Sparse Coding for Classification via a Locality Regularizer with Applications to Agriculture

Lukai Li[1], James M. Murphy[1], Abiy Tasissa[1]

[1] Department of Mathematics, Tufts University, Medford, MA 02155, USA.

A paper from the Proceedings of the
16th International Conference on Precision Agriculture
21-24 July 2024
Manhattan, Kansas, United States

Abstract.
High-dimensional data is commonly encountered in various applications, including genomics, as well as image and video processing. Analyzing, computing, and visualizing such data pose significant challenges. Feature extraction methods are crucial in addressing these challenges by obtaining compressed representations that are suitable for analysis and downstream tasks. One effective technique along these lines is sparse coding, which involves representing data as a sparse linear combination of a set of exemplars. In this study, we propose a local sparse coding framework within the context of a classification problem. The objective is to predict the label of a given data point based on labeled training data. The primary optimization problem encourages the representation of each data point using nearby exemplars. We leverage the optimized sparse representation coefficients to predict the label of a test data point by assessing its similarity to the sparse representations of the training data. The proposed framework is computationally efficient and provides interpretable sparse representations. To illustrate the practicality of our proposed framework, we apply it to agriculture for the classification of crop diseases.

Keywords.
Classification, Sparse coding, Crop disease identification, Feature extraction.

Introduction

Agriculture is an essential activity that supplies the basic necessities required for our survival. In addition, in many middle-income and low-income countries, agriculture is a foundational aspect of the national economy (Alston et al., 2014). The agricultural sector faces several challenges, with their effects being more pronounced in less-developed countries. These challenges include climate change, pests, crop diseases, and a growing population (Calicioglu et al., 2019). This paper focuses on crop diseases due to their critical impact on food security. Identifying and managing crop diseases is a central issue in agriculture, requiring expert knowledge and labor (Chaube et al., 2005). In recent years, signal processing and machine learning techniques have been used to assist experts in the time and cost intensive process of disease identification (Ferentinos et al., 2018, Haridasan et al., 2023, Shruti et al., 2019).

We consider the setting where we have existing training data consisting of labeled examples of healthy and diseased plants. Using this data, our goal is to predict whether a new plant from the same category is healthy or not. The training data is assumed to be a collection of images, such as plant leaf images. Given this setup, a typical initial step is to use established image processing tools to extract useful features or obtain compressed representations that aid in disease identification (Antonini et al., 1992, Lowe et al., 1999, Nixon et al., 2019). Recently, machine learning and deep learning techniques have been widely used for image classification, largely due to the availability of large amounts of training data (Krizhevsky et al., 2017, Sheykhmousa et al., 2020). In this paper, we propose a local sparse coding approach for plant disease identification. Sparse coding is a signal processing framework where data points are represented as a combination of a few underlying components, known as atoms, which form a dictionary (Aharon et al., 2006, Olshausen et al., 1990). Specifically, the $\boldsymbol{n}$ data points are represented as $\boldsymbol{y_1}, \boldsymbol{y_2}, \dots, \boldsymbol{y_n}$, with $\boldsymbol{A} = [\boldsymbol{a_1}, \boldsymbol{a_2}, \dots \boldsymbol{a_m}]$ denoting the dictionary composed of $\boldsymbol{m}$ atoms. The goal of sparse coding is to represent the i-th data point as $\boldsymbol{y_i} \approx \boldsymbol{Ax_i}$, where $\boldsymbol{x_i}$ is a sparse vector, meaning most of its entries are zero. Various techniques are used to promote sparsity (Foucart et al., 2013, Micchelli et al., 2013). In this paper, we employ a local sparse coding approach first proposed in (Tasissa et al., 2023) In this approach, beyond regular sparsity, the representation also aims to utilize nearby atoms. In addition, entries of the sparse vectors are non-negative and sum to 1 and can be equivalently modeled as probabilities. Once sparse coding is completed, the resulting sparse representations can be used for downstream tasks. The main contribution of this paper is a plant disease identification algorithm based on this local sparse coding approach. The core idea is to use sparse coding on the data or extracted features to obtain sparse representations of the data points. To classify a test plant as healthy or diseased, we first compute its sparse representation and then compare it to the sparse representations of labeled plants from different classes using the cosine similarity metric. Given two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ the cosine similarity between the two vectors is defined as

,
$$sim(x, y) = \frac{x \bullet y}{||x||||y||} \tag{1}$$

where $x \bullet y$ indicates the dot product of two vectors and || || denotes the length of a vector.

Local sparse coding

Given n data points represented as $y_1, y_2, \dots, y_n$, with $A = [a_1, a_2, \dots a_m]$ denoting the dictionary, sparse coding seeks to obtain sparse representations $x_1, x_2, \dots, x_n$ such that $y_i \approx Ax_i$. Here on, we assume the data points lie in $\mathcal{R}^d$. Hence, $A \in \mathcal{R}^{d \times m}$ and $x_i \in \mathcal{R}^m$. In the typical setting, sparse coding solves the following problem to estimate $x_i \in \mathcal{R}^m$ (Lee et al., 2006):

$$\min_{x_i \in \mathcal{R}^m} \; \left|\left|Ax_i - y_i\right|\right|_2^2 + \lambda\left|\left|x_i\right|\right|_1, \tag{2}$$

where $||z||_2 = \sqrt{\sum_i z_i^2}$ denotes the ℓ_2 norm of a vector and $||z||_1 = \sum_i |z_i|$ denotes the ℓ_1

norm of a vector. The parameter λ is a regularization parameter that balances reconstruction of the data point y_i using the dictionary A with the sparsity level of x_i. The ℓ_1 regularization is one choice to promote sparsity of x_i. There are also regularizers designed to promote structured sparsity (Elhamifar et al., 2011, Huang et al., 2009). In this paper, we adopt the local sparse coding approach in (Tasissa et al., 2023). We briefly highlight the differences of this approach in contrast to the standard ℓ_1 regularized problem. First, each $x_i \in \mathcal{R}^m$ has non-negative entries that sum to 1. From here on, the set S represents the set of vectors in $\mathcal{R}^m$ with non-negative entries and that sum to 1. Second, in representing a data point y_i using the dictionary A, we seek sparse representations that put more weights on nearby atoms. More concretely, local sparse coding solves the following optimization problem to estimate $x_i \in \mathcal{R}^m$:

$$\min_{x_i \in S} \quad \left||Ax_i - y_i|\right|_2^2 + \lambda \sum_{j=1}^{m} (x_i)_j \left||y_i - a_j|\right|_2^2, \tag{3}$$

where $(x_i)_j$ indicates the j-th entry of x_i. We contrast the optimization problem in (2) with (3). We note that the second term in (3) is a proximity regularization that promotes representation of y_i using nearby atoms i.e., for suitably set regularization, if y_i is relatively far from a_j, $(x_i)_j$ will be relatively small. We note that the set S is a convex set and the objective of the optimization program in (3) is also convex. This implies that (3) is a convex optimization program, for which efficient algorithms exist to obtain the optimal sparse representations. In this paper, we use the algorithm in (Tasissa et al., 2023) which is based on algorithm unrolling (Monga et al., 2021). In our framework, it is assumed that the dictionary A is fixed. With that, the sparse representations $x_1, x_2, \dots, x_n$ can be computed independently.

Proposed Classification Algorithm

We assume that there are n training data points that belong to k distinct different classes. The image classification problem is to identify the class an unlabeled testing image belongs to. Here on $Y = [y_1, y_2, \dots y_n]$ denotes the matrix of the training data. The test data is denoted by y_test. In (Wright et al., 2008), a sparse coding approach for classification was proposed. The first step of the algorithm is to do sparse coding for each data point as follows:

$$min \ \left||x_i|\right|_1 \ subject\ to \left||Ax_i - y_i|\right|_2 \leq \epsilon \tag{3}$$

where ϵ models the level of noise in the image y_i. In (Wright et al., 2008), the dictionary A is assumed to be the training data matrix Y. To identify to which of the k classes a test data point belongs to, the work in (Wright et al., 2008) makes use of the following characteristic function. For each class i, define $\delta_i: R^m \to R^m$ as the function that selects the entries corresponding to class i. Formally, given a vector $x \in \mathcal{R}^m$, $[\delta_i(x)]_j = x_j$ if y_j is in class i and 0 otherwise. With that, a test data point is classified as follows:

$$\underset{i \in [1,2,\dots,k]}{argmin} \quad \left||A\delta_i(x_{test}) - y_{test}|\right|_2 \tag{4}$$

For our proposed method, to design an efficient sparse coding framework, the dictionary is not set to be all the training data points. Rather, it is based on m-centroids obtained from the K-means algorithm (Arthur et al., 2007, Lloyd, 1982). Given that, the atoms of dictionary are not directly associated to any training data. Therefore, we use cosine similarity metric, defined in (1), between the sparse representations of training and testing data to do classification. We compute the average cosine similarity between a testing data point and the training data points in a given class i as follows:

$$sim_{avg,i} = \frac{1}{n_i} \sum_{y_j \in \text{class } i} sim(x_j, x_{test}),$$

where n_i is the number of training data points in class i. The testing data point is assigned to the class with the highest average cosine similarity:

$$\text{identity}(y_{test}) = argmax_{i \in \{1,2,\dots,k\}}\, sim_{avg,i}.$$

If training and testing data points in the same class are generated by the same set of dictionary atoms with similar sparse codes, then the cosine similarity between the testing data points and these training data points should be high and close to 1. On the other hand, if the set of dictionary atoms used by training and testing data points in different classes use different sets of dictionary atoms, then the corresponding cosine similarity should be low and close to 0. In this way, we get a reliable label assignment for the testing data. We note that computing and comparing all the relevant cosine similarities with our local sparse codes can be done efficiently.

Results

In this section, we apply our proposed classification algorithm for plant disease identification using the PlantVillage dataset (Hughes et al., 2015, Mohanty et al., 2016), which consists of 54,305 images of crop leaves. This dataset includes 14 crop species and 38 classes of healthy or infected leaves. For our experiments, we use only the apple leaves, categorized into four classes: leaves with apple scab, leaves with black rot, leaves with cedar-apple rust, and healthy leaves. There are over 3,000 data points across these four classes, with the number of images per class ranging from 275 to 1,645.

The original data consists of 256-by-256 pixels RGB images. We use two methods to process the data for input into our algorithm. The first method is converting the RGB images to grayscale images and downsampling them to 15-by-15 pixels. The second method uses transfer learning (Zhuang et al., 2020). In this approach, we input the RGB images to the VGG16 network (Simonyan et al., 2015) and use the last fully connected layer with 4096 channels as the features of the images. The network is pretrained by ImageNet (Deng et al., 2009).

For both data pre-processing methods, we compare our algorithm with two baseline algorithms. Our algorithm uses the input data and applies local sparse coding to generate the sparse representations. In the end, it employs cosine similarity to classify the images. The first baseline algorithm classifies the images directly using cosine similarity on the input data. The second baseline algorithm first applies K-means clustering with the number of clusters equal to the number of classes. The sparse code for the data is based on the cluster that each data point gets. For example, if the data point i is assigned to the second cluster, then the sparse representation of it is a sparse vector where the only non-zero entry is a 1 in the second entry. With the sparse representation obtained in this manner, cosine similarity is again used for classification. In each experiment, half of the data is randomly chosen for training, while the remaining half is used for testing. We repeat the experiment five times and report the average results. The code to reproduce our numerical experiments can be found at this link.

Table 1: Average classification accuracy and standard deviation of 3 algorithms. For local sparse coding, the number of dictionary atoms is set to be 400. Methods with the highest average accuracy are in bold.

Methods	Direct Downsampling	VGG16 features
Local Sparse Coding	$\mathbf{0.572 \pm 0.011}$	$\mathbf{0.876 \pm 0.013}$
Raw Data	0.295 ± 0.018	0.744 ± 0.008
K-means Sparse Coding	0.256 ± 0.056	0.862 ± 0.015

Results are summarized in Table 1. When downsampling is used to process the data, local sparse coding obtains a 57.2% accuracy, significantly outperforming the baseline algorithm on the same data. When VGG16 features are used as the input, accuracy increases for all three methods. However, local sparse coding still outperforms the two baseline algorithms with an accuracy of 87.6%.

Conclusion or Summary

In this paper, we explored the application of local sparse coding for plant disease identification. Our proposed method efficiently obtains sparse features from input data, and classification is subsequently performed using average cosine similarity. Preliminary experiments on the PlantVillage dataset show the method yields competitive results. Future work will test the proposed method on extensive datasets.

Acknowledgments

Abiy Tasissa acknowledges support from NSF through grant DMS-2208392.

References

Aharon, M., Elad, M., & Bruckstein, A. (2006). K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, *54*(11), 4311-4322.

Alston, J. M., & Pardey, P. G. (2014). Agriculture in the global economy. *Journal of Economic Perspectives*, *28*(1), 121-146.

Antonini, M., Barlaud, M., Mathieu, P., & Daubechies, I. (1992). Image coding using wavelet transform. *IEEE Trans. Image Processing*, *1*, 20-5.

Arthur, D., & Vassilvitskii, S. (2007, January). k-means++: The advantages of careful seeding. In *Soda* (Vol. 7, pp. 1027-1035).

Calicioglu, O., Flammini, A., Bracco, S., Bellù, L., & Sims, R. (2019). The future challenges of food and agriculture: An integrated analysis of trends and solutions. *Sustainability*, *11*(1), 222.

Chaube, H. S., & Pundhir, V. S. (2005). *Crop diseases and their management*. PHI Learning Pvt. Ltd..

Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 248-255).

Elhamifar, E., & Vidal, R. (2011). Sparse manifold clustering and embedding. *Advances in neural information processing systems*, *24*.

Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis. *Computers and electronics in agriculture*, *145*, 311-318.

Foucart, S., & Rauhut, H. (2013). A Mathematical Introduction to Compressive Sensing.

Haridasan, A., Thomas, J., & Raj, E. D. (2023). Deep learning system for paddy plant disease detection and classification. *Environmental monitoring and assessment*, *195*(1), 120.

Huang, J., Zhang, T., & Metaxas, D. (2009, June). Learning with structured sparsity. In *Proceedings of the 26th Annual International Conference on Machine Learning* (pp. 417-424).

Hughes, D., & Salathé, M. (2015). An open access repository of images on plant health to enable the development of mobile disease diagnostics. *arXiv preprint arXiv:1511.08060*.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84-90.

Lee, H., Battle, A., Raina, R., & Ng, A. (2006). Efficient sparse coding algorithms. *Advances in neural information processing systems*, *19*.

Lloyd, S. P. (1982). Least Squares Quantization in PCM. *IEEE transactions on information theory*, *28*(2).

Lowe, D. G. (1999, September). Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision* (Vol. 2, pp. 1150-1157). Ieee.

Micchelli, C. A., Morales, J. M., & Pontil, M. (2013). Regularizers for structured sparsity. *Advances in Computational Mathematics*, *38*, 455-489.

Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using deep learning for image-based plant disease detection. *Frontiers in plant science*, *7*, 215232.

Monga, V., Li, Y., & Eldar, Y. C. (2021). Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. *IEEE Signal Processing Magazine*, *38*(2), 18-44.

Nixon, M., & Aguado, A. (2019). *Feature extraction and image processing for computer vision*. Academic press.

Olshausen, B. A., & Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, *381*(6583), 607-609.

Sheykhmousa, M., Mahdianpari, M., Ghanbari, H., Mohammadimanesh, F., Ghamisi, P., & Homayouni, S. (2020). Support vector machine versus random forest for remote sensing image classification: A meta-analysis and systematic review. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *13*, 6308-6325.

Shruthi, U., Nagaveni, V., & Raghavendra, B. K. (2019, March). A review on machine learning classification techniques for plant disease detection. In *2019 5th International conference on advanced computing & communication systems (ICACCS)* (pp. 281-284). IEEE.

Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. *International Conference on Learning Representations*.

Tasissa, A., Tankala, P., Murphy, J. M., & Ba, D. (2023). K-Deep Simplex: Manifold Learning via Local Dictionaries. *IEEE Transactions on Signal Processing*.

Wright, J., Yang, A. Y., Ganesh, A., Sastry, S. S., & Ma, Y. (2008). Robust face recognition via sparse representation. *IEEE transactions on pattern analysis and machine intelligence*, *31*(2), 210-227.

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., ... & He, Q. (2020). A comprehensive survey on transfer learning. *Proceedings of the IEEE*, *109*(1), 43-76.