CS230

The Latent Shape of Image Data: Anomaly Detection with Deep Clustering

Eric J. Wang
Department of Computer Science
Stanford University
ejwang@stanford.edu

Nancy Xu
Department of Computer Science
Stanford University
xnancy@stanford.edu

Abstract

Our research works toward a robust data pipeline for crowdsourced and decentralized high-dimensional data that is capable of filtering out irrelevant entries. We use deep convolutional networks to process image data, and apply a number of probabilistic modeling and dimensionality reduction techniques on artificially contaminated image datasets to identify anomalies. Using deep learning approaches, we were able to learn the conceptual boundaries between distributions relatively easily in combination with basic statistical methods. Our research suggests an algorithm for the sanitization of image data for applications like Google Maps or Yelp, as well as possible applications to detecting erroneous data in large training sets.

1 Introduction

A large number of applications today rely on crowdsourced image data to illustrate a topic. Google Maps, for instance, crowdsources photos from different locations in order to display them in their app. Yelp also relies on user-submitted image data to illustrate the interiors of websites. This user-submitted data is often contaminated by unrelated images that need to be manually removed. However, it is difficult and time-consuming to discriminate between valid and invalid user-submitted images, especially without prior knowledge of what the location looks like. The goal is to develop unsupervised end-to-end pipeline that, given no knowledge of what the dataset is about, maximizes for sensitivity (proportion of anomalous images labeled as anomalous) and secondarily recall (proportion of target images labeled as target images) in anomaly detection.

We characterize a convolutional neural net's training process as the learning of a latent feature space $\Psi(\mathbb{R}^{n\times n\times k})$ (induced by the activations of the penultimate layer $\Psi(x)$) in which the training categories are linearly separable. This property of the space generalizes fairly well to other categories—as demonstrated by the effectiveness of transfer learning methods.

As a result, we believe that using pre-trained vectors provides a good heuristic for unsupervised classification models, as we redirect learning to this latent space on which we can apply simple statistical methods.

1.1 Probabilistic Model

This motivates us to model our dataset as a probabilistic mixture model: say that the latent representation is drawn from one of two Gaussian distributions in the latent space, the *theme* distribution and the *anomaly* distribution. In our mixture model, the distribution is sampled according to a binary

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

variable $z \sim \text{Bernoulli}(\phi)$; the distribution is sampled from z, the sample is sampled from either the theme distribution or the anomaly distribution:

$$\Psi(x^{(i)}) \sim \begin{cases} N(\mu_a, \Sigma_a) & \text{if } z = 0 \\ N(\mu_t, \Sigma_t) & \text{if } z = 1 \end{cases}$$

This applies to $N(\mu_t, \Sigma_t)$ or $N(\mu_a, \Sigma_a)$.

We know three facts about these distributions:

1. $|\Sigma_a|$, the generalized variance of the anomaly distribution, is almost certainly greater than $|\Sigma_t|$, the generalized variance of the "relevant" distribution.
2. If we can take a uniformly random selection of images in the dataset and "verify" a number of them as relevant, we can treat these as samples from $N(\mu_t, \Sigma_t)$ and establish normal and χ^2 priors on μ_t and Σ_t respectively.
3. We have a reasonably accurate estimate of ϕ, or the "cleanliness" of the overall dataset, which is significantly larger than 0.5.

We probe three questions in this paper:

- Are distributions of images in the latent space actually simple enough to be modeled with these assumptions?
- Are these priors, especially (3), sufficient to identify the correct clusters?
- How can we approach this estimation in a computationally feasible way?

2 Prior Work

Prior work in this area has focused on two primary directions: 1.) clustering algorithms for segmentation of high-dimensional data [12] [11] and 2.) unsupervised image processing techniques (ex. unsupervised image segmentation) [6]. In this paper, we combine work in the deep learning space from the latter category and use unsupervised methods from the former to develop an approach to unsupervised bi-modal Guassian image clustering on high-dimensional image data.

In the past decades, a large family of clustering methods have been proposed to partition data points into clusters based on similarity measurements, while ensuring maximal dissimilarity among classes. Many of these clustering techniques have been successfully used for image clustering in applications such as content based image retrieval (CBIR), image indexing, and image annotation [2].

However, many of these approaches rely on classical statistical techniques that operate on lower-order optimization parameters. For example, graph based manifold learning methods have shown promising performance in extracting features for image clustering [12], along with discrete and continuous image models using information-theoretic-based criteria for unsupervised hierarchical image-set clustering [5]. These continuous image models are based on a mixture of Gaussian densities, while the unsupervised clustering model is based on a generalized version of the information bottleneck principle. In both cases, images are clustered such that the mutual information between the clusters and the image content is maximally retained.

In the case of bi-Gaussian anomaly predictions, we are concerned primarily with preservation of image content vs. retention of similarity magnitude between images. Furthermore, we want a model that is fully unsupervised and computational efficient that performs well on small datasets. As a result, pre-trained deep learning networks seem promising for providing vectorization of image input for distribution-sensitive dimensionality reduction techniques. Instead of modeling a direct statistical manifold to the dataset, we develop a deep learning pipeline that vectorizes using pre-trained image models nad performs dimensionality reduction, before performing continuous Gaussian clustering as in some prior work.

This allows us to analyze data in a significantly reduced vector space that takes into account distribution densities while also rapidly identifying key image features that relate to content (due to pre-training). The model of choice is based on Inception v4, introduced in 2016 by Szegedy et al. [10], for the classification of data from the ImageNet challenge [9]. A number of applications have shown that using an Inception v4 model pretrained on ImageNet produces latent space representations in the final layers of the network that are good vectorizations for a variety of image applications [4][3]. As such, we choose to use the Inception v4 architecture as the image vectorization basis of our methods.

3 Dataset and Features

We create mixtures of target and anomaly data using the Food-101 dataset and ImageNet. More specifically, we produce three data categories:

- The ***theme-pie*** dataset, consisting of a large number of pictures of apple pie;
- The ***theme-sushi*** dataset, consisting of pictures of sushi
- The ***anomaly*** dataset, consisting of a completely random selection of 900 images scraped from ImageNet.

We experiment on three mixtures, which mix the above categories. Two mixtures have a target (sushi / pie) mixed with random noise from ImageNet and one has a target (pie) and clustered random noise (sushi). Each dataset image's values are pre-normalized to $[0, 1]$, and resized such that the largest square central crop is 299×299 with bicubic interpolation. Some images were not RGB-encoded, so we replaced them. We then evaluated ϕ on each of these 3000 images to generate a set of 98304-dimensional vectors in three different CSV files.

The mixtures were then created at the following proportions:

	Target	**Anomaly**	**# Targets**	**# Anomalies**
Mixture 1	Pie	ImageNet	1000	333
Mixture 2	Sushi	ImageNet	1000	333
Mixture 3	Pie	Sushi	1000	333

Figure 1: Simulation mixtures from ImageNet & Food-101

To simulate the three scenarios, we wrote a script that shuffled each of the three lists and randomly selected images up to a threshold. In each scenario, we simulated a relatively high amount of noise ($\phi = .75$) by mixing the datasets in unequal proportions.

4 Methods

For our map, we used a TensorFlow[1] implementation of the Inception-v4 architecture. The network weights were trained on the ImageNet ILSVRC 2012 dataset, and the last layer was removed so that the network represented the map $\Psi : \mathbb{R}^{299\times299\times3} \longrightarrow \mathbb{R}_+^{16\times16\times384}$. (As the model uses ReLU activation, the latent space contains only nonnegative components.)

Because of the memory issues involved in mixture modeling in $\mathbb{R}^{98304}$, we experimented with four data-sensitive dimensionality reduction methods, moving from the least computationally expensive to the most: sparse random projections, principal component analysis, sparse PCA, and autoencoders. Our pipeline applied each method to each of the three simulation.

Sparse random projections: The first dimensionality reduction method we tried. SRPs are a computationally efficient way to perform dimensionality reduction by generating a sparse, nearly-orthogonal matrix and projecting the data with it. We generate this matrix according to the parameters recommended in Li et al. [7], and reduce the vectors to 5000 low-information vectors.

Principal component analysis: A standard dimensionality reduction technique, significantly more computationally intensive than sparse random projection. Iteratively generates an orthogonal matrix of height $\leq d$ by choosing orthogonal unit projection vectors that maximize the variance. Early versions of the pipeline used an approximate PCA algorithm that sampled smaller parts of the dataset; with increases in computing power, we were eventually able to generate around 500 principal components.

	Vector Embedding Size
Sparse Random Projection	5000
PCA	500
Sparse PCA	500
Auto-encoder	100

Figure 2: Dimensionality reduction sizes for clustering techniques

Sparse PCA: A more recent technique pioneered by Zou et al. [13] that applies PCA with the variance objective regularized by the elastic net (a linear combination of L_1/LASSO and L_2 penalties). Popular for dimensionality reduction because of L_1 regularization's sparsity; this increases the semantic closeness of the data to the original input. Takes much longer than PCA; minibatch approximations necessary.

Autoencoders: After promising results from the 500-dimensional representations, we wanted to push the compression even further by training an autoencoder network. An autoencoder takes a latent vector as input, forces it through several lower-dimensional layers, and expects the original latent vector as output. By doing so, the bottleneck layer becomes a compact representation of the data. (It's worth noting that there's only a minor difference between PCA and a one-layer autoencoder network, and PCA is already "most of the way" to an autoencoder in a certain sense.) Autoencoders would also enable us to generalize to more complex classification boundaries. The training ultimately took longer than expected.

It is important that each method (besides sparse random projections) is run on the mixed scenario files, rather than the individual datasets; this is because these methods are heavily dataset-dependent.

For each file, we sampled a proportional number of each class of vectors into 100-vector test sets, then trained a Gaussian mixture model from the scikit-learn library[8] on the remaining vectors, emulating our prior for the theme/target mean by initializing the means to sample means of 20 randomly sampled target images and the weights to the known ϕ. The data labels were unknown to the algorithm. Our pipeline ran this code with different numbers of projected components and output the resulting sensitivity and recall statistics to a text file.

5 Experiments/Results/Discussion

Technique	**Sensitivity**	**Recall**
Sparse Random Projection	99.7%	66.4%
PCA	99.7%	85.0%
Sparse PCA	82.1%	62.4%

Figure 3: Performance metrics for dimensionality reduction techniques, averaged over scenarios

As our experiments progressed, we tried dimensionality reduction techniques in increasing order of computational intensity and decreasing order of the resulting dimension. Sparse random projections were able to capture a lot of the variance within the latent space, but only using a very high-dimensional representation. Sparse PCA, which we hypothesized would be useful because of the sparsity and semantic importance of ReLU data, actually performed unilaterally worse than the other models. Regular PCA outperformed sparse random projections significantly on recall and achieved comparable sensitivity.

Note that these figures only represent effectiveness if we use 50% as our threshold of prediction. By only accepting examples with a target probability of, say, 70%, we could increase the sensitivity with corresponding cost to recall. This might make sense in terms of user-facing applications like Google Maps, but could also harm the amount and the distribution of experimental training data in data science contexts.

Our explanation for the high performance of the PCA algorithm was that it accomplished two things: from an ANOVA standpoint, it isolated axes that maximized the variance *within* the largest dataset, and also maximized the variance *between* the clusters.

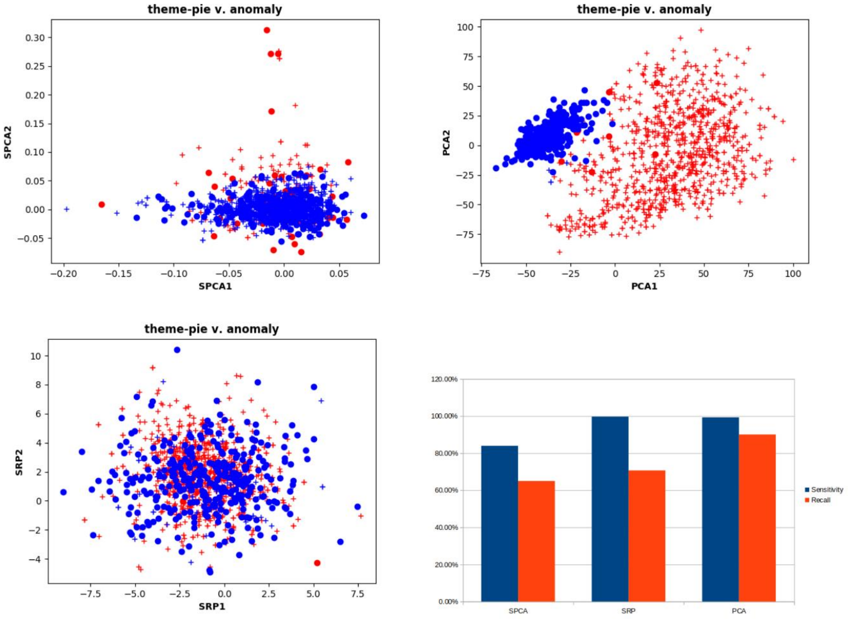

Figure 4: Classification plots of one scenario over the first two projection vectors. Shape represents data source; color represents estimated cluster.

6 Conclusion/Future Work

Our promising results demonstrate the feasibility of an automated anomaly detection system, but more work needs to be done to replicate our results on other datasets and on other dimensionality reduction techniques.

With more time, we would implement the EM algorithm for the full distribution with the sampling priors detailed in Section 1.1 and finish the autoencoder representation learner. We strongly suspect that the right combination of dimensionality reduction and Bayesian Gaussian mixture modeling would be able to yield a robust algorithm for separating distributions in the latent space with linear (or even non-linear, in the autoencoder case) classification boundaries.

7 Contributions

Both team members contributed equally to all parts of the project ideation, development, and final presentation including: creating initial datasets, training networks, building data pipelines, testing on various methods, as well as mathematical formulation of the problem statement. Contribution to the final poster and this paper was also shared evenly and done together.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] R. Datta, D. Joshi, J. Li, and J. Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys (Csur)*, 40(2):5, 2008.

[3] S. Diamond, V. Sitzmann, S. Boyd, G. Wetzstein, and F. Heide. Dirty pixels: Optimizing image classification architectures for raw sensor data. *arXiv preprint arXiv:1701.06487*, 2017.

[4] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115, 2017.

[5] J. Goldberger, S. Gordon, and H. Greenspan. Unsupervised image-set clustering using an information theoretic framework. *IEEE transactions on image processing*, 15(2):449–458, 2006.

[6] E. Kim, H. Li, and X. Huang. A hierarchical image clustering cosegmentation framework. pages 686–693, 2012.

[7] P. Li, T. J. Hastie, and K. W. Church. Very sparse random projections. pages 287–296, 2006.

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[9] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[10] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. 4:12, 2017.

[11] Y. Yang, D. Xu, F. Nie, S. Yan, and Y. Zhuang. Image clustering using local discriminant models and global integration. *IEEE Transactions on Image Processing*, 19(10):2761–2773, 2010.

[12] J. Yu, R. Hong, M. Wang, and J. You. Image clustering based on sparse patch alignment framework. *Pattern Recognition*, 47(11):3512–3519, 2014.

[13] H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *Journal of computational and graphical statistics*, 15(2):265–286, 2006.

GITHUB: https://github.com/xnancy/anomaly