Clipped Projections for More Informative Visualizations [A Work-in-Progress Report]

Bo Kang, Junning Deng, Jefrey Lijffijt, Tijl De Bie
Department of Electronics and Information Systems, IDLab
Ghent University — Imec
firstname.lastname@ugent.be

ABSTRACT

Imagine data that contains a dense core with points scattered around away from the core. An example would be data with outliers. A figure with a full view of a 2-dimensional projection of data then typically shows the points in the core all close to each other. Due to the fact that the visualization medium as well as our eyes have finite resolution, it may then not be possible to discern the location of the points in the core. In that case it may be more interesting to show a zoomed-in visualization that allows one to explore the structure of the core, while providing only limited information about points that are not part of the core. A trade-off emerges between showing small and large-scale structure, parametrized by the size of a bounding box. The quantification of this trade-off using Information Theory, and the concurrent optimization of the size of a bounding box and finding informative linear projections are the topics of this paper.

CCS CONCEPTS

•Mathematics of computing → Exploratory data analysis; Dimensionality reduction; Information theory;

KEYWORDS

Exploratory Data Mining, Dimensionality Reduction, Information Theory, Subjective Interestingness

ACM Reference format:
Bo Kang, Junning Deng, Jefrey Lijffijt, Tijl De Bie. 2017. Clipped Projections for More Informative Visualizations
[A Work-in-Progress Report]. In *Proceedings of KDD 2017 Workshop on Interactive Data Exploration and Analytics (IDEA'17), Halifax, Nova Scotia, Canada, August 14th, 2017 (IDEA @ KDD'17),* 8 pages.
DOI:

1 INTRODUCTION

Assume a data analyst wishes to glean insights into a 2-dimensional data set by visualizing it. For real-valued data, the most straightforward technique for doing this would be by means of a scatter plot. However, when the data consists of a dense core, with additionally a number of points scattered some distance around it, the dense core tends to show up as a blob of partially occluding points; see Figure 1(a) for an example.

IDEA @ KDD'17, Halifax, Nova Scotia, Canada

DOI:

The result of showing the far-away points is thus that the points in the core become less discernible from each other (whether due to limitations in the plot resolution or in human perception). The amount of information provided about them is effectively reduced by having to zoom out, while the resolution remains constant.

Clipped scatter plots. In this paper we propose the notion of a *clipped scatter plot*. Informally speaking, it can be obtained by overlaying a bounding box on the scatter plot (red box in Figure 1a), and clipping all points outside of this bounding box to the nearest point on the boundary (as indicated by the blue dashed lines in Figure 1a). After doing this, one can zoom in to ensure the bounding box fills the plotting area. The clipped points are then shown with a different marker to distinguish them from points that appear near the boundary, but are within the bounding box area (Figure 1b).

What a user learns from a clipped scatter plot is this: For an unclipped point, the user knows its location up to the resolution (of the displayed plot or human perception, whichever is worse). For a point clipped along a certain dimension, the user only learns that it is further away from the origin than the size of the bounding box along that dimension. Informally, the user learns that such points are 'far away' (outside of the bounding box) in some direction, but precisely how far is not revealed.

By varying the size of the bounding box and corresponding zoom level, clipped scatter plots thus allow one to trade-off detail about the points with small norms, with detail about the points with large norms. In this paper, we discuss how to formalize and optimize this trade-off in a rigorous manner, relying on Information Theory. As an illustration, the bounding box used in Figure 1 is optimal with respect to this measure.

Clipped projections. Data is usually high-dimensional, and in order to visualize it in a clipped scatter plot, its dimensionality needs to be reduced to 2, for example by means of a projection. We will call a clipped scatter plot of a projection of the data a *clipped projection*. To most efficiently inform the data analyst about the data, one can then search for the most informative clipped projection. This amounts to a simultaneous optimization problem over all possible 2-dimensional projections[1] and all possible bounding box sizes (for both dimensions).

We note that clipped projections are distinct from projections of a data set after outlier removal, for several reasons. First, a point that is outlying along one dimension may not be so along another one, such that it may be clipped in one clipped projection but not in another. It may also be clipped along just one dimension in the clipped projection of the data. Second, determining which points are outliers and which are not is often a hard call (Figure 1 is misleading in this respect). Our approach does not require one to make

[1] In fact, our work trivially extends to d-dimensional projections for arbitrary d.

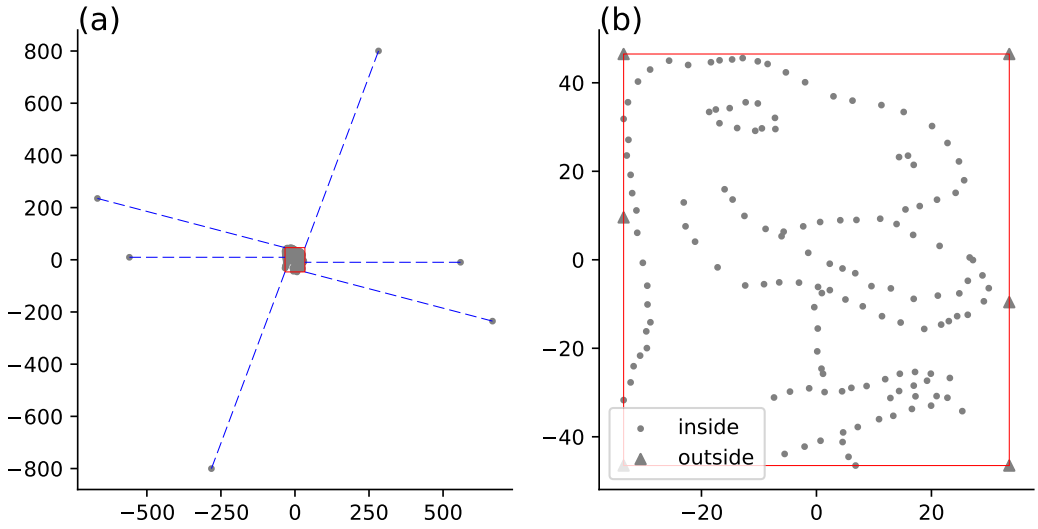

Figure 1: A scatter plot of a 2-dimensional data set (a) is not necessarily effective in revealing information about the data. Indeed, the points with large norm necessitate zooming out to such an extent that the detail in the core set of points is lost to the eye. By clipping points far away from the center to the boundaries of the red bounding box (blue dashed lines show where these points are clipped to) and zooming in as much as possible, a *clipped scatter plot* (b) is obtained. Here, we can discern the dinosaur hidden in the data. (The data consists of six artificial outliers plus Alberto Cairo's figure: http://www.thefunctionalart.com/2016/08/download-datasaurus-never-trust-summary.html.)

that call—our focus is merely on conveying as much information as possible about the entire data set through an intuitive visualization. Third, we never actually remove any points from the visualization: we merely convey less information about those outside the bounding box.

Contributions. In this work-in-progress paper, we report on early results on the following aspects.

- We introduce the notion of a *clipped scatter plot* and *clipped projection* (Section 2.2).
- We quantify the amount of information a clipped projection conveys about the data (Section 2.3). This quantification is parameterized both by the projection and the size of the bounding box.
- We introduce an algorithm that aims to optimize the information content, searching for the most informative clipped projection of a given data set (Section 3).
- We include some experiments that empirically analyse the ability of the algorithm to find the most informative clipped projection, the scalability of the algorithm, and the usefulness of the approach (Section 4).

2 CLIPPED PROJECTIONS AND THEIR INFORMATION CONTENT

In this section, we provide the formal definition of clipped projections and then discuss how to quantify their informativeness. First, we have to introduce some notation.

2.1 Notation

We use upper case bold face letters to denote matrices, lower case bold face letters for vectors, and normal lower case letters for scalars. We denote a d-dimensional real-valued dataset as $\hat{\mathbf{X}} \triangleq (\hat{\mathbf{x}}'_1, \hat{\mathbf{x}}'_2, \ldots, \hat{\mathbf{x}}'_n)' \in \mathbb{R}^{n \times d}$, and the corresponding random variable as $\mathbf{X}$. We will refer to $\mathbb{R}^{n \times d}$, the space the data is known to belong to, as the *data space*. Dimensionality reduction methods search weight vectors $\mathbf{w} \in \mathbb{R}^d$ of unit norm (i.e. $\mathbf{w}'\mathbf{w} = 1$) onto which the data is projected by computing $\hat{\mathbf{X}}\mathbf{w}$. If k vectors are sought, they will be stored as columns of a matrix $\mathbf{W} \in \mathbb{R}^{d \times k}$ where $\mathbf{W}'\mathbf{W} = \mathbf{I}$. We will denote the projections of a data set $\hat{\mathbf{X}}$ onto the column vectors of $\mathbf{W}$ as $\hat{\mathbf{\Pi}}_{\mathbf{W}} \in \mathbb{R}^{n \times k}$, or formally: $\hat{\mathbf{\Pi}}_{\mathbf{W}} \triangleq \hat{\mathbf{X}}\mathbf{W}$, and analogously for the random variable counterpart $\mathbf{\Pi}_{\mathbf{W}} \triangleq \mathbf{X}\mathbf{W}$. We will write $\mathbf{I}$ to denote the identity matrix of appropriate dimensions, and $\mathbf{1}_{n \times d}$ (or $\mathbf{1}$ for short if the dimensions are clear from the context) to denote a n-by-d matrix with all elements $\mathbf{1}_{ij} = 1$. We define $\mathbf{A}_{n \times m} \in [\mathbf{B}_{n \times m}, \mathbf{C}_{n \times m}]$ as $a_{i,j} \in [b_{i,j}, c_{i,j}]$ for every $c_{i,j} \geq b_{i,j}$, $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$.

2.2 Clipped projections

We define a *bounding box* to be a centered k dimensional hyper-rectangular region with range $(-\mathbf{c}, \mathbf{c})$ where $\mathbf{c} \in \mathbb{R}^k_+$. Given a bounding box defined by $\mathbf{c}$ and a projection $\mathbf{z}_i = \mathbf{W}'\mathbf{x}_i$ of a data point $\mathbf{x}_i$, its j-th coordinate either:

- falls outside of $(-\mathbf{c}_j, \mathbf{c}_j)$. Then position of the point is specified within range $\mathbf{z}_{ij} \in [\mathbf{c}_j, \infty)$ (or $\mathbf{z}_{ij} \in (-\infty, -\mathbf{c}_j]$).
- falls in $(-\mathbf{c}_j, \mathbf{c}_j)$. The position of the point is specified only up to a *pixel* of size $f \cdot 2\mathbf{c}_j$, i.e., $\mathbf{z}_{ij} \in [\mathbf{z}_{ij} - f\mathbf{c}_j, \mathbf{z}_{ij} + f\mathbf{c}_j]$, where f is the resolution parameter.

In order to concisely define a projection with respect to the bounding box, we need to introduce a few concepts. Firstly, we define a mapping function that corresponds to the clipping procedure: $t(\mathbf{z}_{ij}, \mathbf{c}_j) = \max(-\mathbf{c}_j, \min(\mathbf{c}_j, \mathbf{z}_{ij}))$. Now, we can express the

lower and upper boundaries of the location of $\mathbf{z}_{ij}$, as conveyed by the clipped scatter plot:

$$\mathbf{l}_j(\mathbf{c}_j, t(\mathbf{z}_{ij}, \mathbf{c}_j)) = \begin{cases} -\infty & : t(\mathbf{z}_{ij}, \mathbf{c}_j) = -\mathbf{c}_j, \\ \mathbf{z}_{ij} - f\mathbf{c}_j & : -\mathbf{c}_j < t(\mathbf{z}_{ij}, \mathbf{c}_j) < \mathbf{c}_j, \\ \mathbf{c}_j & : t(\mathbf{z}_{ij}, \mathbf{c}_j) = \mathbf{c}_j. \end{cases}$$

$$\mathbf{u}_j(\mathbf{c}_j, t(\mathbf{z}_{ij}, \mathbf{c}_j)) = \begin{cases} -\mathbf{c}_j & : t(\mathbf{z}_{ij}, \mathbf{c}_j) = -\mathbf{c}_j, \\ \mathbf{z}_{ij} + f\mathbf{c}_j & : -\mathbf{c}_j < t(\mathbf{z}_{ij}, \mathbf{c}_j) < \mathbf{c}_j, \\ +\infty & : t(\mathbf{z}_{ij}, \mathbf{c}_j) = \mathbf{c}_j. \end{cases}$$

Collectively, we define matrix $\mathbf{L}(\hat{\Pi}_\mathbf{W}, \mathbf{c})$ as a $n \times k$ matrix where each entry $\mathbf{L}(\hat{\Pi}_\mathbf{W}, \mathbf{c})_{ij}$ is the lower boundary of j-th coordinate of the projection of i-th data point. That is,

$$\mathbf{L}\left(\hat{\Pi}_\mathbf{W}, \mathbf{c}\right)_{ij} = \mathbf{l}_j(c_j, t(\hat{\Pi}_\mathbf{W}^{(ij)}, \mathbf{c}_j)). \tag{1}$$

Similarly, $\mathbf{U}\left(\hat{\Pi}_\mathbf{W}, \mathbf{c}\right)$ is the $n \times k$ matrix where each entry

$$\mathbf{U}\left(\hat{\Pi}_\mathbf{W}, \mathbf{c}\right)_{ij} = \mathbf{u}_j(c_j, t(\hat{\Pi}_\mathbf{W}^{(ij)}, \mathbf{c}_j)). \tag{2}$$

Finally, we can define the syntax of a *clipped projection* as:

$$\hat{\mathbf{X}}\mathbf{W} \in \left[\mathbf{L}\left(\hat{\Pi}_\mathbf{W}, \mathbf{c}\right), \mathbf{U}\left(\hat{\Pi}_\mathbf{W}, \mathbf{c}\right)\right]. \tag{3}$$

2.3 Information content of clipped projections

Prior belief model. Our aim is to quantify the information content of a clipped projection. Just like statistics are computed in comparison with a null model, in order to compute the information content of information, we need to specify a background model. Ideally, such a background model would reflect the actual prior knowledge of the user, such that the information content is an appropriate measure for the amount of information the visualization provides to the specific user [2, 3].

We adopt the same approach: we encode these prior beliefs in a probability density $p_\mathbf{X}$ over the data space $\mathbb{R}^{n\times d}$. The probability it assigns to any measurable subset of $\mathbb{R}^{n\times d}$ corresponds to the probability that the data $\hat{\mathbf{X}}$ would belong to that subset under the prior belief. The density function $p_\mathbf{X}$ can typically not be specified directly and instead we infer it from a given set of prior beliefs, as the Maximum Entropy distribution subject to those beliefs. As such, the notion of interestingness here is *subjective*, as the ranking of patterns depends on the belief state of the user.

In this paper, we assume the user has prior beliefs about the scale of a dataset.[2] The user might believe that the average scale of the data points, quantified by their squared norms, is some constant denoted as $\sigma^2 d$ and have no other knowledge. This can be encoded as a constraint on the second moment of the distribution $p_\mathbf{X}$:

$$\mathbb{E}_{\mathbf{X}\sim p_\mathbf{X}}\left[\mathrm{Tr}(\mathbf{X}'\mathbf{X})\right] = \sigma^2 \cdot nd. \tag{4}$$

The MaxEnt distribution subject to this constraint is well known and equal to a product distribution of multivariate Normal distributions $\mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$, i.e.,

$$p_\mathbf{X}(\mathbf{X}) = \frac{1}{\sqrt{(2\pi\sigma^2)^{nd}}} \exp(-\frac{1}{2\sigma^2}\mathrm{Tr}[\mathbf{X}'\mathbf{X}]). \tag{5}$$

[2]There may be many other prior beliefs a user may have. Different prior belief types may result in background distributions of different types (See e.g., [4]). As the goal of this paper is to demonstrate the idea of the clipped projections, we leave the investigation of different prior beliefs as future work.

Given a projection matrix, the marginal distribution $p_{\Pi_\mathbf{W}}$ for projection $\mathbf{\Pi}_\mathbf{W} = \mathbf{X}\mathbf{W}$ then reads:

$$p_{\Pi_\mathbf{W}}(\mathbf{X}\mathbf{W}) = \frac{1}{\sqrt{(2\pi\sigma^2)^{nk}}} \exp(-\frac{1}{2\sigma^2}\mathrm{Tr}[\mathbf{W}'\mathbf{X}'\mathbf{X}\mathbf{W}]). \tag{6}$$

Probability of a clipped projection. If the projection $\mathbf{z}_i$ of a point $\mathbf{x}_i$ falls on the inside of the bounding box in j-th dimension, for small f its probability is well approximated by:

$$\begin{aligned}\mathbf{Pr}_{\Pi_\mathbf{W}}(\mathbf{z}_{ij} \in [\hat{\mathbf{z}}_{ij} - f\mathbf{c}_j, \hat{\mathbf{z}}_{ij} + f\mathbf{c}_j]) &= \int_{\hat{\mathbf{z}}_{ij}-f\mathbf{c}_j}^{\hat{\mathbf{z}}_{ij}+f\mathbf{c}_j} p_{\Pi_\mathbf{W}}(\mathbf{z}_{ij})d\mathbf{z}_{ij} \\ &\approx \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{\hat{\mathbf{z}}_{ij}^2}{2\sigma^2}) \cdot 2f\mathbf{c}_j.\end{aligned} \tag{7}$$

A point $\mathbf{z}$ that falls outside the bounding box on the j-th dimension has probability

$$\begin{aligned}\mathbf{Pr}_{\Pi_\mathbf{W}}(\mathbf{z}_{ij} \in [\mathbf{c}_j, +\infty)) &= \mathbf{Pr}_{\Pi_\mathbf{W}}(\mathbf{z}_{ij} \in (-\infty, -\mathbf{c}_j]) \\ &= \int_{c_j}^{+\infty} \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{t^2}{2\sigma^2})dt.\end{aligned} \tag{8}$$

The first equality follows the symmetricity of the bounding box.

Now, the probability of a clipped projection can be written as,

$$\begin{aligned}&\mathbf{Pr}(\hat{\Pi}_\mathbf{W} \in \left[\mathbf{L}\left(\hat{\Pi}_\mathbf{W}, \mathbf{c}\right), \mathbf{U}\left(\hat{\Pi}_\mathbf{W}, \mathbf{c}\right)\right]) \\ &= \int_{\Pi_\mathbf{W} \in \left[\mathbf{L}\left(\hat{\Pi}_\mathbf{W}, \mathbf{c}\right), \mathbf{U}\left(\hat{\Pi}_\mathbf{W}, \mathbf{c}\right)\right]} p_{\Pi_\mathbf{W}}(\Pi_\mathbf{W})d\Pi_\mathbf{W} \\ &= \prod_{i=1,2,\ldots,n} \left\{ \int_{\mathbf{l}_1(c_1, t(\mathbf{x}_i\mathbf{W}_1, c_1))}^{\mathbf{u}_1(c_1, t(\mathbf{x}_i\mathbf{W}_1, c_1))} \cdots \int_{\mathbf{l}_k(c_k, t(\mathbf{x}_i\mathbf{W}_k, c_k))}^{\mathbf{u}_k(c_k, t(\mathbf{x}_i\mathbf{W}_k, c_k))} p_{\Pi_\mathbf{W}}(\mathbf{z}_i) \right. \\ &\left. d\mathbf{z}_1 d\mathbf{z}_2 \ldots d\mathbf{z}_k \right\}.\end{aligned} \tag{9}$$

The information content. Relying on the background distribution (Eq. 9), we can now quantify the information content (IC) of a clipped projection as the negative log probability of the projection under the distribution. Denote the index set of the projection of points on j-th dimension fall into $(-\mathbf{c}_j, \mathbf{c}_j)$ as $\mathbf{I}_j$, then the number of points falls outside of the bounding box is $n - |\mathbf{I}_j|$. Formally, we have the information content:

$$\begin{aligned}\mathrm{IC}(\mathbf{W}, \hat{\Pi}_\mathbf{W}, \mathbf{c}) &= -\log \Pr\left(\hat{\Pi}_\mathbf{W} \in \left[\mathbf{L}\left(\hat{\Pi}_\mathbf{W}, \mathbf{c}\right), \mathbf{U}\left(\hat{\Pi}_\mathbf{W}, \mathbf{c}\right)\right]\right) \\ &= -\log\left[\prod_{j=1}^{k}\left(\prod_{i\in\mathbf{I}_j} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\mathbf{z}_{ij}^2}{2\sigma^2}\right) \cdot 2f\mathbf{c}_j \right.\right. \\ &\left.\left. \cdot \prod_{i\notin\mathbf{I}_j} \int_{\mathbf{c}_j}^{+\infty} \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{t^2}{2\sigma^2})dt\right)\right]\end{aligned} \tag{10}$$

Now our goal of finding the most informative clipped projection can be formalized as an optimization problem:

$$\begin{aligned}\underset{\mathbf{W},\mathbf{c}}{\text{argmax}}\ & \mathrm{IC}(\mathbf{W}, \hat{\Pi}_\mathbf{W}, \mathbf{c}) \\ \text{s.t.}\ & \mathbf{W}'\mathbf{W} = \mathbf{I} \\ & \mathbf{c} > \mathbf{0}.\end{aligned} \tag{11}$$

In general, the solution of problem (11) corresponds to clipped projections that include as much information as possible complementary to the prior beliefs. Notice that given the prior belief stated above, if we ignore clipping—i.e., if the optimal clipped projection includes all points inside the bounding box—, the optimal solution is equivalent to PCA [3]. However, with clipping also the optimal $\mathbf{W}$ is typically different. In the next section, we analyze problem (11) and propose an approximation scheme to approach it.

3 FINDING THE MOST INFORMATIVE CLIPPED PROJECTION

Solving the optimization problem (11) requires to evaluate the objective function efficiently. However, the integral function (tail probability of normal distribution) in Equation (10) does not have a closed form of elementary functions hence can only be computed approximately. Moreover, note that for different projection matrix $\mathbf{W}$, the positions of points in the projection are different, hence the optimal $\mathbf{c}$ may change. This means the optimal bounding box (half size $\mathbf{c}$) and the number of points falling into the bounding box (i.e., $|\mathbf{I}|$) both depend on the projection matrix $\mathbf{W}$. Such dependency makes the optimization problem difficult to solve.

In this section, we first approximate the tail probability of a normal distribution by an upper bound and obtain an objective function consisting only elementary functions. We then propose an efficient optimization strategy that relies on automatic differentiation and gradient manifold optimization.

3.1 Bounding the tail probabilities

To obtain a closed form representation of (Eq. 10) with elementary functions we approximate the tail probability of a normal distribution as follows[3]:

$$\begin{aligned}\int_{\mathbf{c}_j}^{+\infty} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-t^2/(2\sigma^2)} dt &\le \int_{\mathbf{c}_j}^{+\infty} \frac{t}{\mathbf{c}_j} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-t^2/(2\sigma^2)} dt \\ &= \frac{\sigma e^{-\mathbf{c}_j^2/(2\sigma^2)}}{\mathbf{c}_j\sqrt{2\pi}}.\end{aligned} \tag{12}$$

The first inequality follows the fact that $\frac{t}{\mathbf{c}_j} \ge 1$.

Now the objective function in (Eq. 10) can be re-written as:

$$\begin{aligned}&\mathrm{IC}(\mathbf{W}, \hat{\mathbf{I}}_{\mathbf{W}}, \mathbf{c}) \\ &\approx -\log\left[\prod_{j=1}^{k}\left(\prod_{i\in\mathbf{I}_j} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(\mathbf{z}_{ij}^2)/(2\sigma^2)} \cdot 2f\mathbf{c}_j \cdot \prod_{i\notin\mathbf{I}_j} \frac{\sigma}{\mathbf{c}_j} \frac{e^{-\mathbf{c}_j^2/(2\sigma^2)}}{\sqrt{2\pi}}\right)\right] \\ &= \sum_{j=1}^{k}\left[\sum_{i\in\mathbf{I}_j} \frac{\mathbf{z}_{ij}^2}{2\sigma^2} + (n-|\mathbf{I}_j|)\frac{\mathbf{c}_j^2}{2\sigma^2} + (n-2|\mathbf{I}_j|)\log(\mathbf{c}_j)\right. \\ &\quad \left. |\mathbf{I}_j|\frac{1}{2}\log(2\pi\sigma^2) - |\mathbf{I}_j|\log(2f) + (n-|\mathbf{I}_j|)\log(\frac{\sqrt{2\pi}}{\sigma})\right].\end{aligned} \tag{13}$$

Notice the parameter $\mathbf{c}$ and $\mathbf{I}$ both depend on $\mathbf{W}$. That is, for every $\mathbf{W}$ we need to search for an $\mathbf{c}$ that maximizes the IC. Once $\mathbf{c}$ is computed, then the point sets within bounding box $\mathbf{I}$ with respect to each dimension are determined.

[3] A more detailed discussion can be found at https://mikespivey.wordpress.com/2011/10/21/normaltails/

3.2 Optimization strategy

The dependency of parameters $\mathbf{c}$ and $\mathbf{I}$ on $\mathbf{W}$ as well as the constraint $\mathbf{W}'\mathbf{W} = \mathbf{I}$ make objective (Eq. 13) difficult to be optimized simultaneously over all three parameters. Nevertheless, we propose a gradient method to perform the simultaneous optimization. Observe the orthonormality constraint posed on $\mathbf{W}$ leads to problem (13) being a *Stiefel* manifold optimization problem[4]. This can be addressed fairly efficiently with a standard toolbox. We use the *pyManopt* toolbox [5] to obtain an approximate solution.

In order to apply gradient based solver in *pyManopt*, we need to further compute the gradient of problem (13) with respect to variable $\mathbf{W}$. By encoding the objective function using *TensorFlow*, pyManopt can use TensorFlow's to calculate the gradient automatically.

The remaining question is: how to encode the step of searching optimal $\mathbf{c}$ (hence $\mathbf{I}$) into an objective function which then can be efficiently evaluated in a single step? The answer relies on the observation that for a specific $\mathbf{W}$ we only need to evaluate for each dimension $O(n)$ number of c (hence $O(kn)$ in total) to find the optimal $\mathbf{c}$. Without losing generality, we formally state the observation for j-th dimension as:

Proposition 1. *The optimal $\mathbf{c}_j^*$ that maximizes objective function (Eq. 13) coincides with the j-th absolute coordinate value of some projected point $\mathbf{z}_i$, namely, $\mathbf{c}_j^* = |\mathbf{z}_{ij}|$ for some $i = 1, \ldots, n$.*

Proof. To prove the proposition, it is equivalent to show that the objective function between the two neighboring coordinates (i.e., $\mathbf{c}_j \in [|\mathbf{z}_{m,j}|, |\mathbf{z}_{m+1,j}|]$) tends to be either monotonically increasing or convex.

The monotonic increase case can be easily identified by computing the first derivative of $\mathrm{IC}(\mathbf{c})$ with respect to $\mathbf{c}_j$, which is

$$\begin{aligned}\frac{d}{d\mathbf{c}_j}\mathrm{IC}(\mathbf{c}) &= \frac{n-|\mathbf{I}_j|}{\sigma^2}\mathbf{c}_j + \frac{n-2|\mathbf{I}_j|}{\mathbf{c}_j} \\ &= \frac{(n-|\mathbf{I}_j|)\mathbf{c}_j^2 + (n-2|\mathbf{I}_j|)\sigma^2}{\mathbf{c}_j\sigma^2}.\end{aligned} \tag{14}$$

When $n - 2|\mathbf{I}_j| > 0$, since $(n - |\mathbf{I}_j|) > 0$, it is straightforward that $\frac{d}{d\mathbf{c}_j}\mathrm{IC}(\mathbf{c}) \ge 0$. This implies that $\mathrm{IC}(\mathbf{c})$ monotonically increases as $\mathbf{c}_j$ increases, and the local maximum occurs at the right boundary, i.e., $|\mathbf{z}_{m+1,j}|$.

The other case is $n - 2|\mathbf{I}_j| \le 0$. To show this gives a convex piecewise $\mathrm{IC}(\mathbf{c})$. Let us look at the second derivative of $\mathrm{IC}(\mathbf{c})$ w.r.t c

$$\begin{aligned}\frac{d^2}{d\mathbf{c}_j^2}\mathrm{IC}(\mathbf{c}) &= \frac{n-|\mathbf{I}_j|}{\sigma^2} + \frac{2|\mathbf{I}_j|-n}{\mathbf{c}_j^2} \\ &= \frac{(n-|\mathbf{I}_j|)\mathbf{c}_j^2 + (2|\mathbf{I}_j|-n)\sigma^2}{\mathbf{c}_j^2\sigma^2}\end{aligned} \tag{15}$$

Since $2|\mathbf{I}_j| - n \ge 0$, we can easily notice $\frac{d^2}{d\mathbf{c}_j^2}\mathrm{IC}(\mathbf{c})$ is always positive, which essentially implies convexity. This means the largest function value is obtained on the boundary of interval $[|\mathbf{z}_{m,j}|, |\mathbf{z}_{m+1,j}|]$.

Thus, in call cases, the local maximum of $\mathrm{IC}(\mathbf{c})$ lies either at the left boundary or the right one. This observation allows us to search for an optimal c in the set $\{|\mathbf{z}_1|, |\mathbf{z}_2|, ..|\mathbf{z}_n|\}$. □

[4] A Stiefel manifold $\mathbf{V}_k(\mathbb{R}^n)$ is the set of ordered k-tuples of othornomal vetors in $\mathbb{R}^n$.

The proposition allows us to restrict the search space from $\mathbb{R}$ to n points. A naive strategy of searching optimal $\mathbf{c}$ would be to enumerate all points (set $\mathbf{c}_j = |\mathbf{z}_{ij}|$ for $i = 1, \ldots, n$) and find $|\mathbf{z}_{ij}|$ that gives the best objective value. For each $|\mathbf{z}_{ij}|$, evaluating the point set that fall in a bounding box (i.e., $\mathbf{I}_j$) requires time $O(n)$. Hence, the naive search strategy has complexity $O(n^2)$.

By a more careful thinking, the search can be improved to $O(n \log(n))$. The idea is as follows:

- sort $|\mathbf{z}_{ij}|$, $i = 1, \ldots, n$ in ascending order, $O(n \log(n))$. In TensorFlow, sorting can be encoded as a node in computational graph using function **tf.nn.top_k**[5].
- search the $\mathbf{z}_{ij}$ using the new order, $O(n)$. For each $\mathbf{c}_j = |\mathbf{z}_{ij}|$, we compare the objective values obtained by either containing $\mathbf{z}_i$ in the bounding box (i.e., $\mathbf{z}_i \in \mathbf{I}_j$) or without (i.e., $\mathbf{z}_i \notin \mathbf{I}_j$), and keep the larger value of two, $O(1)$. As the points are sorted, $|\mathbf{I}_j|$ is simply the number of the evaluated points, $O(1)$.
- to efficiently evaluate the first term (summation) in objective function (Eq.13), we accumulate the sum along the search of new $|\mathbf{z}_{ij}|$. This step costs $O(n)$. In TensorFlow, this can be encoded using function **tf.cumsum**[6].

Based on the above discussion, we can now evaluate objective function (13) by summing over the objective value over k dimensions. For each dimension j, we evaluate the summand (in rectangular brackets of Equation. 13) on a vector of $\mathbf{c}_j$s ($\{|\mathbf{z}_{1j}|, |\mathbf{z}_{2j}|, ..|\mathbf{z}_{nj}|\}$) and find the $\mathbf{c}_j^*$ that maximizes the summand. Thus, the evaluation of the objective function costs $O(n \log(n) + kn)$.

4 EMPIRICAL EVALUATION

In this section, we present two case studies which demonstrate how clipped projections may help users to explore data. Note that the purpose of our experiments is not to investigate superiority of clipped projections over existing methods for dimensionality reduction. Instead, we aim to investigate whether and to which extent the clipped projections usefully depend on the prior beliefs, in highlighting information that is complementary to them. Because the optimal solution for the specific prior belief assumed above without clipping is equivalent to PCA [3], we indeed compare the results from the method with projections corresponding to the principal components of the data.

4.1 UCI image segmentation dataset

The UCI Image segmentation dataset[7] consists of 210 data points. Each data point corresponds to an small 3×3 region (9 pixels) and was drawn randomly from a database of 7 outdoor images. The images were hand-segmented to create a classification for every pixel. The data points are described by 19 image features (e.g., centroid-row-position, hue-mean, intensity-mean). Each data points is classified as one of the following 7 classes: brickface, sky, foliage, cement, window, path, grass. As preprocessing, we centered the data.

We computed a visualization using $f = 0.01$, meaning that we expect to be able to effectively discern 100 points along one axis. We set the variance of the background distribution is to be the average variance of the data, namely $\sigma^2 = \mathrm{Tr}(\mathbf{X}'\mathbf{X})/nd$. To find an informative projection, we tried 100 random starts and used the best scoring result.

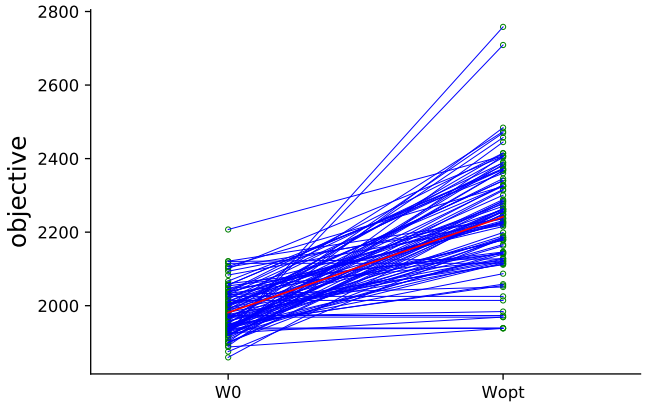

Figure 2: Objective values obtained from 100 random starts on the UCI Segmentation data. Blue lines connect the objective value (green circle) obtained at the initial step ($\mathbf{W}_0$) and the final step ($\mathbf{W}_{\mathbf{opt}}$) of each random start. The red line shows the average initial and final score.

We found that each random start took 0.21 seconds on average. In order to understand something about the difficulty of the optimization problem, and to find whether the gradient descent manages to find a good result, we plotted the objective scores of the initial random w with optimized c and the corresponding final objective scores in Figure 2. The red line shows how the objective value improves on average. We find that the quality indeed varies.

The best clipped scatter plot is shown in Figure. 3c. As a comparison, we consider the scatter plot of the projection against first (x-axis) and second (y-axis) principal component of the full data (3a). The principal components are dominated by a single point that has statistics much unlike the rest of the data. In contrast, our method indeed presents a quite different view. The projection is somewhat different (Figure. 3b) and the information content is greatly increased by clipping several points (3d). We can see the clipped scatter plot shows variation in the center of the data. It also gives the information about direction of the clipped points (points corresponding to the triangular markers on the edges).

Note that in Figure. 3c, the bounding box does not tightly fit the scattered points on the right side. This is due to the constraint that the bounding box is centered, i.e., the distances from origin to the bounding box boundaries are the same in both directions of each dimension. We plan to remove this assumption in the future and have a more flexible bounding box with the location of its center being optimized together with the box size.

4.2 UCI shuttle dataset

The UCI Shuttle dataset[8] consists of 14500 data points and 9 integer features. Each data point belongs to one of the 7 classes: 'Rad Flow', 'Fpv Close', 'Fpv Open', 'High', 'Bypass', 'Bpv Close', 'Bpv Open'. Similar to the case described in the previous section, we set

[5] see TensorFlow API https://www.tensorflow.org/api_docs/python/tf/nn/top_k
[6] see TensorFlow API https://www.tensorflow.org/api_docs/python/tf/cumsum
[7] http://archive.ics.uci.edu/ml/datasets/image+segmentation, 'segmentation.data'
[8] https://archive.ics.uci.edu/ml/datasets/Statlog+(Shuttle), 'shuttle.tst'

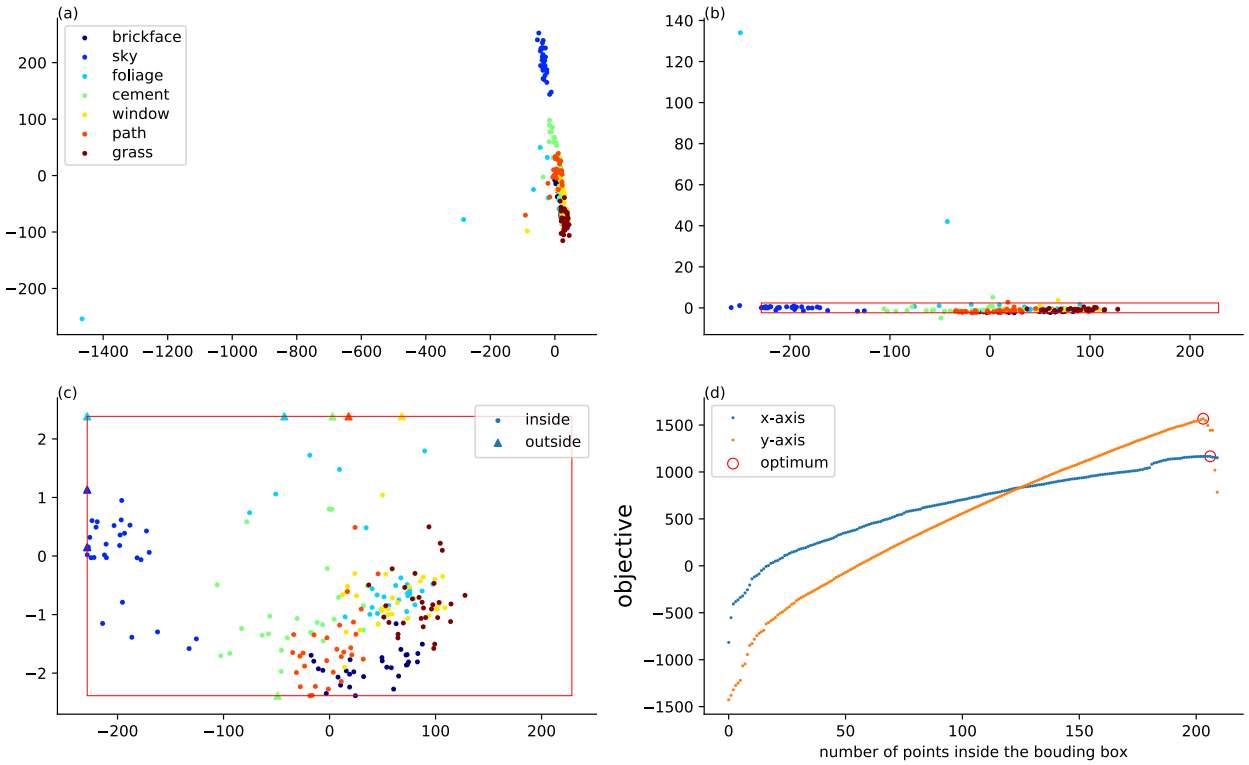

Figure 3: Results on the UCI Segmentation data. (a) The full data projected onto PCA first and second components. (b) The full data projected after optimization but without clipping. The seven class labels are indicated with colors and the red box gives the bounding box. (c) The end-result of our method. Round dots here correspond to data points that are fully inside the box, while triangles correspond to points (partially) outside the bounding box. (d) The objective values obtained for different bounding box sizes c_x, c_y along the directions of W. The PCA projection scores 2050.15 for the two dimensions combined.

$f = 0.01$. The variance of the background distribution is also set to be the average variance of the data, namely $\sigma^2 = \mathrm{Tr}(\mathbf{X}'\mathbf{X})/nd$. The dataset is centered. Same as the previous experiment, we tried 100 random starts. Each random start take on average 0.51 seconds. This may illustrate that the optimization strategy scales well, since the size of shuttle data is ten times larger as the segment data, yet the average time per random start only doubled.

Figure 4 contains the same plots as Figure 3 but for the Shuttle data. The PCA result (4a) shows the variance structure dominated by a set of points with large norms. The bounding box found after optimization is so small in this case, that it is not even visible on the non-clipped scatter plot (4b). The clipped scatter plot (4c) shows that the majority of the data points form a layered structure on a small scale. The layered structure may partially be due to the discreteness of the data. 4d shows the objective values for various $\mathbf{c}$ along the final two projection direction.

For both dimensions, the objective increased very fast initially. This is because the projection most of the points lies close to the origin. When the size of bounding box increase the objective function also increase linearly. The objective function starts to drop rapidly at the end, when the points with large magnitude are included.

4.3 Runtime

Table 1 summarizes the runtime of our method in all experiments presented in this paper. In all cases, we used the solver offered by pyManopt to perform gradient descent (with automatic differentiation provided by TensorFlow) over the Stiefel manifold. We tried ten random starts in all three cases and picked the projection that gives the best objective. Note in the first row of the table, our optimization strategy scales gracefully when the data size increases from Synthetic dataset (148×2) to UCI Segment (210×9) and then UCI Shuttle (14500×9). Although evaluating the objective function involves optimizing the bounding box size, the costs (second row) remain almost constant for increasing data size; the constant overheads from pyManopt and TensorFlow dominate this step.

5 CONCLUSION

A scatter plot of a projection is arguably the most basic way of conveying complete information about a high-dimensional numerical data set. If suitable projections can be found, it promises to empower human data analysts by allowing them to use the remarkable pattern recognition capabilities of human perception: clusters, (local) correlations, outliers, etc. are readily noticed without effort.

Yet, often the scale of a scatter plot is too strongly influenced by a possibly small set of points that are farther than usual from

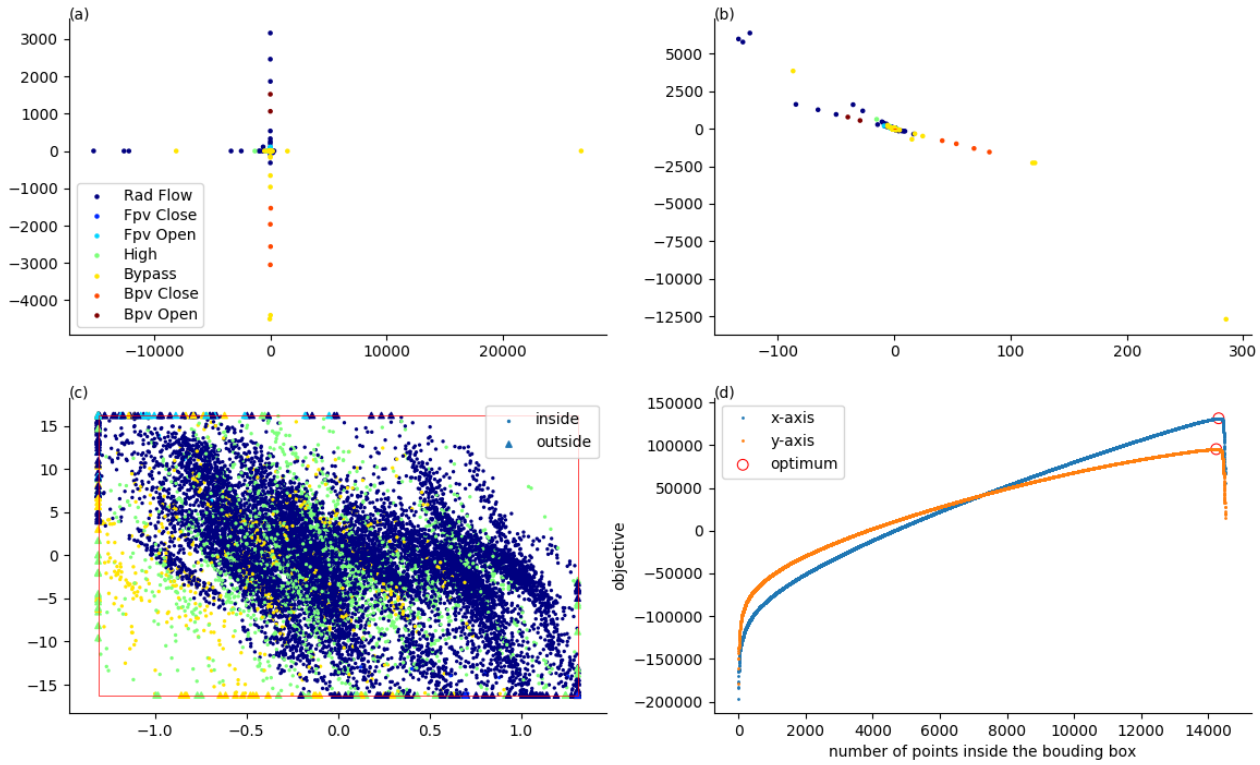

Figure 4: Equivalent to Figure 3, for the UCI Shuttle data. (a) The full data projected onto PCA first and second components. (b) The full data projected after optimization without clipping. The seven class labels are indicated with colors and the red box gives the bounding box. (c) The end-result of our method. Round dots here correspond to data points that are fully inside the bounding box, while triangles correspond to points (partially) outside the bounding box. (d) The objective values obtained for different bounding box sizes c_x, c_y along the directions of W. The PCA projection scores only 69366.9 for the two dimensions combined, against 226630.0 for our method.

	synthetic	UCI Segment	UCI Shuttle
Optimization	0.9717	1.5268	8.2443
Evaluation	0.1307	0.1316	0.1344

Table 1: Runtime (in seconds) of our method for all experiments (§4.3). Each measurement of optimization (first row) is an average over ten runs, where each run consists of ten random start of the gradient based solver from pyManopt and TensorFlow. We also measured the cost of evaluating the objective function (second row). Each measurement is an average over ten evaluations. We used a machine with Intel Quad Core 2.7 GHz CPU and 16 GB RAM.

the centre of the data. As a result, the amount of detail that can be shown for the points closer to the centre is reduced, which is problematic if such points are numerous and the variation among them important. As a result, the overall information conveyed by a scatter plot can be disappointing.

To address this issue, we proposed the notion of a *clipped projection*, which clips the farthest points in a data projection to a bounding box, and subsequently zooms in to let the bounding box fill the plotting area. We then quantified the amount of information a clipped projection conveys about the data, and proposed an algorithm for maximizing this information content over all possible projections and bounding box sizes.

The information content of a clipped projection is formalized by relying on the FORSIED[9] framework [1]. This framework aims to formalize the information content of data mining patterns in a subjective manner: considering the data analyst's prior beliefs about the data. In the current work-in-progress report, we assumed that the user has no prior idea about the data other than its overall scale (which can be easily computed). Our ongoing work, which also builds and improves on previous applications of the FORSIED framework to dimensionality reduction [3], focuses on deploying these principles for other prior beliefs as well, as well as to visualizations of high-dimensional data other than clipped projections.

Acknowledgements. The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement no. 615517, from the FWO (project no. G091017N, G0F9816N), and from the European Union's Horizon 2020 research and innovation programme and the FWO under the Marie Sklodowska-Curie Grant Agreement no. 665501.

[9] 'Formalizing Subjective Interestingness in Exploratory Data mining'.

REFERENCES

[1] T. De Bie. 2011. An information-theoretic framework for data mining. In *Proc. of KDD*. 564–572.

[2] T. De Bie. 2013. Subjective interestingness in exploratory data mining. In *Proc. of IDA*. 19–31.

[3] Tijl De Bie, Jefrey Lijffijt, Raúl Santos-Rodríguez, and Bo Kang. 2016. Informative Data Projections: A Framework and Two Examples. In *Proc. of ESANN*. 635–640.

[4] Bo Kang, Jefrey Lijffijt, Raúl Santos-Rodríguez, and Tijl De Bie. 2016. Subjectively Interesting Component Analysis: Data Projections That Contrast with Prior Expectations. In *Proc. of KDD*. 1615–1624.

[5] James Townsend, Niklas Koep, and Sebastian Weichwald. 2016. Pymanopt: A Python Toolbox for Optimization on Manifolds using Automatic Differentiation. *Journal of Machine Learning Research* 17, 137 (2016), 1–5. http://jmlr.org/papers/v17/16-177.html