Appl. Comput. Harmon. Anal. 44 (2018) 774–785

Contents lists available at ScienceDirect

Applied and Computational Harmonic Analysis

www.elsevier.com/locate/acha

Case Studies

Bigeometric organization of deep nets

Alexander Cloninger [a,*], Ronald R. Coifman [a], Nicholas Downing [b], Harlan M. Krumholz [b]

[a] *Applied Mathematics Program, Yale University, United States*
[b] *Center for Outcomes Research and Evaluation, Yale University, United States*

ARTICLE INFO

Article history:
Received 1 July 2015
Received in revised form 16 April 2016
Accepted 2 August 2016
Available online 4 August 2016
Communicated by Charles K. Chui

Keywords:
Diffusion embedding
Deep learning
Intrinsic organization
Hospital quality

ABSTRACT

In this paper, we build an organization of high-dimensional datasets that cannot be cleanly embedded into a low-dimensional representation due to missing entries and a subset of the features being irrelevant to modeling functions of interest. Our algorithm begins by defining coarse neighborhoods of the points and defining an expected empirical function value on these neighborhoods. We then generate new non-linear features with deep net representations tuned to model the approximate function, and re-organize the geometry of the points with respect to the new representation. Finally, the points are locally z-scored to create an intrinsic geometric organization which is independent of the parameters of the deep net, a geometry designed to assure smoothness with respect to the empirical function. We examine this approach on data from the Center for Medicare and Medicaid Services Hospital Quality Initiative, and generate an intrinsic low-dimensional organization of the hospitals that is smooth with respect to an expert driven function of quality.

1. Introduction

Finding low dimensional embeddings of high dimensional data is vital in understanding the organization of unsupervised data sets. However, most embedding techniques rely on the assumption that the data set is locally Euclidean [7,14,1]. In the case that features carry implicit weighting, some features are possibly irrelevant, and most points are missing some subset of the features, Euclidean neighborhoods can become spurious and lead to poor low dimensional representations.

In this paper, we develop the method of expert driven functional discovery to deal with the issue of spurious neighborhoods in data sets with high dimensional contrasting features. This allows small amounts of input and ranking from experts to propagate through the data set in a non-linear, smooth fashion. We then build a distance metric based off these opinions that learns the invariant and irrelevant features from

* Corresponding author.
E-mail address: alexander.cloninger@yale.edu (A. Cloninger).

http://dx.doi.org/10.1016/j.acha.2016.08.001
1063-5203/

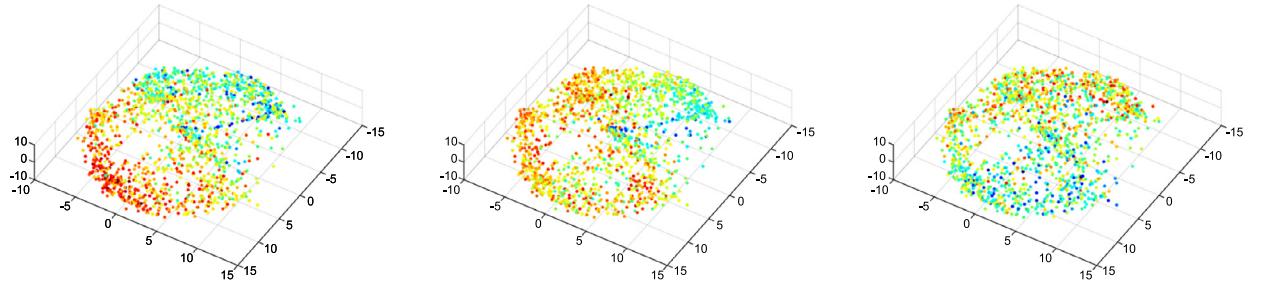

Fig. 1. Organization colored by: (left) risk standardized 30 day hospital wide readmission, (center) percent patients rating overall hospital 9 or 10 out of 10, (right) risk standardized 30 day mortality for heart failure. Embedding generated via bigeometric organization of deep nets. Red is good performance, blue is bad. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

this expert driven function. Finally, we locally normalize this distance metric to generate a global embedding of the data into a homogeneous space.

An example to keep in mind throughout the paper, an idea we expand upon in Section 4, is a data set containing publicly-reported measurements of hospital quality. The Center for Medicare and Medicaid Services Hospital Quality Initiative reports approximately 100 different measures describing various components of the quality of care provided at Medicare-Certified hospitals across the United States. These features range in measuring hospital processes, patient experience, safety, rates of surgical complications, and rates of various types of readmission and mortality. There are more than 5,000 hospitals that reported at least on measure during 2014, but only 1,614 hospitals with 90% measures reported. The measures are computed quarterly, and are publicly available through the Hospital Compare website [8]. The high dimensional nature of these varied measures make comprehensive inferences about hospital quality impossible without summarizing statistics.

Discovering the topology of the hospitals is non-trivial. The features may have significant disagreement, and not be strongly correlated across the population. To examine these relationships, one can consider linear correlations via principal component analysis. The eigenvalues of the correlation matrix do not show the characteristic drop off shown in linear low dimensional data sets. In fact, 76 of the 86 eigenvalues are above 1% the size of the largest eigenvalue. Previous medical literature has also detailed the fact that many of the features don't always correlate [9,4].

For this reason, there does not exist an organization for which all features are smooth and monotonically increasing. This is why the meta-features, and organization, must be driven by minimal external expert opinion. This observation makes the goal of our approach three fold: develop an organization of the data that is smooth with respect to as many features as possible, build a ranking function f that agrees with this organization, and minimize the amount of external input necessary to drive the system.

Our goal is more than just learning a ranking function f on the set of hospitals X. We are trying to characterize the cohort of hospitals and organize the geometry of the data set, and learn a multi-dimensional embedding of the data for which the ranking function is smooth. This gives an understanding of the data that doesn't exist with a one dimensional ranking function. Specifically, we are looking for meta-features of the data in order to build a metric $\rho : X \times X \to \mathbb{R}^+$ that induces a small Lipschitz constant on the function f, as well as on features measured by CMS.

An example of this organization is shown in Fig. 1. The organization is generated via our algorithm of expert driven functional discovery, the details of which are found in Sections 2 and 3. The colors in each image correspond to three notable CMS features: risk standardized 30 day hospital-wide readmission, patient overall rating of the hospital, and risk standardized 30 day mortality for heart failure. This organization successfully separates hospitals into regimes for which each feature is relatively smooth.

Our organization is accomplished via a three step processing of the data. First, we build an initial organization of the data via coupled partition trees [6], and use this partitioning to generate pseudopoints

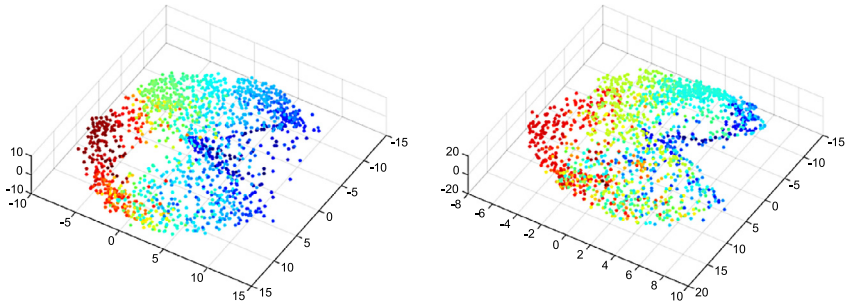

Fig. 2. Two sets of organizations with similar neighborhood structure. The representations used to generate these embeddings are fundamentally different, with the right figure using 5× as many features as the left, and with features being generated with vastly different algorithmic parameters.

that accurately represent the span of the data at a coarse scale. This step is explained in Section 2. This organization is analyzed and used as an input to a series of stacked neural nets, which learn invariant representations of the data and separate disparate clusters of points. This step is explained in Section 3. See [2] for a review of stacked neural nets and deep learning. Finally, we build a metric on that topology by taking a local Mahalanobis distance on the stable neighborhoods (i.e. local z-scoring) [10,15]. This is done in Section 3.3, and guarantees that the induced metric is homogeneous. This means that, if U_x denotes the neighborhood of a point x, then $\rho(x, y)$ for $x, y \in U_x$ measures the same notion of distance as $\rho(x', y')$ for $x', y' \in U_{x'}$.

By taking a local z-scoring of the features, we generate an organization that is dependent only on the neighborhoods U_x, rather than being dependent on the specific representations used. Fig. 2 shows the organizations of the hospitals generated by algorithms with two very different parameter sets which, after z-scoring the neighborhoods, generate similar embeddings. More details about this embedding can be found in Section 4.2.2.

It is important to note that our use of stacked neural nets is different from traditional deep learning applications. We discuss these differences in Section 3.4. The purpose of using deep learning and organizing of the generated representations is to create a notion of fine neighborhoods between points; neighborhoods where the number of neighbors scales smoothly with the distance metric.

We then examine and validate our algorithm on the CMS Quality Initiative features in Section 4.

2. Information organization and expert driven functional discovery

2.1. Training on data with full knowledge

Let the data matrix be

$$M = [v_1, ..., v_N], \tag{1}$$

where $v_i \in \mathbb{R}^m$ is a vector of observations describing the ith data point. Each v_i is allowed to have arbitrarily many missing entries. Define $\text{supp}(v_i) = \{k \in \{1, ..., m\} : v_{i,k} \text{ is observed}\}$.

Due to the missing entries, calculating an affinity between every two points v_i and v_j is not necessarily possible, given that the intersection of the supports of their known values may be small or even disjoint. For this reason, we begin by restricting ourselves to points v_{i_j} that have at most η missing entries. We shall begin by organizing the set of points $\Omega = [v_{i_1}, ..., v_{i_n}]$.

To gain an initial understanding of the geometry of Ω, we consider the cosine affinity matrix A where

$$A_{j,k} = \frac{\langle v_{i_j}, v_{i_k} \rangle}{\|v_{i_j}\| \|v_{i_k}\|}, \tag{2}$$

where the inner product is calculated only on the entries in $\text{supp}(v_{i_j}) \cap \text{supp}(v_{i_i})$. By definition of Ω, this set contains at least $m - 2\eta$ known values.

The cosine affinity matrix serves as a good starting point for learning the geometry of Ω. However, we must develop a way to extend any analysis to the full data M. For this reason, we partition both the data points and the observation sets of Ω. This gives us two advantages: partitioning the data points captures the ways in which different observations may respond to different subsets of Ω, and partitioning the observation sets into similar question groups gives a method for filling in the missing observations in M.

We construct a coupled geometry of Ω using the algorithm developed in [6]. The initial affinity is given by the cosine affinity matrix A, and the iterative procedure is updated using Earth Mover Distance [13].

Remark: Let the final affinity matrix be called $\widetilde{A} : M \times M \to [0, 1]$. Let the eigenpairs of $\widetilde{A}$ be called $\{(\lambda_i, \phi_i)\}$ with $1 = \lambda_0 \geq ... \geq \lambda_N$. Then the organization of M is generated by

$$\Phi^t(x) = [\lambda_1^t \phi_1(x), ..., \lambda_d^t \phi_d(x)], \, x \in M,$$

where d is the dimension of the underlying manifold.

2.2. Filling in missing features

Let $\mathscr{T}_{obs}$ be the hierarchical tree developed on the observations in $\mathbb{R}^m$ from Section 2.1. Let the levels be $\mathscr{X}^1, ..., \mathscr{X}^L$, with the nodes for level l named $\mathscr{X}_1^l, ..., \mathscr{X}_{n(l)}^l$. Let $v_i \notin \Omega$ be a data point with the entry $v_{i,k}$ missing. In order to add v_i into the geometry of Ω, we must estimate the entries in $(\text{supp}(v_i))^c$ to calculate an affinity between v_i and other points.

$\mathscr{T}_{obs}$ gives a tree of correlations between the observations. This allows us to fill in $v_{i,k}$ with similar, known entries. Find the lowest level of the tree (most strongly correlated questions) for which observation $k \in \mathscr{X}_j^l$ and $\exists m \in \mathscr{X}_j^l$ such that $v_{i,m}$ is known. Then the estimate of $v_{i,k}$ satisfies

$$\widetilde{v}_{i,k} = \frac{1}{|\mathscr{X}_j^l|} \sum_{m \in \mathscr{X}_j^l} v_{i,m}. \tag{3}$$

Along with an estimate of $v_{i,k}$, (3) also gives a level of uncertainty for the estimate, as smaller l (i.e. coarser folders) have lower correlation and give larger reconstruction error.

2.3. Expert driven function on the folders

Let $\mathscr{T}_{points}$ be the hierarchical tree developed on the data points in Ω from Section 2.1. Let the levels be $\mathscr{X}^1, ..., \mathscr{X}^L$, with the nodes for level l named $\mathscr{X}_1^l, ..., \mathscr{X}_{n(l)}^l$. As the partitioning becomes finer (i.e. l approaches L), the folders contain more tightly clustered points. This means that the distance from any point to a centroid of a folder becomes smaller as the partitioning becomes finer.

Fix the level l in the tree. The centroids of these folders can be thought of as "types" of data points, or *pseudopoints*. There are two major benefits: there are few pseudopoints relative to n that span the entire data space, and the pseudopoints are less noisy and more robust to erroneous observations.

These pseudopoints are the key to incorporating expert knowledge and opinion. The pseudopoints are easier and much faster to classify than individual points, as there are a small number and they are less noisy than individual points. Also, the pseudopoints effectively synthesize the aggregate performance of multiple hospitals. The classifications generated by experts can be varied, anything from quality rankings to discrete classes to several descriptive features or "meta-features" of the bins. Specifically, the user assigns a set of classes $\mathscr{C}$ and a classification function $g : \Omega \to \mathscr{C}$ such that

$$\forall x \in \mathscr{X}_j^l, \quad g(x) = y_j \in \mathscr{C}. \tag{4}$$

This function is understood as a rough estimate, since the classification is applied to all $x \in \mathscr{X}_j^l$ even though the class is determined only from the centroid of $\mathscr{X}_j^l$.

The function generates a metric $\rho : \Omega \times \Omega \to \mathbb{R}^+$ that has dependencies of the form

$$\rho(x,y) = f\left(x-y; g(x), g(y)\right). \quad f : \mathbb{R}^m \times \mathscr{C} \times \mathscr{C} \to \mathbb{R}^+. \tag{5}$$

This metric needs to satisfy two main properties:

1. $\exists \delta_0$ such that $\rho(x,y) < \epsilon_{dist}$ if $\|x-y\| < \delta$ for $\delta < \delta_0$ and some norm $\|\cdot\|$, and
2. $\rho(x,y) > \epsilon_{class}$ if $g(x) \neq g(y)$.

A metric that satisfies these two properties naturally relearns the most important features for preserving clusters while simultaneously incorporating expert knowledge to collapse non-relevant features. We learn this function using neural nets, as described in Section 3.

3. Deep learning to form meta-features

There are entire classes of functions that approximate the behavior of ρ in (5). For our algorithm, we use neural nets for several reasons. First, the weight vectors on the first layer of a neural net have clear, physical interpretation, as the weight matrix can be thought of as a non-linear version of the weight vectors from principal component analysis. Second, current literature on neural nets suggest a need for incredibly large datasets to develop meaningful features on unsupervised data. Our algorithm provides a way to turn an unlabeled data set into a semi-supervised algorithm, and incorporates this supervision into the nodes of the neural net. This supervision appears to reduce the number of training points necessary to generate a non-trivial organization. Past literature has used radial basis functions as the non-linear activation for deep learning [5], as well as for analysis of the structure of deep net representations [11].

3.1. Neural nets with back-propagation of rankings

For our algorithm, we build a 2 layer stacked autoencoder with a sigmoid activation function, with a regression layer on top. The hidden layers are defined as

$$h^{(l)}(x) = \sigma\left(b^{(l)} + W^{(1)} h^{(l-1)}(x)\right),$$

with $\sigma : \mathbb{R} \to [0,1]$ being a sigmoid function applied element-wise, and $h^{(0)}(x) = x$. The output function $f(x)$ is logistic regression of these activation units

$$f(x) = \sigma(b^{(3)} + V h^{(2)}(x)).$$

The reconstruction cost function for training our net is an L^2 reconstruction error

$$C = \frac{1}{n} \sum_{i=1}^{n} \|g(x_i) - f(x_i)\|^2. \tag{6}$$

The overall loss function we minimize, which combines the reconstruction cost with several bounds on the weights, is

$$L = \frac{1}{n} \sum_{i=1}^{n} \|g(x_i) - f(x_i)\|^2 + \mu \sum_{l} \sum_{i,j} \left(W_{i,j}^{(l)}\right)^2$$

Note that we rescale g if it takes values outside $[0,1]$. We then backpropagate the error by calculating $\frac{\delta L}{\delta w_{i,j}^{(l)}}$ and $\frac{\delta L}{\delta w_i^{(l)}}$ and adjusting the weights and bias accordingly. See [12] for a full description of the algorithm.

Definition 3.1. The deep neural net metric on Ω with respect to an external function f is defined as

$$\rho_{DNN}(x,y) = \|h^{(1)}(x) - h^{(1)}(y)\|.$$

Lemma 3.2. *A deep neural net with a logistic regression on top generates a metric ρ_{DNN} that satisfies Condition 1 from (5) with a Lipschitz constant of $\|W_1\|/4$ with respect to Euclidean distance. The output function f also has a Lipschitz constant of $\|W_1\|\|W_2\|\|V\|/64$ with respect to Euclidean distance.*

The proof of Lemma 3.2 follows immediately from the fact that $\sigma(x)$ has bounded derivative and simple norm inequalities.

Lemma 3.3. *A two layer neural net with a logistic regression on top creates a function f which satisfies a variant of Condition 2 from (5), namely that*

$$\mathbb{E}_{\neq}\left(\|f(x)-f(y)\|^2\right) \geq \mathbb{E}_{\neq}\left(\|g(x)-g(y)\|^2\right) - 2\left(\frac{\max_{i\in\mathscr{C}} S_i \cdot n}{S}\right) C, \tag{7}$$

where $S = \#\{(x,y)\in\Omega\times\Omega : g(x)\neq g(y)\}$, $S_i = \#\{y\in\Omega : g(y)\neq i\}$, and $\mathbb{E}_{\neq}$ is the expected value over the set S.

Moreover, the deep neural net ρ_{DNN} generated also satisfies

$$\mathbb{E}_{\neq}\left(\|f(x)-f(y)\|^2\right) \geq \left(\frac{1}{\prod_{i=2}^{L}\|W_i\|}\right)\left[\mathbb{E}_{\neq}\left(\|g(x)-g(y)\|^2\right) - 2\left(\frac{\max_{i\in\mathscr{C}} S_i \cdot n}{S}\right) C\right]. \tag{8}$$

Proof. We have

$$\begin{aligned}\|f(x)-f(y)\|^2 &= \|f(x)-g(x)-f(y)+g(y)+g(x)-g(y)\|^2\\ &\geq \|g(x)-g(y)\|^2 - \left(\|f(x)-g(x)\|^2 + \|f(y)-g(y)\|^2\right).\end{aligned}$$

Unfortunately, because (6) is a global minimization, we cannot say anything meaningful about the difference for individual points. However, we do have

$$\begin{aligned}\sum_{g(x)\neq g(y)}\|f(x)-f(y)\|^2 &\geq \sum_{g(x)\neq g(y)}\|g(x)-g(y)\|^2 - \sum_{g(x)\neq g(y)}\left(\|f(x)-g(x)\|^2+\|f(y)-g(y)\|^2\right)\\ &= \sum_{g(x)\neq g(y)}\|g(x)-g(y)\| - 2\sum_{x\in\Omega}\#\{y : g(x)\neq g(y)\}\cdot\|f(x)-g(x)\|^2\\ &\geq \sum_{g(x)\neq g(y)}\|g(x)-g(y)\| - 2\left(\max_{i\in\mathscr{C}}\#\{y : g(y)\neq i\}\right)\cdot nC,\end{aligned}$$

where $\#\{y : g(y)\neq i\}$ denotes the number of elements in this set. Let $S = \#\{(x,y)\in\Omega\times\Omega : g(x)\neq g(y)\}$ and $S_i = \#\{y\in\Omega : g(y)\neq i\}$. Then

$$\mathbb{E}_{\neq}\left(\|f(x)-f(y)\|^2\right) \geq \mathbb{E}_{\neq}\left(\|g(x)-g(y)\|^2\right) - 2\left(\frac{\max_{i\in\mathscr{C}} S_i \cdot n}{S}\right) C.$$

This means that by minimizing C, we are forcing the separation of points with different initial ranking to be as large as possible. This enforces Condition 2 of (5) in the aggregate over all such points.

The scaling of $\frac{1}{\prod_{i=2}^{L}\|W_i\|}$ for ρ_{DNN} is a simple application of Lemma 3.2. □

3.2. Heat kernel defined by ρ_{DNN}

The weights generated by the neural net represent "meta-features" formed from the features on Ω. Each hidden node generates important linear and non-linear combinations of the data, and contains much richer information than a single question or average over a few questions.

A problem with neural nets is that they can be highly unstable under parameter selection (or even random initialization). Two identical iterations can lead to completely different weights set. Along with that, back propagation can force points into isolated corners of the cube in $[0,1]^k$.

For this reason, we rerun the neural net K times with varied random seeds, number of hidden layers, sparsity parameters, and dropout percentages. After K iterations, we build the new set of features on points as $\Omega^*(x) = [h_1^{(1)}(x), ..., h_K^{(1)}(x)]$. This defines an adjacency matrix on A with affinity defined between two points as

$$A(x,y) = e^{-\|\Omega^*(x)-\Omega^*(y)\|^2/\epsilon}. \tag{9}$$

Along with that, the final ranking function on M comes from $f(x) = \frac{1}{K}\sum_{i=1}^{K} f_i(x)$. Note that $\|\Omega^*(x) - \Omega^*(y)\|^2 = \sum_{i=1}^{K} \rho_{DNN,i}(x,y)^2$.

The expert driven heat kernel defined in (9) generates an embedding $\Phi : \Omega \to \mathbb{R}^d$ via the eigenvectors $\Phi^t(x) = [\lambda_1^t\phi_1(x), ..., \lambda_1^t\phi_d(x)]$.

For each neural net h_i, we keep the number of hidden layers small relative to the dimension of the data. This keeps the net from overfitting the data to the initial organization function g.

3.3. Standardizing distances to build an intrinsic embedding

While this generates a global embedding based off local geometry, it does not necessarily generate a homogeneous space. In other words, $\|\Phi^t(x) - \Phi^t(y)\| = \|\Phi^t(x') - \Phi^t(y')\|$ does not necessarily guarantee that x and y differ by same amount as x' and y'. This is because $\Omega^*(x)$ and $\Omega^*(y)$ may differ in a large number of deep net features, whereas $\Omega^*(x')$ and $\Omega^*(y')$ may only differ in one or two features (though those features may be incredibly important for differentiation).

For this reason, we must consider a local z-score of the regions of the data. For each point $\Phi^t(x)$, there exists a mean and covariance matrix within a neighborhood U_x about x such that

$$\mu_x = \frac{1}{|U_x|}\sum_{z\in U_x} \Phi^t(z),$$

$$\Sigma_x = \frac{1}{|U_x|}\sum_{z\in U_x} \left(\Phi^t(z) - \mu_x\right)^{\intercal}\left(\Phi^t(z) - \mu_x\right).$$

This generates a new whitened distance metric

$$d_t(x,y) = \frac{1}{2}\left[(\Phi^t(x) - \mu_x) - (\Phi^t(y) - \mu_y)\right]^{\intercal}\left(\Sigma_x^{\dagger} + \Sigma_y^{\dagger}\right)\left[(\Phi^t(x) - \mu_x) - (\Phi^t(y) - \mu_y)\right], \tag{10}$$

where $\Sigma^{\dagger}$ is the Penrose–Moore pseudoinverse of the covariance matrix.

One can generate a final, locally standardized representation of the data via the diffusion kernel

$$W(x,y) = e^{-d_t(x,y)/\epsilon},$$

and the low frequency eigenvalues/eigenvectors of W, which we call $\{(s_i, \psi_i)\}$. The final representation is denoted

$$\Phi^t_{std}(x) = [s_i^t \psi_1(x), ..., s_i^t \psi_d(x)]. \tag{11}$$

The embedding $\Phi^t_{std}(x)$ can be extended from the analysis on Ω to the rest of the points in M via a simple Nystrom extension [3]. New point x is fed through the collection of networks to generate $\Omega^*(x)$, use Nystrom to calculate $\Phi^t(x)$, and finally use Nystrom again to calculate the final embedding $\Phi^t_{std}(x)$.

3.4. Different approach to deep learning

Our algorithm, as we will discuss in detail in Section 3, uses the meta-features from stacked neural nets in a way not commonly considered in literature. Most algorithms use back propagation of a function to fine-tune weights matrices and improve the accuracy of the one dimensional classification function. However, in our algorithm, the purpose of the back propagation is not to improve classification accuracy, but instead to organize the data in such a way that is smooth relative to the classification function. In fact, we are most interested in the level sets of the classification function and understanding the organization of these level sets.

At the same time, we are not building an auto-encoder that pools redundant and correlated features in an attempt to build an accurate, lossy compression (or expansion) of the data. Due to the high level of disagreement among the features, non-trivial features generated from an auto-encoder are effectively noise, as we see in Section 4.2. This is the motivation behind propagating an external notion of "quality".

4. Expert driven embeddings for hospital rankings

4.1. Hospital quality ranking

For preprocessing, every feature is mean centered and standard deviation normalized. We begin by building a questionnaire on the hospitals. Our analysis focuses on the 2014 CMS measures. We put a 2× weight on mortality features and 1.5× weight on readmission features due to importance, because the outcome measures describe the tangible results of a hospitalization are particularly important for patients. The questionnaire learns the relationships between the hospitals, as well as the relationships between the different features. In doing this, we are able to build a partition tree of the hospitals, in which hospitals in the same node of the tree are more similar than hospitals in different nodes on the same level of the tree. As a side note, the weighting on mortality features only guarantees that the intra-bin variance of the mortality features is fairly low.

For the ranking in this example, we use the 5th layer of a dyadic questionnaire tree, which gives 32 bins and pseudohospitals. Experts on CMS quality measures rank these pseudohospitals on multiple criteria and assign a quality score between 1 to 10 to each pseudohospitals. We use an average of 50 nodes per layer of the neural net. Also, we average the results of the neural nets over 100 trials. We shall refer to the final averaged ranking as the deep neural net ranking (DNN ranking) to avoid confusion.

4.2. Embedding of hospitals

The left image of Fig. 3 shows a final embedding of the subset of hospitals used in training the neural net, as well as the full set of hospitals. It is colored by the quality function assigned to those hospitals.

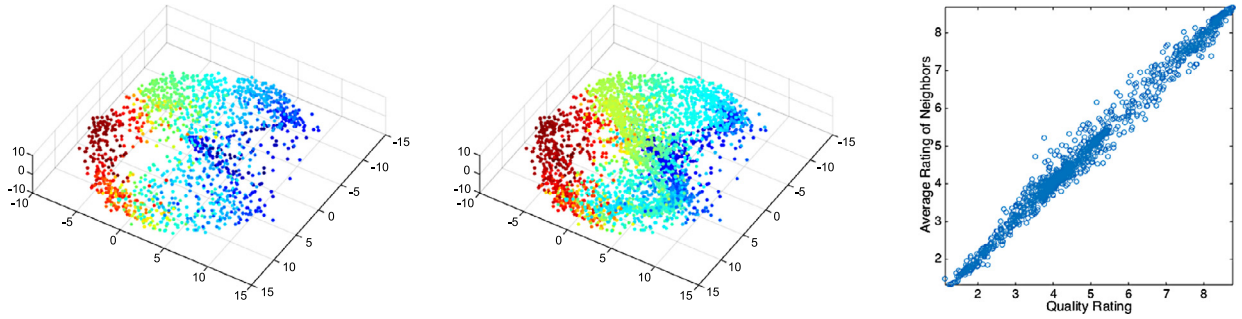

Fig. 3. (left) Embedding of hospitals with at most τ missing entries. (center) Embedding of full set of hospitals. Red corresponds to top quality, blue to bottom quality. (right) Quality function assigned for hospital versus average quality across the neighborhood. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

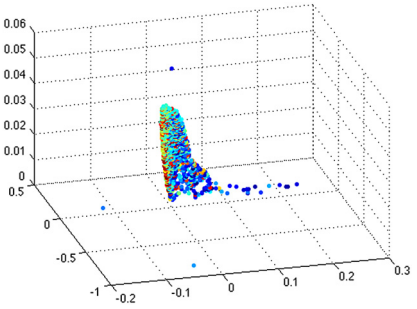

Fig. 4. Embedding of Ω without back propagation of expert driven hospitals rankings. Coloring comes from quality rating function. Notice that without back propagation the embedding is effectively noise compared to hospital quality.

While the goal of the hospital organization is to determine neighborhoods of similar hospitals, it is also necessary for all hospitals in a shared neighborhood to share a common quality rating. Fig. 3 plots the quality function assigned to the hospitals against the weighted average quality function of its neighborhood, where the weights come from the normalized affinities between the given hospitals and its neighbors. The strong collinearity demonstrates that the assigned quality function is consistent within neighborhoods of similar hospitals.

To demonstrate that the expert input back propagation is necessary for a viable ranking and affinity, we include Fig. 4. Here, we build the same diffusion map embedding, but on the features of the autoencoder before back propagation of the expert input function. Due to the small number of data points relative to the number of features, an untuned autoencoder fails to form relevant meta-features for the hospitals.

4.2.1. DNN satisfies conditions for ρ

Fig. 5 verifies that the back propagation neural net satisfies Condition 1 of ρ from (5). The left plot shows the ranking of each hospital plotted against the average of its ten nearest neighbors under Euclidean distance between hospital profiles. The fact that the average correlates with the original quality rating shows that the embeddings of the hospitals remain close if they are close under Euclidean distance.

Fig. 5 shows that the back propagation neural net satisfies Condition 2 of ρ from (5). The histogram in the center shows the DNN affinity between hospitals with different initial quality ratings (i.e. when $g(x) \neq g(y)$). To satisfy Condition 2, $g(x) \neq g(y) \implies \rho(x,y) > \epsilon_{class}$ (i.e. $A(x,y) < 1-\epsilon$). Also, for the histogram on the right we define

$$P_{\neq}(t) = \Pr\left(A(x,y) > t : g(x) \neq g(y)\right), \quad P_{=}(t) = \Pr\left(A(x,y) > t : g(x) = g(y)\right).$$

The histogram on the right shows $\frac{P_{\neq}(t)}{P_{=}(t)}$. The smaller the ratio as t approaches 1, the stronger the influence of the initial ranking on the final affinity.

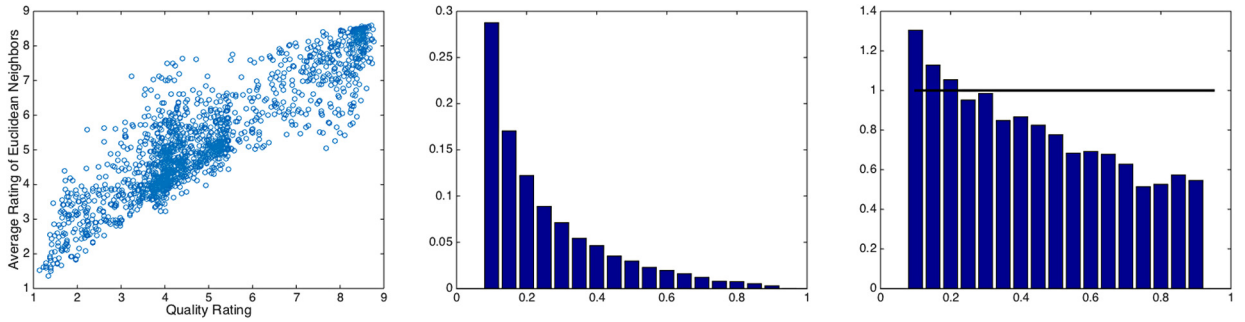

Fig. 5. (left) Quality function for hospital versus average quality function across closest Euclidean points. (center) Histogram of DNN affinity between two points with different initial quality ratings. (right) Normalized histogram of DNN affinity between two points with different initial quality ratings divided by affinity between two points with same initial quality rating. If initial ranking was unimportant, bar graph would concentrate around 1.

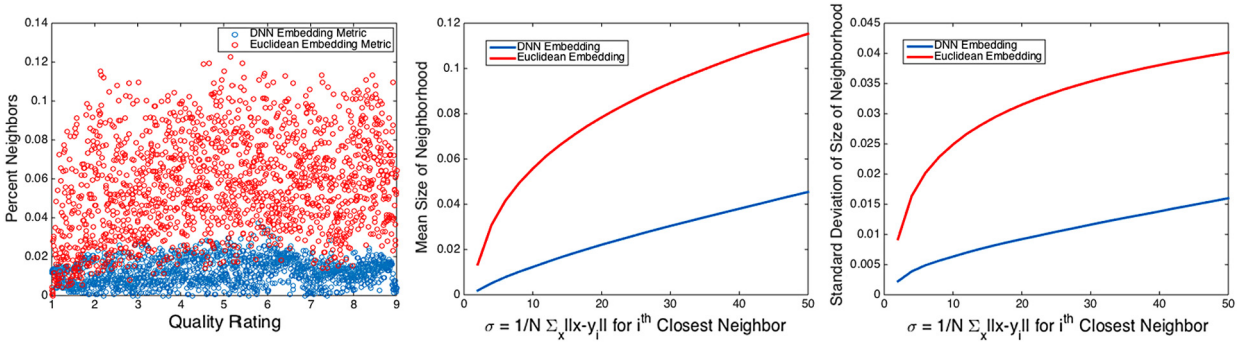

Fig. 6. (left) Percent of points necessary to sum half the total transition probability. Here $\sigma = \frac{1}{N}\sum_x \|x - y_x\|$, where y_x is the 10th closest neighbor of x. (center) the average percentage of neighbors needed to contain half the transition probability from a given point to its neighborhood. (right) the standard deviation of the number of neighbors needed to contain half the transition probability from a given point to its neighborhood. Metrics are generated in same fashion: $K(x,y) = e^{-\|x-y\|^2/\sigma_i^2}$. Here $\sigma_i = \frac{1}{N}\sum_x \|x - y_x\|$, where y_x is the ith closest neighbor of x.

Moreover, we can compare the contraction guaranteed by (8). For the hospital ratings,

$$\mathbb{E}_{\neq}\left(\|f(x)-f(y)\|^2\right) = 2.44,\ \mathbb{E}_{\neq}\left(\|g(x)-g(y)\|^2\right) = 3.36,\ \frac{\max_{i\in\mathscr{C}} S_i \cdot n}{S} = 1.22,\ C = 0.9959,$$

which makes the right hand side of (8) equal 2.15.

Fig. 6 indicates that the DNN metric from (5) gives a better notion of small neighborhoods than a simple Euclidean metric. Each metric defines a transition probability $P(x,y)$. For each point x, the plot finds

$$\underset{I\subset\Omega}{\text{minimize}} \quad \#I \quad \text{subject to} \quad \sum_{y\in I} P(x,y) \geq \frac{1}{2}. \tag{12}$$

It is important for (12) to be small, as that implies the metric generates tightly clustered neighborhoods. As one can see from Fig. 6, the DNN metric creates much more tightly clustered neighborhoods than a Euclidean metric. Fig. 6 gives summary statistics of these neighborhoods for varying diffusion scales.

Another positive characteristic of our DNN embedding is the reduced local dimension of the data. Consider the eigenvalues $\{S_i^t\}_{i=0}^n$ of the diffusion kernels A_t, where the Markov chain A_t describes either comes from the Euclidean metric or the DNN metric. These eigenvalues give us information about the intrinsic dimension of the data, as small eigenvalues do not contribute to the overall diffusion. To compare the eigenvalues across different diffusion kernels, we normalize the eigenvalues by setting $t = \frac{1}{1-S_1^1}$, as this is the average time it takes to diffuse across the system. Cutting off the eigenvalues to determine the dimension is fairly arbitrary without a distinct drop off, but an accepted heuristic is to set the cutoff at

Table 1
Confusion matrix between initial ranking on hospital bins and final ranking from neural net. For simplicity, rankings have been rounded into quartiles for purposes of confusion matrix.

	Final Rank			
	8	11	0	0
First rank	15	730	129	0
	0	97	330	219
	0	0	27	48

$$\dim(A_t) = \max\{d \in \mathbb{N} : S_d > 0.01\}.$$

With this definition of dimension, $\dim(A_t) = 7$ for the DNN metric embedding, whereas $\dim(A_t) = 14$ for the Euclidean metric embedding.

4.2.2. Dependence on initial rankings

Another important feature of a ranking algorithm is its robustness across multiple runs. To demonstrate stability of the neural net section of the algorithm, we run multiple experiments with random parameters to test the stability of the quality function. Fig. 2 shows the rankings for all $1,614$ hospitals across multiple runs of the 100 iterations of the neural net. Given the strong similarity in both quality rating and overall organization, it is clear that the average over 100 iterations of a neural net is sufficient to decide organization of the hospitals.

It is also important to examine the dependence our initial binning and ranking has on the final ranking of the hospitals. Clearly the initial binning is only meant to give approximate ranks, so a strong dependence on these rankings would be problematic. Table 1 shows the confusion matrix between the initial hospital rankings and the final rankings assigned from the neural net. The purpose of the neural net second step is to reclassify hospitals that are binned incorrectly due to spurious correlations.

5. Conclusion

We introduced an algorithm for generating new metrics and diffusion embeddings based off of expert ranking. Our algorithm incorporates both data point geometry via hierarchical diffusion geometry and non-linear meta-features via stacked neural nets. The resulting embedding and rankings represent a propagation of the expert rankings to all data points, and the resulting metric generated by the stacked neural net gives a Lipschitz representation with respect to Euclidean distance that learns important and irrelevant features according to the expert opinions and in automated fashion.

Although the ranking algorithm seems tied to the process of expert rankings of hospitals, the underlying idea of generating metrics in the form of (5) and propagating first pass rankings to the rest of the data, is quite general. For example, this method could be used to propagate sparsely labeled data to the rest of the dataset, with expert bin rankings replaced by the mode of the labeled data in the bin.

This method also touches on the importance of incorporating data point organization into the neural net framework when dealing with smaller data sets and noisy features. Without the expert driven function to roughly organize the data points, the stacked autoencoder fails to determine any relevant features for separation, as shown in Fig. 4.

We will examine the implications of our hospital ratings on health policy, as well as discuss the various types of hospitals in our embedding, in a future paper. Future work will also examine further examination of the influence of data point organization on neural nets and the generation of meta-features. Also, it would be interesting to examine other applications of propagation of qualitative rankings and measures.

Acknowledgments

The authors would like Arjun K. Venkatesh MD, MBA, MHS and Elizabeth E. Drye MD, SM for helping to develop the initial rankings of the pseudohospitals, Uri Shaham for use of his deep learning code, and Ali Haddad for providing the base code for the questionnaire. Alexander Cloninger is supported by NSF Award No. DMS-1402254.

References

[1] Mikhail Belkin, Partha Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, IEEE Trans. Neural Comput. (2003) 1373–1396.
[2] Yoshua Bengio, Deep learning of representations: looking forward, in: Statistical Language and Speech Processing, 2013.
[3] Yoshua Bengio, Jean Francois Paiement, Pascal Vincent, Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering, in: Advances in Neural Information Processing Systems, MIT Press, 2003, pp. 177–184.
[4] E.H. Bradley, J. Herrin, B. Elbel, et al., Hospital quality for acute myocardial infarction: correlation among process measures and relationship with short-term mortality, JAMA (2006) 72–78.
[5] Y. Cho, L.K. Saul, Kernel methods for deep learning, in: Advances in Neural Information Processing Systems, 2009.
[6] Ronald R. Coifman, Matan Gavish, Harmonic analysis of digital data bases, in: Wavelets and Multiscale Analysis, 2011.
[7] Ronald R. Coifman, Stéphane Lafon, Diffusion maps, Appl. Comput. Harmon. Anal. 21 (1) (2006) 5–30.
[8] Center for Medicare and Medicaid Services (CMS), Hospital compare, CMS website: www.medicare.gov/hospitalcompare/, access date: June 15, 2015.
[9] H.M. Krumholz, Z. Lin, P.S. Keenan, et al., Relationship between hospital readmission and mortality rates for patients hospitalized with acute myocardial infarction, heart failure, or pneumonia, JAMA (2013) 587–593.
[10] Prasanta Chandra Mahalanobis, On the generalized distance in statistics, in: Proceedings of the National Institute of Sciences, 1936.
[11] Gregorie Montavon, Mikio Braun, Klaus-Robert Muller, Kernel analysis of deep networks, J. Mach. Learn. Res. (2011) 2563–2581.
[12] R. Rojas, Neural Networks: A Systematic Introduction, Springer Science and Business, Media, 1996.
[13] Will E. Leeb, Ronald R. Coifman, Earth mover's distance and equivalent metrics for spaces with hierarchical partition trees, Yale CS Technical Report, 2013.
[14] L. Saul, S. Roweis, Think globally, fit locally: unsupervised learning of nonlinear manifolds, J. Mach. Learn. Res. 4 (12) (2003) 119–155.
[15] Amit Singer, Ronald R. Coifman, Non-linear independent component analysis with diffusion maps, Appl. Comput. Harmon. Anal. 25 (2) (2008) 226–239.