Bicluster phenotyping of healthcare providers, procedures, and prescriptions with deep learning

Nathanael Fillmore[1], Ansh K. Mehta[2], and Jeremy C. Weiss[3]

[1] Department of Veterans Affairs & Dana-Farber Cancer Institue, Boston, MA, USA
[2] University of South Alabama Hospitals, Mobile, AL, USA
[3] Carnegie Mellon University, Heinz College, Pittsburgh, PA, USA

Abstract. We consider the task of biclustering healthcare providers ($\approx 10^6$), procedures ($\approx 10^4$), and drugs ($\approx 10^3$) based on 2015 counts of claims by provider. The problem with many clustering methods is the scale of the data and the sparse count data where the notion of "distance" is unclear. We use neural networks, which are frequently used for representation learning, to represent the sparse high-dimensional information with a low-dimensional embedding. While previous work has performed standard clustering methods in the embedding space, we instead leverage a low-dimensional embedding with geometric constraints that enables simultaneous image recovery and cluster determination.

1 Introduction

We consider the task of biclustering healthcare providers ($\approx 10^6$), procedures ($\approx 10^4$), and drugs ($\approx 10^3$). Standard ontologies exist that group medical providers, procedures, and prescriptions into categories, but these categories do not always correspond to clusters that are relevant for every question of interest, and some standard categories, like Internal Medicine, are quite broad. Instead, we would like a tool that captures practice characteristics de-novo to compare against collected measures and ontologies. This has several applications: we can measure concordance of the existing hierarchies with those identified in data, we can look at changes in practice patterns over time, we can investigate differences in practice within and across nominal subgroups based on practice and utilization, and we can investigate co-occurrence.

Many clustering methods perform poorly on this task because of the scale of the data and because, in sparse count data, the most appropriate notion of "distance" is not entirely clear. Additionally, centroid-based clustering in our sparse count data space is problematic because identifying an appropriate centroid location is non-obvious. Using the (possibly weighted) average value of members assumes an L2 space which is problematic for high dimensional, sparse count data. Moreover, it is well known that distances tend to be similar in high dimensions, and this problem is only made worse by sparse count data.

In this work, we describe a method that learns a low-dimensional embedding with geometric constraints that enables simultaneous image recovery and cluster determination. We propose a generic architecture that simultaneously

(1) optimizes image recovery in an autoencoding framework, (2) creates a low-dimensional embedded representation of the high-dimensional space, (3) assigns examples to clusters in a soft clustering, and (4) optimizes the quality of this clustering.

Several authors have recently proposed methods that are conceptually similar to our contribution. For example, [5] also formulates a method that integrates a clustering module in a deep learning framework, as does [4]. However, both these methods are based on identifying centroids in the latent space, which is not appropriate for our data. Other related work includes methods to learn metric embeddings with neural networks, e.g., [2], but as these methods do not incorporate a clustering objective in the neural network, the learned embedding is not necessarily suitable for clustering.

In our application, cosine similarity is appropriate, so we use it in our examples. Many healthcare datasets consist of sparse count data in high-dimensional spaces as a consequence of counting visits, billing codes, procedure codes, drug prescriptions, etc. Our tool is well-suited to data of this nature.

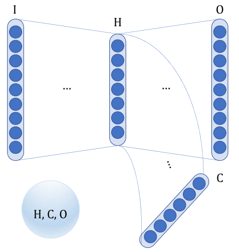

Fig. 1. Autoencoder with clustering module. Nodes are Input (I), Output (O), Hidden (H), and Cluster (C). H and O are constrained by batch cosine similarity. H (h°) and C are constrained by batch cosine similarity.

2 Method

Consider a sparse count matrix with N rows and K columns. Define M an $N \times K$ as a matrix with elements log transformed using the function $f(x) = \log(1+x)$. We are interested in clustering over rows (providers) and columns (prescriptions and procedures). We will consider clustering across rows and columns separately. For each case, we construct an autoencoder consisting of an encoder Ψ_1 and a decoder Ψ_2. Define the hidden representation h of size H such that $\Psi_1(M) = h$ and $\hat{M}$ as the reconstruction given by $\Psi_2 \circ \Psi_1(M) = \hat{M}$. Define Ψ_3 as the function transformation for the clustering module that takes as input h and outputs cluster probabilities c: $\Psi_3(h) = c$. Fig. 1 illustrates the framework for the neural network.

For log-transformed sparse count data, cosine similarity is a useful distance representation. However, computing the cosine similarity for all pairs may be problematic when N is large: $O(N^2)$. We propose to learn a hidden representation $h = \{h^{\circ}, h^{||\cdot||}\}$ where the cosine similarity between examples in M is approximately preserved in h° of size $H-1$ units and where the final unit $h^{||\cdot||}$ captures information in the norm. To achieve this, we define batch cosine similarity ($O(B^2)$) and its loss $\mathcal{L}_{h^{\circ},O}$ as a penalization to the autoencoder loss.

Define batch size B such that h_B and $\hat{M}_B$ are of size $B \times \{\cdot\}$ for H and M respectively. Let $\delta(x_i, x_j)$ be the pairwise cosine similarity. Define δ_B the batch cosine similarity, *i.e.* $\delta_B(\{x_1, \ldots, x_B\}) = [\delta(x_i, x_j)]_{ij}\ \forall i, j \in \{1, \ldots, B\}$. Then, $\mathcal{L}_{h^{\circ},O} = \frac{1}{B}||\delta_B(h_B^{\circ}) - \delta_B(\hat{M}_B)||_2^2$.

While $\mathcal{L}_{h^\circ,O}$ encourages matching angles in h° and $\hat{M}$, two vectors in h° could have the same cosine angle but different embedding locations. Note this could be problematic because the hidden vectors are used to learn cluster membership probabilities c. 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.

Layer	Index	Size	Act.
(Input)	–	$B \times K$	
Dense	0	$B \times H$	LReLU
Haar wavelet	–	$B \times H$	–
Expand, permute	–	$B \times H \times P$	–
Pool	1	$B \times H$	–
Sum (0) and (1)	–	$B \times H$	–
Batch norm	–	$B \times H$	–
Dense	–	$B \times H$	LReLU
Batch norm	–	$B \times H$	–
Dense	–	$B \times H$	LReLU
Batch norm	–	$B \times H$	–
Dense	–	$B \times H$	LReLU
Dense	–	$B \times H$	LReLU
(Hidden)	2	$B \times H$	
Dense	–	$B \times H$	LReLU
Dense	–	$B \times H$	LReLU
Dense	–	$B \times K$	LReLU
(Output)	–	$B \times K$	
Copy (2)	–	$B \times H$	–
Dense	–	$B \times H$	LReLU
Dense	–	$B \times H$	LReLU
Dense	–	$B \times C$	Softmax
(Cluster)	–	$B \times C$	

Table 1. Deep network architecture

The hidden representation h° preserves angular distances of the output space, and its unit norm and approximate injectivity are useful for our clustering. 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, 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 set of probability vectors indicating cluster membership. Note that the cosine similarity of ${c_B}^{\frac{1}{2}}$ is non-negative and we are not interested in incurring loss due to differences between 0 and negative cosine similarities. We define $\mathcal{L}_{c,h^\circ} = \frac{1}{B}||\delta_B({c_B}^{\frac{1}{2}}) - \max(\delta_B(h_B^\circ), 0)||_2^2$.

We add two more loss terms to assist representation learning: h spread and c entropy. For level sets of batch cosine loss, we prefer that embedded vectors in h be spread apart to minimize coincidental overlap for suboptimally located members of h and for future unseen points from the underlying distribution. We also impose an entropy loss term to encourage cluster probabilities to be spread across more than one cluster to encourage exploration in cluster membership and avoid local optima. Thus, our overall objective function is $\sum_i \lambda_i \mathcal{L}_i$ for $\mathcal{L}_i \in \{\mathcal{L}_{M,\hat{M}}, \mathcal{L}_{||h^\circ||=1}, \mathcal{L}_{h^\circ,O}, \mathcal{L}_{c,h^\circ}, \mathcal{L}_{\text{spread}}, \mathcal{L}_{\text{entropy}}\}$. We set λ_i respectively: $\lambda_i \in \{1, 10^{-1}, 10^{-1}, 1, 10^{-4}, 10^{-4}\}$.

The autoencoder framework we adopt is shown in Table 1. The permute-pool layer [3] copies the tensor P times, permutes the values, and performs max-pooling over the P dimension with size 3 and stride 2. For 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 merge the clusters identified based on maximum pairwise cluster cosine similarity.

3 Results

Evaluation on simulated data. We evaluate our method on real and simulated data. First, on simulated data generated so as to be similar to our target healthcare application, we show that our method typically finds a clus-

	Rand			Jaccard			Recall			Precision		
Data	Ours	OKM	HKM	Ours	OKM	HKM	Ours	OKM	HKM	Ours	OKM	HKM
baseline	**1.00**	0.84	1.00	**1.00**	0.12	0.99	**1.00**	0.14	1.00	**1.00**	0.58	1.00
samples=100	**1.00**	0.85	0.91	**0.90**	0.11	0.17	**0.90**	0.12	0.19	**1.00**	0.60	0.65
dims=100000	**0.93**	0.21	0.48	**0.08**	0.04	0.04	**0.14**	0.04	0.05	0.17	**0.87**	0.62
explode=1000000	**1.00**	0.84	**1.00**	**1.00**	0.13	**1.00**	**1.00**	0.14	**1.00**	**1.00**	0.59	**1.00**
centroids=100	0.99	0.81	**1.00**	0.47	0.03	**0.95**	0.47	0.03	**0.96**	**1.00**	0.64	0.99
sd=0.1	**0.93**	0.28	0.78	**0.08**	0.04	0.05	**0.15**	0.04	0.06	0.15	**0.79**	0.29

Table 2. 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, sd=0.01, explode=10000, and varied in each dataset as indicated. Data are clustered using our method (Ours), k-means in the original, input space (OKM), and k-means in the hidden, embedded space we construct (HKM). The best score is in boldface. Our method almost always outperforms k-means in the original space, and usually outperforms k-means in the hidden space.

tering 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 $\mathcal{N}(\mathbf{0}, \mathbf{1})$ and normalized onto the unit ball. 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 $[1, \kappa]$. 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.

We cluster the data using our method, k-means in the original simulated data space, and k-means in the hidden space. We evaluate results of these clustering methods using precision, recall, the Rand index, and the Jaccard index, all relative to the ground truth labels from the simulation. Results are shown in Table 2. By these measures, our method almost always outperforms k-means in the original space, and usually outperforms k-means in the hidden space.

Evaluation on real healthcare data. As a case study to demonstrate our method's real world significance, we used our method to bicluster healthcare providers, prescriptions, and procedures based on the number of prescriptions and procedures administered by each provider, and the number of providers administering each prescription or procedure as recorded.

Specifically, we used Medicare Provider Utilization and Payment Data: Part D Prescriber Summary Table CY2015 [1], which tabulates all prescriptions given under the Medicare Part D program in 2015 in the United States. In the interest of space, we only show results for a clustering of providers based on the prescriptions they gave. Our physician authors assessed the quality of the resulting clusters.

The clusterings formed by our method and by k-means, each with 20 clusters, are shown in Table 3. Our method produces qualitatively better clusters than k-means does. For example, our clustering consistently includes a larger fraction of specialists in the specialist cluster, *e.g.*, Dentist (85k), Psychiatry (21k), Emergency Medicine (20k). Our clustering, unlike k-means, also identifies clean obstetrics and hematology oncology clusters. K-means identifies a cardiology and interventional cardiology cluster that our method does not, however our clustering identified this cluster and merged it (Cardiology (18k), Nurse Prac (3k), Internal Medicine (2k)) into the internal medicine subgroup in post-processing.

Our method also provides insight in regard to providers in ontology 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. On 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 radiation therapy related complications. 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. Future work will involve exploration of comparisons with other scalable clustering methods and of pertinence to clinical and workforce questions, including nurse practitioner and physician assistant implicit specialty characterization and opioid prescription networks.

(a) Our Clustering (Top 3 Specialties)
Dentist (85k); Oral Surgery (3k); Podiatry (1k)
Internal Med (82k); Family Practice (82k); Nurse Prac (44k)
Psychiatry (21k); Nurse Prac (9k); Psychiatry & Neurology (6k)
Emergency Medicine (21k); Orthopedic Surgery (15k); Phys Asst (15k)
Optometry (18k); Ophthalmology (17k); Student (<1k)
Obstetrics/Gynecology (17k); Nurse Prac (2k); Phys Asst (<1k)
Gastroenterology (12k); Nurse Prac (3k); Internal Med (3k)
Dermatology (10k); Phys Asst (3k); Nurse Prac (1k)
Neurology (10k); Nurse Prac (1k); Psychiatry & Neurology (1k)
Urology (9k); Phys Asst (1k); Nurse Prac (1k)
Nurse Prac (8k); Phys Asst (6k); Emergency Medicine (6k)
Pulmonary Disease (7k); Allergy/Immunology (3k); Otolaryngology (2k)
Hematology/Oncology (6k); Nurse Prac (2k); Medical Oncology (2k)
Internal Med (5k); Emergency Medicine (3k); Nurse Prac (2k)
Phys Asst (4k); Nurse Prac (4k); Orthopedic Surgery (2k)
Dentist (3k); Emergency Medicine (2k); Phys Asst (2k)
Infectious Disease (3k); Obstetrics/Gynecology (2k); Nurse Prac (2k)
Pharmacist (2k); Nurse Prac (1k); Internal Med (1k)
Podiatry (2k); Nurse Prac (1k); Optometry (1k)
Physical Med/Rehab (2k); Podiatry (2k); Nurse Prac (2k)

(b) K-Means Clustering (Top 3 Specialties)
Dentist (52k); Nurse Prac (1k); Phys Asst (1k)
Nurse Prac (35k); Phys Asst (27k); Internal Med (27k)
Family Practice (23k); Internal Med (17k); Nurse Prac (10k)
Family Practice (22k); Internal Med (20k); Nurse Prac (4k)
Emergency Medicine (20k); Orthopedic Surgery (13k); Phys Asst (12k)
Dentist (19k); Oral Surgery (3k); Maxillofacial Surgery (1k)
Internal Med (16k); Nurse Prac (12k); Family Practice (9k)
Family Practice (16k); Nurse Prac (13k); Internal Med (11k)
Cardiology (14k); Nurse Prac (1k); Interventional Cardiology (1k)
Dentist (14k); Oral Surgery (<1k); Infectious Disease (<1k)
Optometry (12k); Ophthalmology (5k); Student (<1k)
Ophthalmology (11k); Optometry (2k); Student (<1k)
Internal Med (11k); Family Practice (10k); General Practice (1k)
Psychiatry (10k); Nurse Prac (3k); Psychiatry & Neurology (1k)
Psychiatry (8k); Psychiatry & Neurology (3k); Nurse Prac (3k)
Urology (8k); Phys Asst (1k); Nurse Prac (1k)
Neurology (7k); Nurse Prac (<1k); Phys Asst (<1k)
Pulmonary Disease (6k); Allergy/Immunology (2k); Otolaryngology (1k)
Neurology (3k); Nurse Prac (1k); Physical Med/Rehab (1k)
Rheumatology (2k); Physical Med/Rehab (2k); Nurse Prac (2k)

Table 3. Provider clustering by (a) our method, (b) k-means; clusters (rows) with top specialties (number).

References

1. Medicare Provider Utilization and Payment Data: Part D Prescriber Summary Table CY2015 (2017 (accessed April 30, 2018)), https://data.cms.gov/Medicare-Part-D/Medicare-Provider-Utilization-and-Payment-Data-Par/qywy-pajd
2. Song, H.O., Jegelka, S., Rathod, V., Murphy, K.: Deep metric learning via facility location. CVPR (2017)
3. Weiss, J.C.: Machine learning for clinical risk: wavelet reconstruction networks for marked point processes. working paper (2018)
4. Xie, J., Girshick, R., Farhadi, A.: Unsupervised deep embedding for clustering analysis. ICML (2016)
5. Yang, B., Fu, X., Sidiropoulos, N.D., Hong, M.: Towards k-means-friendly spaces: Simultaneous deep learning and clustering. ICML (2017)