtains several solution projections from the projection pursuit algorithm. Next, these are plotted in linked displays, using for example, the iplots package (?). Finally, brushing of the isolated points in each projection can lead to a rapid and comprehensive understanding of the data structure. The strategy is similar to one used with dynamic graphics, but by use of static displays has the advantage of being more easily understood. In fact, using the strategy, we were able to cut down the misclassified cases dramatically.

Finally, following an isolation strategy, we removed the oils corresponding to the 5 collection areas that emerged as clusters, and ran the programs with this reduced set. Interestingly, none of the two-dimensional projections showed much structure. Incidentally, the same was true of the principal components as well. We thus shifted to three-dimensional projections in the hope of achieving a better structure. Most of these projections showed similar structure with oils from Calabria, East Liguria and North Apulia separating out to a greater or lesser extent in different projections, while the oils from Sicily did not localize in any projection.

6 DISCUSSION

Methods for high dimensional data are per force highly computational in nature. While their computational nature is precisely what makes the methods feasible, there is also always the black box risk : that the algorithm used is so inflexible that there is no way for the user to guide the process or interact with the output. For exploratory methods this risk is even greater, simply because the demands for user feedback and control are so much higher. While no method can currently provide the same level of user input and interaction that are so naturally and easily incorporated in the low dimensional visualizations like scatterplots, it is still a goal worth aspiring for. On the other hand, there is also always the risk of excess. For instance, it is not uncommon for users to feel disoriented and confused when presented with dynamic graphics like data tours (?) that are largely user-driven.

Exploratory projection pursuit has the potential to strike a balance between these two extremes: while it is highly computational, the output, which is usually a scatterplot is invariably easily understood. Unfortunately, however, the traditional algorithm for the technique is a near-perfect black box, in that it offers no scope for interaction with either the algorithm or its output. In this regard, guided tours (**?**) offer one relaxation, by way of allowing the user to interact with the output. However, the more important aspect regarding selection of solution projections has so far, remained impenetrable for the average user. The proposed algorithm thus represents an attempt at increasing the level of user feedback with the whole process, in the hope that it will lead to more relevant and meaningful results from the analysis.

The current technique has its closest comparison in the clustering (Eslava and Marriott, 1994; Bolton and Krzanowski, 2003) and discrimination indices (Lee et al., 2005). While these are highly specialized indices, ours arises from a more generalized perspective rooted in data exploration. One area of future investigation is the loss of power that this entails for the present technique. Judging from its performance on the datasets considered here, it does not appear to very great, however more comparisons, especially on simulated datasets are necessary.

SUPPLEMENTAL MATERIALS

R package cepp for Context Driven Exploratory Projection Pursuit

The package contains code to perform the methods described in this article. It also contains all datasets used as examples here. (Downloadable from CRAN)

References

D. Asimov. The grand tour: a tool for viewing multidimensional data. *SIAM Journal on Scientific and Statistical Computing*, 6(1):128–143, 1985.

L. Baringhaus and C. Franz. On a new multivariate two-sample test. *Journal of Multivariate Analysis*, 88(1):190–206, 2004.

R.J Bolton and W.J Krzanowski. Projection pursuit clustering for exploratory data analysis. *Journal of Computational and Graphical Statistics*, 12(1):121–142, 2003.

A. Buja, D. Cook, D. Asimov, and C. Hurley. Computational methods for high-dimensional rotations in data visualization. In *Data Mining and Data Visualization*, volume 24, page 391. 2005.

P. Chaudhuri. On a geometric notion of quantiles for multivariate data. *Journal of the American Statistical Association*, 91(434): 862–872, 1996.

Christophe Dutang and Petr Savicky. *randtoolbox: Generating and Testing Random Numbers*, 2010. R package version 1.10.

A. Edelman, T.M. Arias, and S.T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM journal on Matrix Analysis and Applications*, 20(2):303–353, 1998.

G. Eslava and F.H.C. Marriott. Some criteria for projection pursuit. *Statistics and Computing*, 4(1):13–20, 1994.

G. Fasano and A. Franceschini. A multidimensional version of the Kolmogorov-Smirnov test. *Monthly Notices of the Royal Astronomical Society*, 225:155–170, 1987.

J.H. Friedman. Exploratory projection pursuit. *Journal of the American statistical association*, 82(397):249–266, 1987.

J.H. Friedman and L.C. Rafsky. Multivariate generalizations of the Wald-Wolfowitz and Smirnov two-sample tests. *The Annals of Statistics*, pages 697–717, 1979.

J.H. Friedman and J.W. Tukey. A projection pursuit algorithm for exploratory data analysis. *IEEE Transactions on Computers*, 100(9):881–890, 1974.

P.J. Huber. Projection pursuit. *The Annals of Statistics*, 13(2): 435–475, 1985.

MC Jones and R. Sibson. What is projection pursuit? *Journal of the Royal Statistical Society. Series A (General)*, 150(1):1–37, 1987.

A. Justel, D. Peña, and R. Zamar. A multivariate Kolmogorov-Smirnov test of goodness of fit. *Statistics & Probability Letters*, 35(3):251–259, 1997.

VI Koltchinskii. M-estimation, convexity and quantiles. *The Annals of Statistics*, 25(2):435–477, 1997.

E.K. Lee, D. Cook, S. Klinke, and T. Lumley. Projection pursuit for exploratory supervised classification. *Journal of Computational and Graphical Statistics*, 14(4):831–846, 2005.

S.C. Morton. *Interpretable Projection Pursuit*. PhD thesis, Stanford University, 1990. URL www.slac.stanford.edu/cgi-wrap/getdoc/slac-r-355.pdf.

K Naito. A generalized projection pursuit procedure and its significance level. *Hiroshima Mathematical Journal*, 27(3):513–554, 1997.

G.P. Nason. Robust projection indices. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 63:551–567, 2001.

J.A. Peacock. Two-dimensional goodness-of-fit testing in astronomy. *Monthly Notices of the Royal Astronomical Society*, 202: 615–627, 1983.

I. Perisic and C. Posse. Projection pursuit indices based on the empirical distribution function. *Journal of Computational and Graphical Statistics*, 14(3):700–715, 2005.

CG Posse. Projection pursuit discriminant analysis for two groups. *Communications in Statistics - Theory and Methods*, 21(1):1–19, 1992.

R.J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam. Image change detection algorithms : A systematic survey. *Image Processing, IEEE Transactions on*, 14(3):294–307, 2005.

R. Serfling. Quantile functions for multivariate analysis : Approaches and applications. *Statistica Neerlandica*, 56(2):214–232, 2002.

Hadley Wickham, Dianne Cook, Heike Hofman, and Andreas Buja. tourr: An R package for exploring multivariate data with projections. *Journal of Statistical Software*, 40(2):1–18, 2011. URL http://www.jstatsoft.org/v40/i02/.