Finding Representative Objects with Sparse Modeling

Junier B. Oliva joliva@cs.cmu.edu
Dougal J. Sutherland dsutherl@cs.cmu.edu
Yifei Ma * yifeim@cs.cmu.edu

Abstract

It is often useful to summarize a dataset by picking a small number of exemplars. This aids in visualizing and interpreting datasets; it is also applicable e.g. in reducing the search space for robotic motion planners or in reducing the size of a training set given to a classifier. We give an approach inspired by sparse coding that picks a few elements which can best represent the entire dataset through sparse linear combinations. We formulate the approach as a convex program, give algorithms based on generalized gradient descent and coordinate descent, and investigate the dual problem. Finally, we evaluate the efficacy of the optimization algorithms and the quality of the results on several synthetic and real-world datasets.

1 Introduction

In many situations, we wish to summarize a dataset with a handful of exemplars. This is useful in visualizing and interpreting data, in limiting the dimension of the search space for a robotic motion planner, or to reduce the size of a large training set for a classifier.

Many approaches exist that can address this problem, in particular clustering (where the data is assumed to lie in tight clusters; e.g. k-means or k-medoids [1]) and matrix factorization (where the data is assumed to be low-rank; [2, 3]). These also generally assume the data is low-dimensional. We make a different assumption: we seek a few points that can reconstruct the rest of the data via sparse linear combinations. We draw inspiration from sparse coding, where a signal is modeled as the linear combination of a few dictionary elements via the following optimization program:

$$\min_{D,C} \frac{1}{2}\|X - DC\|_F^2 + \lambda \sum_{i=1}^{n} \|\bar{C}_i\|_1, \tag{1}$$

where $X \in \mathbb{R}^{d\times n}$ is our dataset (columns are instances), $D \in \mathbb{R}^{d\times m}$ is the dictionary, $C \in \mathbb{R}^{m\times n}$ are the coefficients combining dictionary elements, and $\bar{C}_i$ represents the ith column of C. The $\ell 1$ penalty in (1) over the reconstruction columns $\bar{C}_i$ will result in sparse codings for each instance; i.e. each instance is represented as a linear combination of a small number of dictionary elements.

The problem in (1) is a convex linear problem in each of the parameters C and D, but not simultaneously. Thus, Aharon, Elad, and Bruckstein [4] present an iterative algorithm that alternates optimizing over C and then D, converging to a local minimum.

For some applications, it may be beneficial, or necessary, to restrict dictionary elements to be members of the datasets themselves. This guarantees that the dictionary elements will be valid members of the dataset, where in general they may sometimes comprise interpretable components of the data, but will often look quite different than the data. In this case, we can use a different objective:

$$\min_{C} \frac{1}{2}\|X - XC\|_F^2 + \lambda_1 \sum_{i=1}^{n} \|\bar{C}_i\|_1 + \lambda_2 \sum_{i=1}^{n} \|C_i\|_2, \tag{2}$$

*Final project for 10-725 Optimization. TA Mentor: Shiva Kaul.

where C_i denotes the ith row of C (as a column vector) and $\|C_i\|_2$ is a group penalty pushing whole rows towards zero, effectively reducing our dictionary to include only a few data points. Note that (2) is a convex optimization problem in C, which allows us to simultaneously find an optimal dictionary (over the dataset instances) and sparse codes using said dictionary. Furthermore, (2) can be solved with generalized gradient methods, which can be further extended with acceleration.

To make our C matrix more interpretable, it may be beneficial to restrict the optimization to be over $C \geq 0$ (elementwise); note that convexity is preserved for (2) with this reformulation. This is analogous to nonnegative matrix factorization [2], which often learns highly interpretable components of objects rather than additive and subtractive components that are harder for humans to understand.

This problem is potentially useful in many contexts. Visualization and interpretation of datasets is much easier if we can look at a few representative examples rather than the entire dataset (which is overwhelming) or a random sample (which will probably have many similar examples and might miss important sections of the space). We will show an application of this approach to modeling snippets of songs, where the exemplars comprise a core library of "fundamental sounds" that can be combined to reconstruct songs. Similarly, this approach might be helpful in reducing the complexity of a training set for classification algorithms, since data instances would now be represented as sparse linear combinations of a few canonical elements (i.e. using the codes that our method finds).

We can also use this model in planning or decision-making domains to reduce the search space, so that we can consider combinations a small subset of "canonical actions" rather than any possible action. We will show an example of using this techniques to find representative actions, also known as *motion primitives*, from logs of actions taken under human control. In addition to helping understand what types of trajectories are useful for surmounting a particular obstacle, certain combinations of the exemplars can be used as the possible actions considered by a planner.

2 Related work

Model Elhamifar, Sapiro, and Vidal [5] used a similar model for finding representative items, focusing on applications in computer vision, but used a different framework. They do not directly enforce sparsity in the reconstructions, instead requiring that representative selection be invariant to a global translation. They also prove that when the data comes from a union of low-rank models, their method selects a few representatives from each. They later extended this work to employ general pairwise similarity measures, rather than assuming the data lie in a Euclidean space [6]. Both optimization problems were solved using the alternating direction method of multipliers (ADMM) approach.

Esser et al. [7] also used a similar model, but encouraged row sparsity with the ℓ_∞ norm instead of ℓ_2. They also consider the $C \geq 0$ case, and optimize using ADMM.

Sprechmann et al. [8] used a general form of the regularization terms in (2) but used an arbitrary dictionary rather than fixing it as the dataset elements themselves. They solve their optimization problem using a proximal gradient method similar to ours, but do not employ acceleration.

Music components Infering music's underlying structure from raw sound data is very difficult and current methods for doing so achieve limited success. Rich interactions with our music data such as library visualization, automatic playlist creation, and recommendation systems may be possible if given relevant musical features from sound. Systems such as Pandora have implemented techniques for playlist generation, but their methods rely heavily on costly human tagging. Instead of using human generated tags to create richer user experiences, it may be possible to do music information retrieval (MIR) from low level auditory features using machine learning.

Previous approaches such as [9, 10] use techniques like Gaussian Mixture Models that largely ignore important temporal aspects of music to pick out salient sounds in very small windows. Given our proposed algorithm's scalabilty, we are able to pick canonical elements of sound that are an order of magnitude or more longer than those with previous methods. Such longer canonical elements may be used to better represent music data and preform MIR tasks.

Robotic motion primitives It has long been recognized that the problem of motor control can be significantly simplified by decomposing motions into primitive "building blocks," often referred to

as *motion primitives* (e.g. [11]). This can drastically reduce the dimensionality of the control space in either biological or robotic settings. In most prior work, however, e.g. [12, 13, 14, 15], the learned motion primitives typically are learned via k-means clustering, and so in addition to being unstable across independent runs, they are not restricted to be motions actually observed, and so may be impractical or misrepresent the geometry of the space.

3 Optimization algorithm

We first show how to solve (2) with a proximal gradient method. We then give an alternative approach based on coordinate minimization, and show that we can easily kernelize our model.

3.1 Proximal gradient method

Note that our objective function (2) contains a differentiable portion:

$$g(C) = \frac{1}{2}\|X - XC\|_F^2 \tag{3}$$

and another convex portion:

$$h(C) = \lambda_1 \sum_{i=1}^{n} \|\bar{C}_i\|_1 + \lambda_2 \sum_{i=1}^{n} \|C_i\|_2 = \lambda_1 \sum_{i=1}^{n} \|C_i\|_1 + \lambda_2 \sum_{i=1}^{n} \|C_i\|_2. \tag{4}$$

Hence, we may use the generalized gradient descent method. Our updates will be of the form:

$$C^{(k)} = \text{prox}_{t_k}(C^{(k-1)} - t_k \nabla g(C^{(k-1))}). \tag{5}$$

Note that this can be easily extended using acceleration [16]. Our implementation uses a backtracking line search search for step sizes.

3.1.1 Matrix derivative

Let C be the previous iterate. To calculate the next iterate, we must find the derivative of $g(\cdot)$:

$$\mathrm{d}g(C) = \mathrm{d}\left(\tfrac{1}{2}\|X - XC\|_F^2\right) = \tfrac{1}{2}\mathrm{d}\,\mathrm{tr}\left((I-C)^T X^T X (I-C)\right).$$

Thus

$$\nabla g(C) = X^T X (C - I). \tag{6}$$

3.1.2 Proximal gradient

The proximal gradient corresponding to $h(\cdot)$ is, letting $\gamma_1 = t\lambda_1$ and $\gamma_2 = t\lambda_1$:

$$\begin{aligned}\text{prox}_t(X) &= \underset{Z}{\text{argmin}}\, \frac{1}{2}\|X - Z\|_F^2 + \gamma_1 \sum_{i=1}^{n} \|Z_i\|_1 + \gamma_2 \sum_{i=1}^{n} \|Z_i\|_2 \\ &= \underset{Z}{\text{argmin}} \sum_{i=1}^{n} \left(\frac{1}{2}\|X_i - Z_i\|_2^2 + \gamma_1\|Z_i\|_1 + \gamma_2\|Z_i\|_2\right).\end{aligned}$$

Hence, we can minimize each row separately. Consider $y = Z_i$, $x = X_i$ for some $i \le n$. Then:

$$\begin{aligned} z &= \underset{y}{\text{argmin}}\, \frac{1}{2}\|x - y\|^2 + \gamma_1\|y\|_1 + \gamma_2\|y\|_2 \\ 0 &= z - x + \gamma_1 v + \gamma_2 u \\ z &= x - \gamma_1 v - \gamma_2 u \end{aligned} \tag{7}$$

where $v \in \partial\|\cdot\|_1(z)$ and $u \in \partial\|\cdot\|_2(z)$. Note that

$$(\partial\|\cdot\|_1(z))_i \in \begin{cases} [-1, 1] & \text{if } z_i = 0 \\ \{\text{sign}(z_i)\} & \text{otherwise} \end{cases}$$

and

$$\partial\|\cdot\|_2(z) \in \begin{cases} \mathcal{B}^2(1) & \text{if } z = 0 \\ \left\{ \frac{z}{\|z\|_2} \right\} & \text{otherwise} \end{cases}$$

where $\mathcal{B}^2(1)$ is the $\ell 2$ unit ball in $\mathbb{R}^n$. Hence, $z = 0$ iff for some $v \in \mathcal{B}^\infty(1)$, $u \in \mathcal{B}^2(1)$

$$\begin{aligned} 0 &= x - \gamma_1 v - \gamma_2 u \\ x - \gamma_1 v &= \gamma_2 u \\ \|S_{\gamma_1}(x)\|_2 &\leq \gamma_2, \end{aligned} \tag{8}$$

where $S_{\gamma_1}(x)$ is vector soft-thresholding. Furthermore, $z \neq 0$, $z_i = 0$ iff

$$\begin{aligned} 0 &= x_i - \gamma_1 v_i - \gamma_2 \frac{0}{\|z\|_2} \\ x_i &\in [-\gamma_1, \gamma_1]. \end{aligned} \tag{9}$$

Likewise, $z \neq 0$, $z_i \neq 0$ iff

$$z_i + \gamma_2 \frac{z_i}{\|z\|_2} = x_i - \gamma_1 \operatorname{sign}(z_i) \tag{10}$$

$$z_i = \frac{x_i - \gamma_1 \operatorname{sign}(x_i)}{1 + \frac{\gamma_2}{\|z\|_2}}, \tag{11}$$

where $\operatorname{sign}(z_i) = \operatorname{sign}(x_i)$ because the LHS of (10) has the same sign as z_i, and (9) implies that the RHS has the same sign as x_i. Thus, combining (9) and (11), $z \neq 0$ iff

$$z_i = \frac{S_{\gamma_1}(x_i)}{1 + \frac{\gamma_2}{\|z\|_2}} = \|z\|_2 \, \frac{S_{\gamma_1}(x_i)}{\|z\|_2 + \gamma_2}. \tag{12}$$

And so $z \neq 0$ iff

$$\begin{aligned} \|z\|_2^2 &= \|z\|_2^2 \, \|S_{\gamma_1}(x)\|_2^2 \; / \; \left(\|z\|_2 + \gamma_2\right)^2 \\ \|z\|_2 &= \|S_{\gamma_1}(x)\|_2 - \gamma_2. \end{aligned} \tag{13}$$

Thus, with (8), (12), and (13), we have that:

$$z = \operatorname{prox}_t(x) = S_{t\lambda_2}\left(\|S_{t\lambda_1}(x)\|_2 \right) \frac{S_{t\lambda_1}(x)}{\|S_{t\lambda_1}(x)\|_2}, \tag{14}$$

so that we do ℓ_1 and then ℓ_2 soft-thresholding. When the $C \geq 0$ restriction is present, we simply set any negative components to zero before thresholding; the proof is similar.

3.2 Coordinate descent

The nonsmooth portion of the objective (4) can be separated into functions on each row of C:

$$h(C) = \sum_{i=1}^{n} \Omega\left(C_i\right) \qquad \text{where} \qquad \Omega(x) = \lambda_1 \|x\|_1 + \lambda_2 \|x\|_2. \tag{15}$$

Coordinate descent on the rows, where at each step we update one row to its optimum holding other points fixed, will therefore converge to the global optimum [17]; we can also use an active set strategy to speed up the process. Let $f_i(y)$ be the objective value of C with the ith row replaced by y; we will minimize f_i for each i sequentially. Defining $Z_i = \sum_{j \neq i} \bar{X}_j C_j^T$, we have

$$\begin{aligned} \operatorname*{argmin}_y f_i(y) &= \operatorname*{argmin}_y \; \tfrac{1}{2}\|X - \sum_{j \neq i} \bar{X}_j C_j^T - \bar{X}_i y^T\|_F^2 + \sum_{j \neq i} \Omega(C_j) + \Omega(y) \\ &= \operatorname*{argmin}_y \; \tfrac{1}{2} \operatorname{tr}\left(y \bar{X}_i^T \bar{X}_i y^T \right) - \operatorname{tr}\left(\left(X - Z_i\right)^T \bar{X}_i y^T \right) + \Omega(y) \\ &= \operatorname*{argmin}_y \; \tfrac{1}{2} y^T y - \frac{1}{\|\bar{X}_i\|_2^2} y^T \left(X - Z_i\right)^T \bar{X}_i + \frac{1}{\|\bar{X}_i\|_2^2} \Omega(y) \\ &= \operatorname*{argmin}_y \; \frac{1}{2} \left\| \frac{1}{\|\bar{X}_i\|_2^2} \left(X - Z_i\right)^T \bar{X}_i - y \right\|_2^2 + \frac{\lambda_1}{\|\bar{X}_i\|_2^2} \|y\|_1 + \frac{\lambda_2}{\|\bar{X}_i\|_2^2} \|y\|_2, \end{aligned}$$

which is exactly the form of (7). Thus, using (14), the coordinate minimizer is:

$$\begin{aligned} C_i^+ &= \operatorname{prox}_{1/\|\bar{X}_i\|_2^2}\left(\frac{1}{\|\bar{X}_i\|_2^2}\left(X^T - \sum_{j\neq i} C_j \bar{X}_j^T\right)\bar{X}_i\right) \\ &= \operatorname{prox}_{1/\|\bar{X}_i\|_2^2}\left(\frac{1}{\|\bar{X}_i\|_2^2}\left(X^T - C^T X^T + C_i \bar{X}_i^T\right)\bar{X}_i\right) \\ &= \operatorname{prox}_{1/\|\bar{X}_i\|_2^2}\left(C_i + \frac{1}{\|\bar{X}_i\|_2^2}\left(I - C\right)^T X^T \bar{X}_i\right). \end{aligned} \tag{16}$$

Unfortunately, this requires a matrix-vector multiplication at each step, and so is too slow for large problems.

3.3 Kernelization

Note that the update steps of Sections 3.1 and 3.2 only use the data as $X^T X$, the Gram matrix of inner products between data points. We can also express the objective in terms of the Gram matrix:

$$\|X - XC\|_F^2 = \|X(I-C)\|_F^2 = \operatorname{tr}\left((I - C^T)X^T X(I-C)\right).$$

Thus we can apply our techniques to data from any Hilbert space, including ones that are high- or even infinite-dimensional, without penalty.

4 Dual problem

We now introduce the dual problem to (2), whose investigation will yield insight into the properties of the primal problem. Using the notation of Section 3.2, we introduce a dummy variable $Z = XC$:

$$\min_{C,Z} \tfrac{1}{2}\|X - Z\|_F^2 + \sum_{i=1}^n \Omega(C_i) \quad \text{s.t.} \quad Z = XC.$$

Introducing a dual (matrix) variable U, our dual problem becomes

$$\max_U \min_{C,Z} \tfrac{1}{2}\|X - Z\|_F^2 + \sum_{i=1}^n \Omega(C_i) + \operatorname{tr}\left(U^T(Z - XC)\right). \tag{17}$$

Splitting up the minimization, the Z terms in (17) yield:

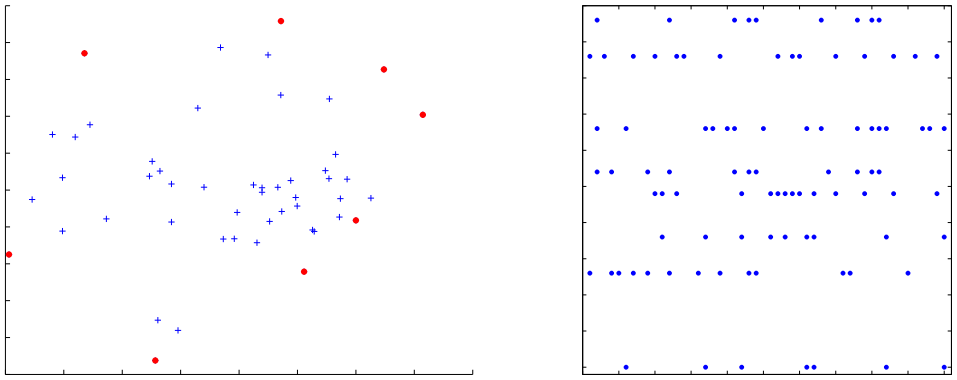
$$\begin{aligned} \min_Z \tfrac{1}{2}\|X - Z\|_F^2 + \operatorname{tr}(U^T Z) &= \min_Z \tfrac{1}{2}\operatorname{tr}(X^T X) - \operatorname{tr}(X^T Z) + \tfrac{1}{2}\operatorname{tr}(Z^T Z) + \operatorname{tr}(U^T Z) \\ &= \tfrac{1}{2}\operatorname{tr}(X^T X) + \min_Z \operatorname{tr}\left((U - X)^T Z\right) + \frac{1}{2}\operatorname{tr}(Z^T Z) \\ &= \tfrac{1}{2}\operatorname{tr}(X^T X) - \tfrac{1}{2}\|U - X\|_F^2 + \min_Z \|(U - X) + Z\|_F^2 \\ &= \tfrac{1}{2}\operatorname{tr}(X^T X) - \tfrac{1}{2}\|U - X\|_F^2. \end{aligned}$$

The C terms are:

$$\begin{aligned} \min_C \sum_{i=1}^n \Omega(C_i) - \operatorname{tr}(U^T XC) &= \sum_{i=1}^n \min_{C_i}\left(\Omega(C_i) - \left(U^T \bar{X}_i\right)^T C_i\right) \\ &= -\sum_{i=1}^n \mathbb{I}\left\{\Omega_*(U^T \bar{X}_i) \leq 1\right\}, \end{aligned}$$

where $\Omega_*(y)$ is the dual norm of $\Omega(y)$. The dual norm has value μ such that $\mu = \frac{1}{\lambda_2}\|S_{\mu\lambda_1}(y)\|_2$; the appendix gives a derivation and a method for evaluation. Thus (17) becomes:

$$\tfrac{1}{2}\operatorname{tr}\left(X^T X\right) - \min_U \tfrac{1}{2}\|X - U\|_F^2 \quad \text{s.t.} \quad \forall i.\ \Omega_*(U^T \bar{X}_i) \leq 1. \tag{18}$$



(a) The data; red circles are the dictionary elements. (b) Sparsity of the C matrix found for data in (a).

Figure 1: Results of optimization using non-negative C and $\lambda_1 = .5$, $\lambda_2 = .5$.

The stationarity condition of (17) with respect to Z is $Z - X + U = 0$, so that $XC = X - U$. The condition with respect to C_i is $U^T \bar{X}_i = \lambda_1 v_i + \lambda_2 u_i$, where $v_i \in \partial \|\cdot\|_1(C_i)$ and $u_i \in \partial \|\cdot\|_2(C_i)$, so that $C_i = 0$ iff $\|S_{\lambda_1}(U^T \bar{X}_i)\|_2 \leq \lambda_2$.

This tells us that for a given λ_1, the value $\lambda_2^{\max}(\lambda_1)$ which results in an optimum of $C = 0$ (i.e. $U = X$) is $\max_i \|S_{\lambda_1}(X^T \bar{X}_i)\|_2$. Note $U = X$ then satisfies the dual norm constraint in (18).

A SAFE rule that zeroes out complete rows of C [18] only requires solving

$$\max_U \|S_{\lambda_1}(U^T \bar{X}_i)\|_2^2 \quad \text{s.t.} \quad \tfrac{1}{2}\operatorname{tr}(X^T X) - \tfrac{1}{2}\|X - U\|_F^2 \geq \gamma. \tag{19}$$

5 Experiments

5.1 Simulated data

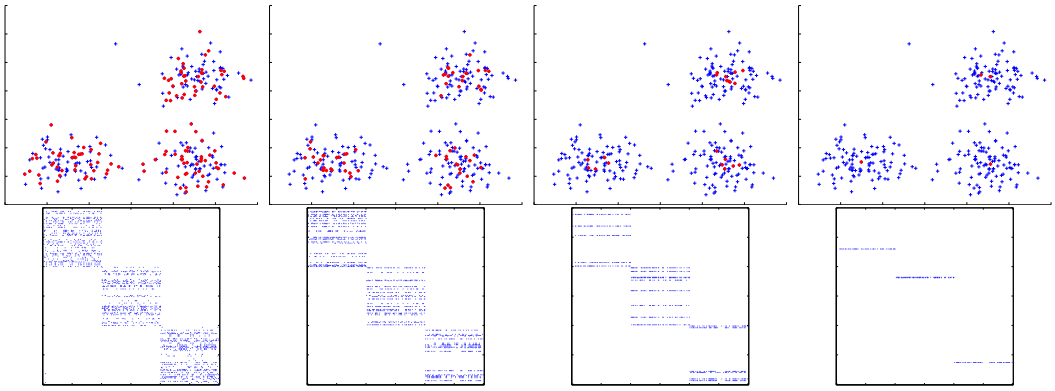
In order to test the behavior of our optimization program, we optimized our objective on two different simulated datasets.

First, we generated 50 data points from a 2-d standard normal. We ran our optimization program on the dataset, restricting C to be non-negative. As one would expect, the dictionary elements found correspond to the convex hull of the dataset (Figure 1). These are by definition the elements that can be combined in a non-negative linear combination to form all other elements.

We next considered a dataset of 150 data points drawn from a mixture of three 2-d normals, each equally likely, with identity covariance and means $(1, 1)$, $(7, 7)$, and $(1, 7)$ respectively. We then optimized the kernelized version of our problem with an RBF kernel ($\sigma = 1$). The optimization was then carried out using various configuration of λ_2 (Figure 2). We see that the optimization in a way subsamples the data (to a degree controlled by penalties) while biasing the selected dictionary elements to be close to their respective centers. That is, kernelized optimization with the RBF kernel seems to be similar to medoid sub-sampling.

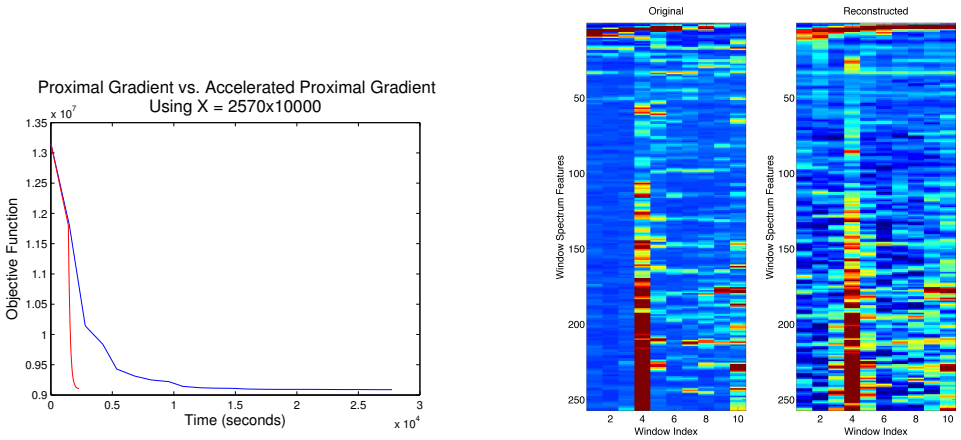
5.2 Music

In this section we outline our methodology's use on a collection of nearly 90,000 30-second `mp3` previews of songs scraped from Amazon's online music store, in the genres of classical, electronica, hip-hop, pop, and rock. Using those songs from the genre of electronica, a subset of 10,000 snipets of 0.1 seconds of sound was randomly subsampled. Each snippet was in turn represented as a vector of power spectrogram features (2570 features in total). In order to optimize our problem with this dataset and choosing $\lambda_1 = 200$, $\lambda_2 = 20,000$, both a back-tracking version of non-accelerated and accelerated proximal gradient descent were run (Figure 3a). As can be seen from Figure 3a, the accelerated version of the optimization algorithm runs much faster, finishing in about 38 minutes versus over 7 hours with no acceleration.



(a) $\lambda_2 = 0.5$ (b) $\lambda_2 = 1$ (c) $\lambda_2 = 1.5$ (d) $\lambda_2 = 2.03$

Figure 2: The exemplars found by optimization using an RBF kernel with the $C \geq 0$ constraint, $\lambda_1 = 0.5$, and various λ_2.



(a) Objective versus runtime for accelerated (red) and non-accelerated (blue) algorithms.

(b) An original feature vector for 0.1 seconds of sound (left), versus a reconstruction using the found dictionary (right).

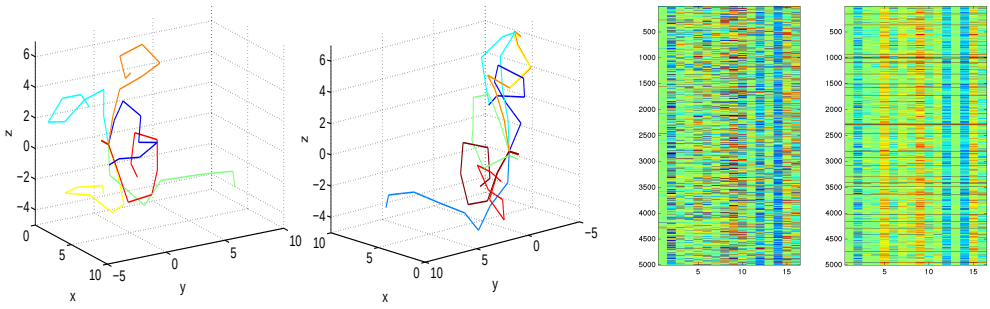
Figure 3: Results on the music dataset.

Preliminary experiments suggest that the found dictionary of 329 data elements is well suited for sparse coding. An example reconstruction of an unseen feature vector using the dictionary found can be seen in Figure 3b. Typically, encoding a song sparsely with the dictionary and then reconstructing the raw sound data kept the song recognizable. Moreover, in 10 different trials of sparse coding 20 different unseen songs with a ℓ_1 penalty of 200 using the found dictionary versus a dictionary of 329 randomly selected data instances (from the same set X), the found dictionary had an average RMSE reconstruction error of 25.5 per 2570 dimensional feature vector versus 29.5 for the random dictionaries.

5.3 Robotic motions

We now present an example of using our system to find motion primitives for a snake-like robot. We have logs of 304 trials where users attempted to control the robot to climb over a low wall-like obstacle. Each trial is approximately one to two minutes long; we have traces of the joint motor activations of the robot and an annotation as to whether the robot successfully passed over the obstacle. 23% of trials were successful.

To construct the dictionary, we randomly sampled 5,000 static snake poses of successful trials and found with $\lambda_1 = 2, \lambda_2 = 100$ a dictionary of 15 elements. Figure 4a shows six of these static poses, placed such that the heads of the six snakes are held together at the origin. We also found



(a) Dictionary from successes. (b) Dictionary from failures. (c) The dataset of original (left) and reconstructed (right) poses.

Figure 4: Learned dictionaries, and the reconstructed dataset of successful trials. In (c), each row is a single sampled vector of joint angle positions. The reconstruction is not perfect because the number of representatives is small.

the dictionary from 500 static snake poses in failed attempts in Figure 4b. Comparing the two dictionaries, successful poses tend to be more curled, as more snapshots are taken when the snake is over the obstacle than when the snake is beginning to crawl onto the obstacle.

We also tried to reconstruct the original 5,000 successful poses, each represented by a vector of its 16 joint angles, using the 15 dictionary elements we learned. The reconstructed poses are shown in Figure 4c.

6 Discussion

Across a wide array of different domains finding exemplars, or canonical elements, in a dataset is a useful task. We present a study of a convex optimization problem for finding data instances that may be sparsely linearly combined to form all other data instances. While the usual formulation (1) for finding a dictionary D and sparse codes C to encode a dataset is not convex in all primal variables, our problem (2) is convex and hence does not suffer from any local minima.

In order to solve our algorithm we derived a proximal gradient descent algorithm (which may be accelerated) and a coordinate descent algorithm that works through the rows of C. While the coordinate descent algorithm may be more efficient for smaller problems, the accelerated proximal gradient descent algorithm is better suited for larger datasets.

Also, in order to further analyze the nature of the solutions to our optimization problem we derived its dual problem (18). Among other things, the dual formulation allows us to find a λ_2 value such that it will completely zero out the C matrix. Moreover, with the dual problem, we see that a SAFE rule for ruling out entire rows of C may be derived by solving the optimization problem in (19) for each row of C.

Experiments on synthetic data provide confirmation that the solution to our objective behaves in a desirable fashion. Furthermore, experiments on music and robot data suggest that our optimization technique is finding useful canonical elements in real-world datasets.

Future work will focus on fully developing a SAFE rule for our objective, studying the statistical properties of solutions, and further evaluating the quality of found canonical elements in real-world datasets.

References

[1] L. Kaufman and P. J. Rousseeuw. "Clustering by Means of Mediods". In: *Statistical Data Analysis based on the L1 Norm*. Ed. by Y Dodge. 1987, pp. 405–416.

[2] D. D. Lee and H. S. Seung. "Learning the parts of objects by non-negative matrix factorization". In: *Nature* 401.6755 (1999), pp. 788–791.

[3] M. Gu and S. C. Eisenstat. "Efficient algorithms for computing a strong rank-revealing QR factorization". In: *SIAM Journal on Scientific Computing* 17.4 (1996), pp. 848–869.

[4] M. Aharon, M. Elad, and A. Bruckstein. "The K-SVD algorithm". In: vol. 5. 2005.

[5] E. Elhamifar, G. Sapiro, and R. Vidal. "See All by Looking at A Few: Sparse Modeling for Finding Representative Objects". In: *Computer Vision and Pattern Recognition*. 2012.

[6] E. Elhamifar, G. Sapiro, and R. Vidal. "Finding Exemplars from Pairwise Dissimilarities via Simultaneous Sparse Recovery". In: *Advances in Neural Information Processing Systems*. 2012.

[7] E. Esser et al. "A Convex Model for Nonnegative Matrix Factorization and Dimensionality Reduction on Physical Space". In: *IEEE Trans on Image Processing* 21.7 (2012), pp. 3239–3252.

[8] P. Sprechmann et al. "C-HiLasso: A Collaborative Hierarchical Sparse Modeling Framework". In: *IEEE Transactions on Signal Processing* 59.9 (2011), pp. 4183–4198.

[9] B. Logan, D. Ellis, and A. Berenzweig. "Toward evaluation techniques for music similarity". In: *SIGIR 2003: Workshop on the Evaluation of Music Information Retrieval Systems, 1 August 2003, Toronto, Canada*. 2003.

[10] M. Mandel and D. Ellis. "Song-level features and support vector machines for music classification". In: *ISMIR 2005: 6th International Conference on Music Information Retrieval: Proceedings: Variation 2: Queen Mary, University of London & Goldsmiths College, University of London, 11-15 September, 2005*. Queen Mary, University of London. 2005, pp. 594–599.

[11] E. Todorov and Z. Ghahramani. "Unsupervised learning of sensory-motor primitives". In: *International Conference of the IEEE Engineering in Medicine and Biology Society*. 2003.

[12] J. DiGiovanna et al. "Arm Motion Reconstruction via Feature Clustering in Joint Angle Space". In: *International Joint Conference on Neural Networks*. 2006.

[13] D. Kulić, W. Takano, and Y. Nakamura. "Incremental on-line hierarchical clustering of whole body motion patterns". In: *IEEE International Symposium on Robot and Human interactive Communication* (2007).

[14] D. Kulić et al. "Incremental learning of full body motion primitives for humanoid robots". In: *IEEE-RAS International Conference on Humanoid Robots*. 2008.

[15] C.-S. L. Elgammal and Ahmed. "Human Motion Synthesis by Motion Manifold Learning and Motion Primitive Segmentation". In: *Articulated Motion and Deformable Objects*. 2006, pp. 1–10.

[16] A. Beck and M. Teboulle. "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems". In: *SIAM Journal on Imaging Sciences* 2.1 (2009), pp. 183–202.

[17] P. Tseng. "Convergence of a block coordinate descent method for nondifferentiable minimization". In: *Journal of optimization theory and applications* 109.3 (2001), pp. 475–494.

[18] L. El Ghaoui, V. Viallon, and T. Rabbani. *Safe Feature Elimination for the LASSO and Sparse Supervised Learning Problems*. Sept. 2010. arXiv: `1009.4219 [cs.LG]`.

A Appendix: Dual Norm

We now consider the dual of the norm $\Omega(x) = \lambda_1\|x\|_1 + \lambda_2\|x\|_2$. We can write its dual as an optimization problem:

$$\begin{aligned}\Omega_*(y) &= \max y^T x \quad \text{s.t.} \quad \Omega(x) \leq 1 \\ &= -\min -y^T x \quad \text{s.t.} \quad \lambda_1\|x\|_1 + \lambda_2\|x\|_2 \leq 1.\end{aligned}$$

The KKT conditions of this strictly feasible convex minimization problem are

$$\begin{gathered}-y + \mu\lambda_1 v + \mu\lambda_2 u = 0 \\ v \in \partial\|\cdot\|_1(x) \qquad u \in \partial\|\cdot\|_1(x) \\ \lambda_1\|x\|_1 + \lambda_2\|x\|_2 \leq 1 \qquad \mu \geq 0 \\ \mu\left(\lambda_1\|x\|_1 + \lambda_2\|x\|_2 - 1\right) = 0.\end{gathered}$$

First note that $\mu = 0$ implies $y = 0$. Thus for any nonzero y, these reduce to

$$\begin{gathered}-y + \mu\lambda_1 v + \mu\lambda_2 u = 0 \\ v \in \partial\|\cdot\|_1(x) \qquad u \in \partial\|\cdot\|_1(x) \\ \lambda_1\|x\|_1 + \lambda_2\|x\|_2 = 1 \qquad \mu > 0.\end{gathered}$$

For any $y \neq 0$, clearly $x \neq 0$ at optimum. Thus $v = x/\|x\|_2$, and the stationarity condition is

$$\frac{x}{\|x\|_2} = \frac{1}{\lambda_2\mu}(y - \mu\lambda_1 v). \tag{20}$$

Suppose that $x_i \neq 0$. Then $v_i = \operatorname{sign}(x_i)$ and

$$\frac{x_i}{\|x\|_2} = \frac{1}{\lambda_2\mu}(y_i - \mu\lambda_1 \operatorname{sign}(x_i)).$$

If $x_i > 0$, this requires that $y_i > \mu\lambda_1$; if $x_i < 0$, $y_i < -\mu\lambda_1$, so that $\operatorname{sign}(x_i) = \operatorname{sign}(y_i)$ for all $x_i \neq 0$. On the other hand, if $x_i = 0$, then $y_i = \mu\lambda_1 v_i$, so that $y_i \in [-\mu\lambda_1, \mu\lambda_1]$. Thus

$$\hat{x} = \frac{x}{\|x\|_2} = \frac{1}{\mu\lambda_2} S_{\mu\lambda_1}(y), \tag{21}$$

so that $\mu = \frac{1}{\lambda_2}\|S_{\mu\lambda_1}(y)\|_2$.

Suppose that we knew the set of indices $I(\mu) = \{i : |y_i| > \mu\lambda_1\}$. Then we would have that

$$\lambda_2^2\mu^2 = \sum_{i \in I(\mu)} \left(|y_i| - \mu\lambda_1\right)^2 = \sum_{i \in I(\mu)} \left(y_i^2 - 2\mu\lambda_1|y_i| + \mu^2\lambda_1^2\right), \tag{22}$$

which is a quadratic in μ we can easily solve in closed form.

But note that clearly $\nu(\mu) = \|S_{\mu\lambda_1}(y)\|_2$ is nonincreasing in μ. Also, the $I(\mu)$ changes only at points such that $\mu = |y_i|/\lambda_1$ for some i. Thus to find the optimal μ, define z to be the sorted array of $|y_i|/\lambda_1$, so that $z_1 = \min|y_i|/\lambda_1$. We then perform binary search over z to find an index k such that $\nu(z_k) \geq \lambda_2 > \nu(z_{k+1})$. We then know that $I(\mu) = I(z_k)$ and can solve (22) to find μ. Since ν is quadratic and nonincreasing on the interval $[z_k, z_{k+1})$, there will be a unique solution.

We now know that our optimum has $x = \alpha\hat{x}$ for some scalar α. But we also have

$$\begin{gathered}\lambda_1\|\alpha\hat{x}\|_1 + \lambda_2\|\alpha\hat{x}\|_2 = 1 \\ \alpha = \frac{1}{\lambda_1\|\hat{x}\|_1 + \lambda_2\|\hat{x}\|_2} \\ x = \alpha\hat{x} = \frac{S_{\mu\lambda_1}(y)}{\mu\lambda_1\lambda_2\|\hat{x}\|_1 + \mu\lambda_2^2\|\hat{x}\|_2} = \frac{S_{\mu\lambda_1}(y)}{\lambda_1\|S_{\mu\lambda_1}(y)\|_1 + \lambda_2\|S_{\mu\lambda_1}(y)\|}.\end{gathered}$$

Using the original stationarity condition:

$$\begin{aligned}
y^T x &= \mu\lambda_1 v^T x + \mu\lambda_2 \frac{x^T x}{\|x\|_2} \\
&= \lambda_1 \frac{\alpha}{\lambda_2} v^T S_{\mu\lambda_1}(y) + \lambda_2 \frac{\alpha}{\lambda_2} \frac{S_{\mu\lambda_1}(y)^T S_{\mu\lambda_1}(y)}{\|S_{\mu\lambda_1}(y)\|_2} \\
&= \alpha \frac{\lambda_1}{\lambda_2} \|S_{\mu\lambda_1}(y)\|_1 + \alpha \|S_{\mu\lambda_1}(y)\|_2 \\
&= \frac{1}{\lambda_2} \frac{\lambda_1 \|S_{\mu\lambda_1}(y)\|_1 + \lambda_2 \|S_{\mu\lambda_1}(y)\|_2}{\frac{1}{\lambda_2\mu}\left(\lambda_1 \|S_{\mu\lambda_1}(y)\|_1 + \lambda_2 \|S_{\mu\lambda_1}(y)\|_2\right)} \\
&= \mu.
\end{aligned}$$

Thus $\Omega_*(y) = -(-\mu) = \mu$.