Hypersphere clustering to characterize healthcare providers using prescriptions and procedures from Medicare claims data

Nathanael Fillmore, Ph.D.[1,2], Sergey D. Goryachev[1], and Jeremy C. Weiss, M.D. Ph.D.[3]
[1]Department of Veterans Affairs, Boston, Massachusetts, USA;
[2]Harvard Medical School, Boston, Massachusetts, USA;
[3]Carnegie Mellon University, Pittsburgh, Pennsylvania, USA

Abstract

We consider the task of producing a useful clustering of healthcare providers from their clinical action signature– their drug, procedure, and billing codes. Because high-dimensional sparse count vectors are challenging to cluster, we develop a novel autoencoder framework to address this task. Our solution creates a low-dimensional embedded representation of the high-dimensional space that preserves angular relationships and assigns examples to clusters while optimizing the quality of this clustering. Our method is able to find a better clustering than under a two-step alternative, e.g., projected K means/medoids, where a representation is learned and then clustering is applied to the representation. We demonstrate our method's characteristics through quantitative and qualitative analysis of real and simulated data, including in several real-world healthcare case studies. Finally, we develop a tool to enhance exploratory analysis of providers based on their clinical behaviors.

Introduction

Exploratory analysis and visualization of health data are central to the advancement of clinical care. Clinical data are naturally high-dimensional and sparse because most medical concepts are not applicable to most individuals. For example, any individual physician performs a small subset of procedures, and any patient fills prescriptions for a small subset of medications. Our work seeks to provide effective representations of these settings by producing useful clusterings of high-dimensional sparse count data. Our focus is in the analysis of large-scale healthcare data arising from claims and health records databases, where we are interested in clustering patients or providers based on the number of time each billing code, procedure code, or drug prescription is observed.

Similarity between sparse, high-dimensional points is typically better measured through angular distances like cosine similarity rather than euclidean distance. As a result, many standard clustering methods will perform poorly in these tasks, and clustering algorithms in lower-dimensional representations have been developed to address this problem. For example, one popular approach adopts a two-step process: (a) learn a low-dimensional representation, (b) apply clustering over the low-dimensional representation, which is illustrated by the exemplar of principal components analysis (PCA) to find a low-dimensional representation followed by k-means to cluster points based on this representation[1]. A variety of deep learning methods have also been used to find a low dimensional representation in the first step[2]. For example, Tian et al.[3] use stacked autoencoders to learn a representation, followed by k-means. Similarly, Trigeorgis et al.[4] define a deep non-negative matrix factorization and use the representation thus learned to cluster data with k-means.

A major drawback of this two-step process is that the representation learned in the first step is not necessarily well suited to producing a clustering in the second step. For example, autoencoding is designed to minimize reconstruction loss, and a representation that minimizes reconstruction loss is not guaranteed – or even particularly likely – to map points to a meaningful clusterable space.

To address this issue, several methods have been defined that integrate both steps – reconstruction learning and clustering – in a single deep learning framework. This enables the representation to be informed by the clustering, not just by its ability to reconstruct data, and thus creates the possibility of finding both a more clusterable representation and a better clustering. For example, Xie et al.[5] add a centroid-based soft clustering criterion to an autoencoder backbone and iteratively refine the autoencoder's parameters so as to improve the quality of the clustering. Similarly, Yang et al.[6] define a method that integrates an autoencoding deep neural network and a k-means clustering loss, and fits this model with an alternating stochastic gradient descent algorithm.

While these approaches show excellent performance on certain types of data, *e.g.* imaging, they are not well-suited to sparse count data. This is primarily due to their dependence on autoencoding with euclidean distances for representation learning. Even with additional constraints to encourage the representation to be well-clusterable, the autoencoders may not retain angular information relevant to a representation for high-dimensional sparse data. The use of centroid-based clustering compounds this problem because the identification of an appropriate centroid location may not be representative of any member of the cluster.

We argue that, instead of a centroid-based approach, an angular-based clustering is better suited for high-dimensional sparse data and that, therefore, an angular-based representation should be adopted. This representation constraint motivates our development of a novel clustering approach that embeds the representation on the surface of a hypersphere, where the angular distance from the axes or poles naturally provides cluster membership. While several support vector clustering algorithms have the option to incorporate angular distance into the kernel defining an embedded space[7,8], these methods do not preserve angular representation during clustering (instead the embedding is transformed into a graph), and these methods have not been shown to be scalable to the large sample sizes present in our applications.

Our contribution. We propose a method that learns a low-dimensional embedding with geometric constraints that enables simultaneous representation learning and cluster determination. We propose an architecture that simultaneously (1) optimizes signature reconstruction in an autoencoding framework, (2) embeds the signature on a low-dimensional hypersphere surface, and (3) assigns examples to clusters in a soft clustering by angular similarity. We further provide the following options. To account for the setting the case where we have expert knowledge about co-cluster membership, we provide a placeholder for an accessory loss to encode this information during optimization. The degree to which each component is prioritized can be adjusted by the user to enable enhanced exploratory analysis. To efficiently compare clusterings with different numbers of clusters, we also build our framework with a fixed hypersphere representation tied to lower-dimensional clustering hyperspheres.

We demonstrate our method's characteristics and its advantages over previous work in simulations, and we illustrate the value of this approach through several case studies: provider characterization from Medicare claims and imaging characterization by radiology notes. We demonstrate the utility of our method with an informatics tool that characterizes providers based on their procedure and prescription signature in Medicare claims.

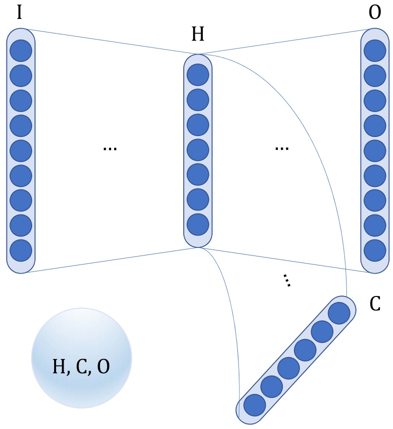
Method

We first describe the method and architecture with respect to inputs and outputs, then describe the distances and losses necessary for learning the representations, and finally specify customizations and implementation parameter choices.

Consider our input of a sparse count matrix M with N rows and K columns. We are interested in clustering over rows (providers) and columns (prescriptions and procedures). We will consider clustering across rows and columns separately. First, we construct an autoencoder consisting of an encoder Ψ_1 and a decoder Ψ_2. For x a row of matrix M, define the hidden representation h of size H such that $\Psi_1(x) = h$ and $\hat{M}$ as the reconstruction given by $\hat{M} = [\Psi_2 \circ \Psi_1(x),\ \forall x \in M]$. Define Ψ_3 as the function transformation for the clustering module that takes as input h and outputs cluster membership probabilities c, a non-negative vector of size C that sums to 1: $\Psi_3(h) = c$. Figure 1 illustrates the framework for the neural network.

Next we define our losses which encode our desire to create a representation that maintains pairwise distances while endowing an angular distance for clustering. Unlike support vector clustering that uses all pairwise distances with complexity $O(N^2)$, our method uses pairwise batch distances $O(B^2)$ with batch size B, which alleviates the computational bottleneck when N is large. Specifically, we propose to learn a hidden representation $h = \{h^\circ, h^{||\cdot||}\}$ given by angle h° and norm $h^{||\cdot||}$, such that the distance measure between examples in M is approximately preserved in the angle h°. To achieve this, we define batch pairwise similarity and its loss $\mathcal{L}_{h^\circ,O}$ as a penalization to the autoencoder loss as follows. Define h_B and $\hat{M}_B$ the representation and reconstructions for a batch of examples with sizes $B \times \{\cdot\}$ for H and M respectively. Let $\delta(x_i, x_j)$ be the pairwise distance measure.

Define δ_B the batch distance measure, *i.e.* $\delta_B(\{x_1, \ldots, x_B\}) = [\delta(x_i, x_j)]_{ij}\ \forall i, j \in \{1, \ldots, B\}$. Then, $\mathcal{L}_{h^\circ,O} = \frac{1}{B}||\delta_B(h_B^\circ) - \delta_B(\hat{M}_B)||_2^2$ captures the differences between the pairwise distances in h_B° and $\hat{M}_B$.



Layer	Index	Activation
(I) Input: $B \times K$	–	
Dense	0	LReLU
Haar wavelet	–	–
Expand, permute, max-pool	1	–
Sum (0) and (1)	–	–
$\{$Batch norm, Dense$\}_{\times 3}$	–	LReLU
Dense	–	LReLU
(H) Hidden: $B \times H$	2	
Dense$_{\times 3}$	–	LReLU
(O) Output: $B \times K$	–	
Copy (2)	–	–
Dense$_{\times 2}$	–	LReLU
Dense	–	Softmax
(C) Cluster: $B \times C$	–	

Figure 1: Autoencoder with clustering module. Nodes are Input (I), Output (O), Hidden (H), and Cluster (C). H and O are constrained by a batch distance measure; we use cosine similarity for sparse count data. H (h°) and C are constrained by batch cosine similarity that ties the angular representation to the cluster probability. See details in the text. The layers and connectivity are shown in the Table on the right. LReLU: leaky rectified linear unit

While $\mathcal{L}_{h^\circ,O}$ encourages matching distances in h° and $\hat{M}$, two vectors in h could have the same angle but different embedding locations, *i.e.*, different $h^{||\cdot||}$. Note this could be problematic because the hidden vectors are used to learn cluster membership probabilities c, and different embedding locations mapping to the same cluster probability vector could induce unwanted within-cluster heterogeneity. To encourage approximate injectivity, we introduce the loss $\mathcal{L}_{||\cdot||=1}$ that penalizes hidden representations away from the surface of the unit norm hypersphere. We could enforce this as a hard constraint, however, the ability to violate the constraint may facilitate alignment of the embedded representation angle with that of the output space.

Now we leverage the angular embedding property of the hidden representation. The key is to note that for a probability vector that sums to 1, its element-wise square root vector has unit norm and can be interpreted as an angle. Therefore, like the previous loss, we can match the angular representation of $c^{\frac{1}{2}}$ to that of h° with batch cosine similarity.

Define $c_B = \Psi_3(h_B)$ the batch of probability vectors indicating cluster membership. Note that the cosine similarities of rows of $c_B{}^{\frac{1}{2}}$ are non-negative and we are not interested in incurring loss due to differences between 0 and negative cosine similarities. We define $\mathcal{L}_{c,h^\circ} = \frac{1}{B}||\delta_B(c_B{}^{\frac{1}{2}}) - \max(\delta_B(h^\circ_B), 0)||^2_2$.

We consider additional loss terms to assist customizable representation learning. First, we introduce an entropy loss term to encourage cluster probabilities to be spread across more than one cluster. This encourages large clusters with overlap to spread across multiple clusters to reveal subgroup characteristics and enables injecting belief about characteristics of the clustering. For optimization, it encourages exploration in cluster membership and could help avoid local optima.

Second and optionally, the framework can use accessory information to inform the clustering. Our primary framework encodes a single hidden representation for clustering, but does allow for specifying multiple numbers of clusters C. In place of a single softmax, the terminal layer C can be a concatenation of possible numbers of clusters, *e.g.* $\{2, 3, \ldots, 100\}$, where the softmax is applied to nodes corresponding to each of the possible numbers. Therefore, one can provide a cluster assignment of size C_{in} that informs the hidden representation and request a more or less granular clustering C_{out}. Unlike hierarchical clustering, this extension produces a multiarchy.

Thus, our overall objective function is $\sum_i \lambda_i \mathcal{L}_i$ for $\mathcal{L}_i \in \{\mathcal{L}_{M,\hat{M}}, \mathcal{L}_{c,h^\circ}, \mathcal{L}_{h^\circ,O}, \mathcal{L}_{||h^\circ||=1}, \mathcal{L}_{entropy}, \mathcal{L}_{C_{in}}\}$. Unless otherwise specified, we set λ_i respectively: $\lambda_i \in \{1, 1, 10^{-1}, 10^{-1}, 10^{-4}, 0\}$.

Finally, we describe our particular implementation choices, and suggest the reader to refer to the autoencoder frame-

Table 1: Quantitative evaluation on simulated data, relative to the true cluster labels. Data are generated as described in the text, with baseline parameters set at centroids=25, samples=1000, dims=1000, sod=0.01, explode=10000, and varied in each dataset as indicated. Data are clustered using our method (Sphere), and the closest neural embedding clustering techniques DCN, SAE+KM, and DEC. The best score is in boldface.

	Adjusted Rand Index (ARI)			
Dataset	Sphere	SAE+KM	DEC	DCN
baseline	**1.00**	0.32	0.10	0.39
samples=100	**0.82**	0.36	0.26	0.36
dims=100,000	**0.64**	0.03	0.00	0.01
explode=1,000,000	**1.00**	0.36	0.12	0.47
centroids=100	**1.00**	0.18	0.00	0.18
sd=0.1	**0.25**	0.02	0.00	0.02

Table 2: Quantitative evaluation of variants of our method. Each column corresponds to a different variant of our own method, as defined in the text. The best score(s) is in boldface.

	Adjusted Rand Index (ARI)				
Dataset	Sphere	Encoder	HKM	Clu:t-SNE	Monotonic
baseline	**1.00**	0.95	0.21	0.96	**1.00**
samples=100	0.82	0.43	0.09	**0.84**	0.75
dims=100,000	**0.64**	0.05	0.01	0.13	0.16
explode=1,000,000	**1.00**	0.99	0.91	0.93	**1.00**
centroids=100	**1.00**	0.72	0.61	0.08	0.75
sd=0.1	0.25	0.02	0.01	0.24	**0.41**

work shown in Figure 1. In pre-processing we log transform each element using the function $f(x) = \log(1 + x)$ to prevent overemphasis of common events in the angular representation. We use cosine similarity for our distance measure between h and $\hat{M}$ and between h and c as it is a natural distance representation for sparse count data. In the architecture, the permute-pool layer copies the tensor P times, permutes the values, and performs max-pooling over the P dimension with size 3 and stride 2. In our experiments we set $B = 128$, $P = 4$, and $H = 128$. We set C to be twice the desired number of centroids, and in post-processing we iteratively merge the clusters identified based on maximum pairwise cluster distances.

Results

We evaluate our method on real and simulated data. Our results on simulated data show (1) that our method produces superior clusterings than other recently proposed deep clustering methods in this setting, and (2) how different components of our method affect the results of the method. Our case study evaluations on three real-world healthcare data show (1) the quality of our method's clustering results, (2) the breadth of applicability of this method in a real-world context, and (3) the method's scalability on large real-world data.

Quantitative evaluation

First, on simulated data generated to share properties of our target healthcare applications, we show that our method typically finds a clustering that, under several measures, is more similar to the ground truth labels from the simulation than competing methods are.

We generate data as follows. Centroids are sampled from a multivariate normal $\mathcal{N}(\mathbf{0}, \mathbf{I})$ and normalized onto the unit sphere. We generate an equal number of samples for each centroid by adding noise distributed according to $\mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$. Then we translate these points away from the origin by multiplying each point by an "explosion" factor κ drawn from a random uniform on $[1, \kappa]$. The purpose of this step is to make the data more similar to that in our healthcare applications, where patients and providers often have varying, *i.e.*, non-constant, norm of total encounter counts.

Thus, the simulation is parameterized by the number of centroids, the number of dimensions, the explode factor, the number of samples, and the standard deviation of the noise. We set each parameter to a baseline level and vary one at a time to test our algorithm.

Comparison to other methods. We cluster the data using our hypersphere clustering method (“Sphere”), as well as three leading deep clustering frameworks that are similar to our method in combining autoencoders and clustering criteria: Stacked Autoencoder plus K-means (SAE+KM)[3], Deep Embedded Clustering (DEC)[5], and Deep Clustering Network (DCN)[6]. We evaluate results of these clustering methods using the Adjusted Rand Index (ARI) relative to the ground truth labels from the simulation.

Results are shown in Table 1. Our method produces better clusterings than the other methods on all six simulated datasets, as measured by ARI. Of note, the two-step SAE+KM method performs approximately as well as the integrated DCN and DEC methods on this data. This suggests that merely integrating clustering and representation learning into a single framework is not enough on its own to produce a good clustering; rather, it is also essential in this context for the representation to capture the relevant geometry of the data, as our method does.

Comparison to variants of our method. Next, we compare our method to variants of our method. These variants are created by omitting different components of the method one at a time, and thus, this evaluation demonstrates the importance of each component.

In the first variant (“Encoder”), we remove the decoder from the framework. Doing so results in worse clustering performance on all six datasets. The possible limitation of decoder removal is that h no longer needs to preserve the information in the input because the network does not need to reconstruct the input.

In the second variant (“HKM”), we run k-means in the hidden, embedded space our method constructs, instead of using our method’s clustering module. The embedded space is not designed to work well with k-means, and in fact, is constrained to have similar geometry to the original space, where k-means performs poorly. Likely for this reason, the HKM variant performs worse than our method on all datasets.

In the third variant (“Clu:t-SNE”), we substitute a t-SNE manifold constraint for $\mathcal{L}_{h^\circ,O}$, with the exception that to conform with the overall architecture of our method, instead of computing all $O(N^2)$ pairwise distances, we define the KL-loss based on $O(B^2)$ batch pairwise distances. This variant slightly outperforms our method when sample size is small (“samples=100”), but otherwise performs worse than our method.

The final variant (“Monotonic”) adds a constraint to our method rather than removing a component. Specifically, motivated by autoregressive flow literature[9], we apply positive weight constraints and strictly monotonic activations (already present) to our network to get a autoregressive network. This variant outperforms ours in one case, but otherwise achieves equal or worse performance.

Case study: Biclustering healthcare providers, prescriptions, and procedures with Medicare data

The quantitative results in the previous section show that our method performs quite well on simulated data that we constructed to have properties of real-world high-dimensional sparse count healthcare data. In this section, we complement these quantitative results with a case study of provider characterization from procedure and prescriptions claims in Center for Medicare and Medicaid Services (CMS) Part D data. We compare our method’s results to those obtained by k-means, and use our method to explore the clusters identified.

Objectives and Data. The purpose of this case study is to cluster healthcare providers, prescriptions, and procedures based on data obtained from CMS claims, with the ultimate objective to gain insights on providers’ patterns of care based on this clustering, as well as on groupings of prescription and procedure use. To that end, we obtained Medicare Provider Utilization and Payment Data: Part D Prescriber Summary Table CY2015[10], which tabulates all prescriptions and procedures given under the Medicare Part D program in 2015 in the United States. This data consists of approximately 10^6 providers, 10^4 procedures, and 10^3 drugs. We used our method and k-means to separately

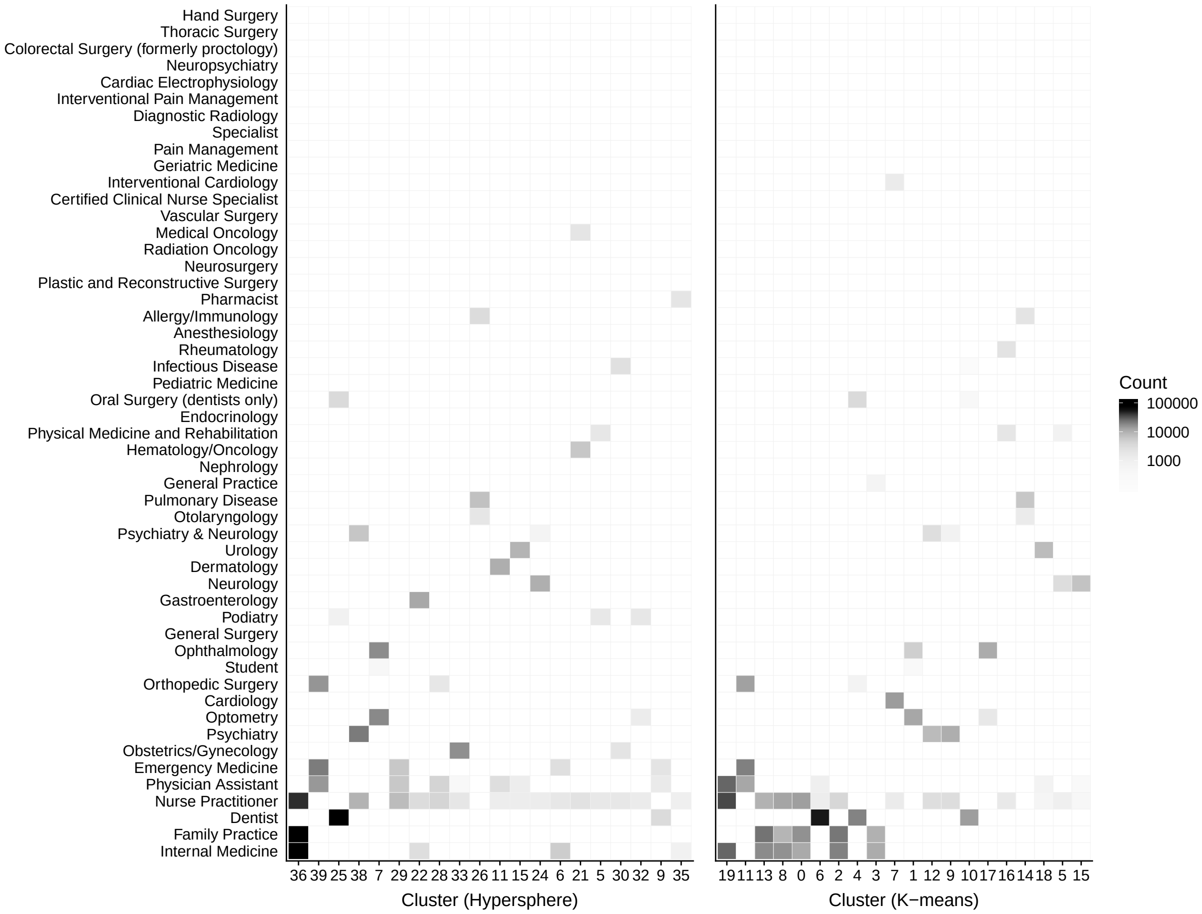


Figure 2: Provider clustering by (a) hypersphere clustering, (b) k-means. Each column corresponds to a cluster and each row a specialty (only the most common ones are shown). We display the top three specialties associated with providers in that cluster, and order the axes according to number of providers and size of cluster. Relative to K-means, hypersphere clustering provides clusters with more diversity in top three specialties.

cluster healthcare providers, prescriptions, and procedures based on this data. Note that the CMS database assigns each provider a specialty code. We do not use these codes for clustering, but we do reference them below to further understand each clustering. In the interest of space, we only show results for a clustering of providers based on the prescriptions they gave and assess the quality of the resulting clusters. We focus our comparison on hypersphere clustering versus k-means given its popularity in healthcare applications[11,12].

Comparison to k-means. The clusterings formed by our method and by k-means, each with 20 clusters, are shown in Figure 2, where the top three specialties within the identified clusters are visualized. The full list of included specialists per cluster is shown in the Appendix. On examination of these results, our method produces qualitatively better clusters than k-means does. For example, our clustering consistently includes a larger fraction of specialists in specialist clusters, *e.g.*, Emergency Medicine (20k, Cluster 39), Dentist (85k, Cluster 25), and Psychiatry (21k, Cluster 38). Our clustering, unlike k-means, also identifies clean obstetrics and hematology oncology clusters. Although k-means identifies a cardiology and interventional cardiology cluster that our method does not initially identify, our method identified this cluster and merged it (Cardiology (18k), Nurse Prac (3k), Internal Medicine (2k)) into the internal medicine cluster (Cluster 38) in post-processing.

Our method also provides insight in regard to providers in specialties that are not as common. For example, our method's urology cluster also includes a large fraction of the radiation oncologists in our dataset (see Appendix). On

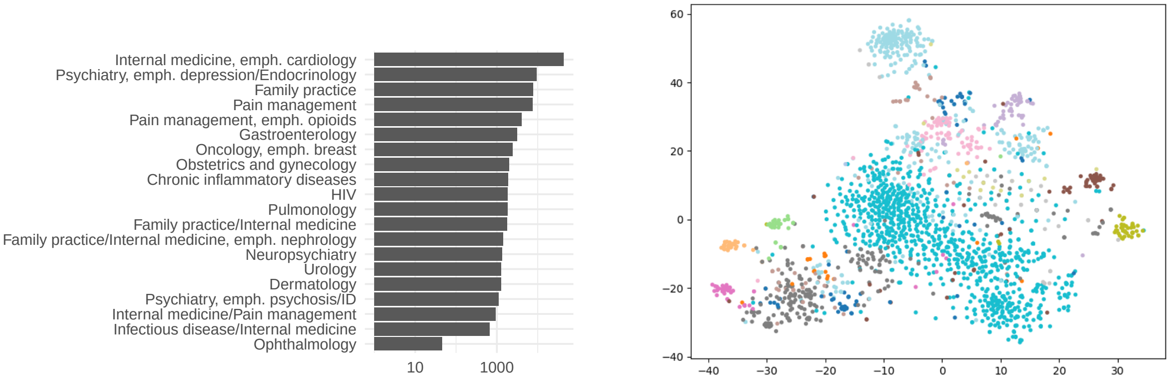


Figure 3: Left: histogram of nurse practitioner counts by cluster. Cluster names are given by clinical assessment of top 10 prescriptions. Right: t-SNE representation of 2000 randomly selected nurse practitioners. Cluster membership is indicated by color, and the axes are the dimensions of the t-SNE embedding.

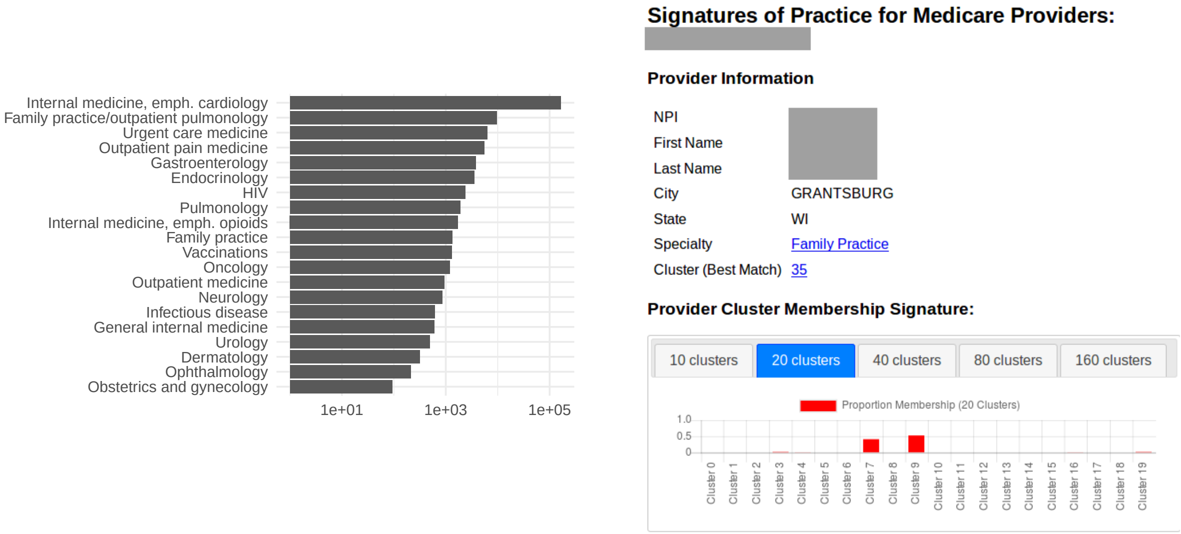
detailed assessment of this cluster, we found that urology medications tamsulosin and finasteride were most commonly prescribed in this cluster and that radiation oncologists most commonly prescribed tamsulosin, followed by hydrocodone/acetaminophen and dexamethasone, possibly for prevention and treatment of complications of radiation therapy. Our clustering identified radiation oncologists and urologists as being similar according to the drugs they commonly prescribe, a finding that would not be identified through the use of a standard ontology alone.

To investigate our method's biclustering (clustering on dimensions of provider and prescriptions), we investigated three questions: (1) does the clustering provide insight into how nurse practitioners provide care, (2) similarly, can we differentiate among family practice and internal medicine doctor subtypes, and (3) is the pain management cluster identifying commonality in opioid prescribing despite provider membership to many specialties?

Nurse practitioners. Nurse practitioners (NPs) tend to provide care in care units defined by medical specialties and subspecialties, but the CMS does not provide characteristics at that granularity. To characterize the type of care they provide, we can investigate their membership to identified clusters. Figure 3 (left) shows the breakdown of membership across clusters. We pulled the top 10 prescriptions administered by NP cluster members, as defined by mean $\log(1+\cdot)$, and used our clinical expertise to characterize the care provided. Evidently, the NPs do cluster based on prescription behavior along lines of medical specialties. The top 10 medications for each cluster are provided in the Appendix. To empirically demonstrate the variation in the NP clustering, Figure 3 (right) shows a t-SNE(cos) plot of a random subset of 2000 NPs to illustrate their similarity (manifold distance) alongside their cluster membership (color). This demonstrates that the method does provide cluster separation and could be used to select NPs based on approximate specialization for future investigation.

Family practice and internal medicine. Similar to nurse practitioners, the specialty titles "family practice" and "internal medicine" are underdifferentiated. To investigate these providers' prescription patterns, we inspected the FP/IM predominating clusters for differences. Figure 4(a) shows the breakdown of the providers across clusters, with labels corresponding to average care descriptions of the members in those groups. We selected three of the clusters that roughly corresponded to IM emphasis cardiology (IMC), IM emphasis infectious disease (IMID), and FP emphasis infectious disease (FPID). We compared the three clusters by listing the three prescriptions with the largest mean $\log(1+\cdot)$ difference in prescribing.

- Compared to FPID providers, IMC providers prescribed more general IM medication $\{$"levothyroxine", "atorvastatin", "lisinopril"$\}$ and fewer antibiotics $\{$"levofloxacin", "amoxicillin-clavulanate", "azithromycin"$\}$.
- Compared to IMID providers, IMC providers prescribed more general internal medicine medication



Signatures of Practice for Medicare Providers: [redacted]

Provider Information

NPI	[redacted]
First Name	
Last Name	
City	GRANTSBURG
State	WI
Specialty	Family Practice
Cluster (Best Match)	35

Provider Cluster Membership Signature:

Figure 4: Left: histogram of FP and IM provider counts by cluster. Cluster names given by clinical assessment of top 10 prescriptions. Right: snapshot of tool showing cluster membership for a selected physician, demo at `http://4e46.net/docproc/`.

{“atorvastatin”, “levothyroxine”, “lisinopril”} and fewer outpatient antibiotics {“clindamycin”, “amoxicillin”, “doxycycline”}.

- Compared to FPID, IMID providers prescribed stronger community-acquired pneumonia antibiotics {“levofloxacin”, “azithromycin”, “amoxicillin-clavulanate”} and fewer older and narrow-spectrum antibiotics and community care medications {“clindamycin”, “amoxicillin”, “ranitidine”}.

Pain management. Table 3 contains, by specialty, the top three medications prescribed by providers in the pain management cluster. The clustering demonstrates that, irrespective of the specialty, these providers prescribe large quantities of various opioids, including oxycodone, hydrocodone, morphine, fentanyl and others. We also compared the opioid-prescribing cluster against the other two clusters containing over 100 pain management specialty providers by listing the three prescriptions with the largest mean $\log(1+\cdot)$ difference.

- Compared to the first cluster, the opioid-prescribing cluster prescribed more {“oxycodone”, “morphine”, “gabapentin”} and less {“celphalexin”, “prednisone”, “tramadol”}.
- Compared to the second cluster, the opioid-prescribing cluster prescribed more {“oxycodone”, “morphine”, “gabapentin”} and less {“ibuprofen”, “acetaminophen with codeine”, “azithromycin”}.

We developed a tool at `http://4e46.net/docproc/` to facilitate learning about providers, their commonalities and memberships based on the their procedure and prescription signature in Medicare Part D claims data. This tool is illustrated in Figure 4 (right). An interested user can search by provider, cluster, or specialty to better understand group characteristics as well as the signature of practice for individual practitioners.

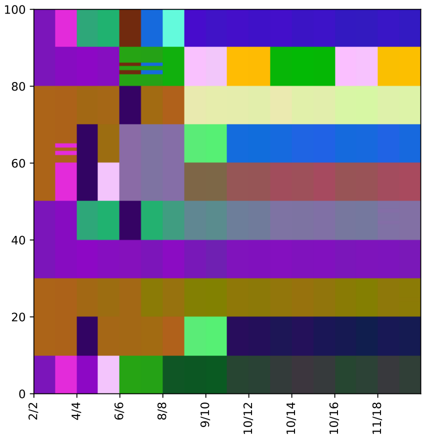
Other Case Studies

We also carried out several additional case studies. These case studies (1) demonstrate flexibility of the method with respect to number of desired clusters, (2) demonstrate the ability of our method to work well on real-world datasets across domains including text.

Single-representation multiple clustering. To illustrate single-representation multiple clustering, we run the simulation variant with 100 samples and 10 clusters, with samples 0-9 in cluster 0, 10-19 in cluster 1, and so on. Figure 5

Table 3: Top three medications by specialty for providers in the opioid-prescribing cluster.

Specialty	Prescriptions
Anesthesiology	oxycodone hcl, hydrocodone/acetaminophen, oxycodone hcl/acetaminophen
Emergency medicine	oxycodone hcl, hydrocodone/acetaminophen, prednisone
Family practice	oxycodone hcl, hydrocodone/acetaminophen, morphine sulfate
General practice	oxycodone hcl, hydrocodone/acetaminophen, gabapentin
General surgery	oxycodone hcl, hydrocodone/acetaminophen, oxycodone hcl/acetaminophen
Hematology/oncology	oxycodone hcl, morphine sulfate, ondansetron hcl
Hospice and palliative care	oxycodone hcl, morphine sulfate, fentanyl
Internal medicine	oxycodone hcl, morphine sulfate, hydrocodone/acetaminophen
Interventional pain management	hydrocodone/acetaminophen, oxycodone hcl, oxycodone hcl/acetaminophen
Medical oncology	oxycodone hcl, ondansetron hcl, dexamethasone
Neurology	oxycodone hcl, gabapentin, hydrocodone/acetaminophen
Neurosurgery	oxycodone hcl, hydrocodone/acetaminophen, gabapentin
Nurse practitioner	oxycodone hcl, hydrocodone/acetaminophen, morphine sulfate
Obstetrics/gynecology	oxycodone hcl, hydrocodone/acetaminophen, oxycodone hcl/acetaminophen
Orthopaedic surgery	oxycodone hcl, hydrocodone/acetaminophen, oxycodone hcl/acetaminophen
Orthopedic surgery	oxycodone hcl, hydrocodone/acetaminophen, tramadol hcl
Otolaryngology	oxycodone hcl, fluticasone propionate, oxycodone hcl/acetaminophen
Pain management	oxycodone hcl, hydrocodone/acetaminophen, oxycodone hcl/acetaminophen
Physical medicine and rehabilitation	oxycodone hcl, hydrocodone/acetaminophen, gabapentin
Physician assistant	oxycodone hcl, hydrocodone/acetaminophen, oxycodone hcl/acetaminophen
Plastic and reconstructive surgery	oxycodone hcl, hydrocodone/acetaminophen, cephalexin
Podiatry	oxycodone hcl, hydrocodone/acetaminophen, oxycodone hcl/acetaminophen
Radiation oncology	oxycodone hcl, tamsulosin hcl, lidocaine hcl
Student	oxycodone hcl, hydrocodone/acetaminophen, oxycodone hcl/acetaminophen



Cluster	Procedure	Count
0	CT ABDOMEN W/CONTRAST	6393
0	CT CHEST W/CONTRAST	5350
0	CT ABDOMEN W/O CONTRAST	3892
1	CHEST (PORTABLE AP)	13010
1	CHEST (PA & LAT)	611
1	CHEST (SINGLE VIEW)	108
2	CHEST (PORTABLE AP)	11938
2	CHEST (PA & LAT)	1693
2	CT CHEST W/O CONTRAST	857
3	CHEST (PORTABLE AP)	14512
3	CHEST PORT. LINE PLACEMENT	1725
3	TRAUMA #3 (PORT CHEST ONLY)	217
4	BABYGRAM (CHEST ONLY)	3117
4	NEONATAL HEAD PORTABLE	2782
4	P BABYGRAM (CHEST ONLY) PORT	1609
5	LIVER OR GALLBLADDER US (SINGLE ORGAN)	4522
5	ABDOMEN U.S. (COMPLETE STUDY)	2102
6	CT C-SPINE W/O CONTRAST	4632
6	T-SPINE	1091
6	L-SPINE (AP & LAT)	937

Figure 5: Left: multiple clustering outputs ($C = 2$ to $C = 20$) from a single representation, with y-axis indicating samples ordered by ground truth cluster membership each of size 10, and x-axis indicating the number of clusters found divided by the number available to the algorithm, and color indicating cluster membership within column. Right: Top three radiology descriptions per cluster (first seven) with procedure counts based on MIMIC radiology notes.

illustrates the predicted cluster assignments for different C along the x axis, with colors for a given C determining co-cluster membership and with colors for different C learned to be close to the previous cluster color but far from other cluster colors. The figure demonstrates recovery of the ground truth clusters (10 of them) and also provides clusterings identified when $C \neq 10$, noting that the algorithm may elect to place 0 members in a cluster if the data warrants it.

Characterizing radiology notes. In our final case study, we sought to organize a large collection of radiology notes based on the notes' contents, for individuals with critical care needs. We used de-identified radiology notes from MIMIC III v1.4[13], a natural language processing pipeline, and a bag of words representation to cluster radiology notes. The note descriptions provided are specific in some senses, *e.g.* anatomical: ”X-ray of left foot 4th digit, two views”, but nonspecific in other senses, *e.g.* by indication: “CT chest w/o contrast” for cardiac, pulmonary, or gastrointestinal disease, or something else? After stop word removal, stemming, and lemmatization, the bag-of-word representation resulted in a sparse matrix of size: 522,279 notes by 275,263 unique word tokens. We performed clustering with 30 clusters. The result was a meaningful clustering, with for example, top members of distinct clusters including: kidney ultrasounds, infant radiographs, and abdominal ultrasounds. The cluster list is shown in Figure 5 (right).

Conclusion

In this paper, we defined a deep clustering method to address this task of clustering high-dimensional sparse count data, as often arises in analysis of healthcare data and other fields. To do so, we described an autoencoder architecture that takes into account not just image recovery and clustering quality, but also geometric relationships in the data. We demonstrated that this approach works well through quantitative comparisons with other methods, as well as through detailed case studies on real healthcare data, including the development of an informatics tool that enables exploratory analysis of providers based on their clinical behaviors.

References

[1] Chris Ding and Xiaofeng He. K-means clustering via principal component analysis. *Proceedings of the Twenty-first International Conference on Machine Learning*, 2004.

[2] Elie Aljalbout, Vladimir Golkov, Yawar Siddiqui, and Daniel Cremers. Clustering with deep learning: Taxonomy and new methods. *CoRR*, abs/1801.07648, 2018.

[3] Fei Tian, Bin Gao, Qing Cui, Enhong Chen, and Tie-Yan Liu. Learning deep representations for graph clustering. *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[4] George Trigeorgis, Konstantinos Bousmalisand Stefanos Zafeiriou, and Bjorn W. Schuller. A deep semi-nmf model for learning hidden representations. *International Conference on Machine Learning*, 2014.

[5] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. *ICML*, 2016.

[6] Bo Yang, Xiao Fu, Nicholas D. Sidiropoulos, and Mingyi Hong. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. *ICML*, 2017.

[7] Asa Ben-Hur, David Horn, Hava T Siegelmann, and Vladimir Vapnik. Support vector clustering. *Journal of machine learning research*, 2(Dec):125–137, 2001.

[8] Yuan Ping, Yun Feng Chang, Yajian Zhou, Ying Jie Tian, Yi Xian Yang, and Zhili Zhang. Fast and scalable support vector clustering for large-scale data analysis. *Knowledge and Information Systems*, 2015.

[9] Chin-Wei Huang, David Krueger, Alexandre Lacoste, and Aaron Courville. Neural autoregressive flows. *Proceedings of Machine Learning Research: International Conference on Machine Learning*, 2018.

[10] Medicare. *Medicare Provider Utilization and Payment Data: Part D Prescriber Summary Table CY2015 (Accessed April 30, 2018)*, 2017.

[11] Christopher W Seymour, Jason N Kennedy, Shu Wang, Chung-Chou H Chang, Corrine F Elliott, Zhongying Xu, Scott Berry, Gilles Clermont, Gregory Cooper, Hernando Gomez, et al. Derivation, validation, and potential treatment implications of novel clinical phenotypes for sepsis. *Journal of the American Medical Association*, 321(20):2003–2017, 2019.

[12] Emma Ahlqvist, Petter Storm, Annemari Käräjämäki, Mats Martinell, Mozhgan Dorkhan, Annelie Carlsson, Petter Vikman, Rashmi B Prasad, Dina Mansour Aly, Peter Almgren, et al. Novel subgroups of adult-onset diabetes and their association with outcomes: a data-driven cluster analysis of six variables. *The lancet Diabetes & endocrinology*, 6(5):361–369, 2018.

[13] Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3:160035, 2016.