Unsupervised Feature Learning via Sparse Hierarchical Representations[1]

Yale Chang

July 4, 2014

1 Introduction

Learning features from labeled data is related to several research areas in machine learning, including multiple kernel learning, neural networks, multi-task learning and transfer learning. The question naturally follows:

How can we learn feature representations in an unsupervised way?

Many unsupervised learning algorithms can be used to learn features from unlabeled data, such as clustering methods (e.g., K-means and Gaussian mixture model), matrix factorization methods (e.g., PCA and other variants, ICA, NMF, and sparse coding), and nonlinear embedding (e.g., ISOMAP, Locally Linear Embedding, Laplacian Eigenmaps, restricted Boltzmann machines, etc.)

Sparse Coding is fundamentally different from those sparse subspace learning methods, e.g., sparsePCA and sparseLDA. Instead of learning a sparse coefficient matrix, the sparse subspace learning methods learn a sparse basis matrix. The low dimensional representation matrix (coefficient matrix) learned by these sparse subspace learning methods is still dense[2].

1.1 Matrix Factorization

Matrix factorization methods can be viewed as decomposing each input example as a linear combination of basis vectors. $X = BS$, where $X \in \mathbb{R}^{d \times m}$, $B \in \mathbb{R}^{d \times n}$, $S \in \mathbb{R}^{n \times m}$, where $d$ is the dimension of the data, $m$ is the number of examples, and each column of $X$ corresponds to an individual input example, $n$ is the number of basis vectors, and each column of $B$ corresponds to an individual basis vector. Therefore, each column of $S$ represents “coefficients” of the basis vectors for each input example. Algorithms such as PCA can be viewed as matrix factorization methods. There are several variants of matrix factorization, such as sparse matrix factorization (e.g. assuming the entries of $B$ and $S$ are sparse), factor analysis, and probabilistic matrix factorization. Other notable matrix factorization methods include independent component analysis (ICA) and non-negative matrix factorization (NMF).

In this thesis, we consider sparse coding as an algorithm for unsupervised feature learning. Sparse coding can be also viewed as matrix factorization where the coefficients are assumed to be sparse. One advantage of sparse coding is that it can learn succinct representations of the input data, where each input example can be represented as a linear combination of small number of basis vectors. As a result, the basis vectors (also called “dictionary”) are forced to capture salient patterns in the data, thus the basis vectors and the coefficients are often easy to interpret. Sparse coding allows
for an overcomplete basis set (i.e., large dictionary), where the number of basis vectors is larger than the dimension of the input vector. Finally, sparse coding is robust to noise in the data or “change” in the input attributes. This is because the coefficients are not determined as a simple linear function of the input, but it goes through a sparsification process due to the sparse prior on the coefficients. This property makes sparse coding a good candidate in settings where (1) unlabeled data and labeled data have different distributions or (2) input data contain much noise.

The above mentioned matrix factorization methods can be viewed as some sort of linear embeddings. A more flexible representation may be obtained by nonlinear embedding. In fact, numbers algorithms exist for nonlinear embedding, such as Kernel PCA, ISOMAP, Locally Linear Embedding, Laplacian Eigenmaps, Gaussian process latent variable models, auto encoders, restricted Boltzmann machines (RBMs) and deep belief networks (DBNs).

The main goal of this thesis is to develop algorithms that can extract rich, high-level structures from unlabeled data. To learn from unlabeled data, we build probabilistic generative models so that we can perform probabilistic inference over the data. Three ingredients of this thesis are: (1) sparsity, (2) hierarchy, (3) scaling up by convolutional structure.

2 Efficient Sparse Coding Algorithms[3]

\[
\min_{B,S} \frac{1}{2\sigma^2} \|X - BS\|_F^2 + \beta \sum_{i,j} \phi(S_{i,j}) \\
\text{s.t. } \sum_i B_{i,j}^2 \leq c, \forall j = 1, \cdots, n
\]

Optimization methods: the feature-sign search for solving the \(L_1\) least squares problem to learn coefficients, and a Lagrange dual method for the \(L_2\)-constrained least squares problem to learn the bases for any sparsity penalty function.

In computer vision, each column of \(X\) could represent small image patch, \(B\) dictionary of features, \(S\) sparse representation of \(X\). In compressed sensing, we attempt to recover \(S\), assume we know the dictionary \(B\). In sparse coding, we attempt to find a dictionary \(B\) so that each column of \(X\) has a sparse representation. Note that when \(B\) is overcomplete, meaning the number of basis vectors is larger then the dimensionality, both problems (sparse coding, compressed sensing) are ill-posed without the sparsity condition.

2.1 Why does sparse coding work?

Sparse coding only applies if the data actually have sparse structure. [Olshausen and Field, 1997]

How can we explain this rigorously using statistical learning theory and model selection? The brain evolved to extract sparse structure in data effectively and efficiently. How does the brain do this? We want to mimic what goes on in the brain for many real-world applications. As hidden features start having dependencies among themselves, is sparsity still effective for model selection?

Given some mild assumptions about the dictionary and code sparsity, one can show that solutions
to the sparse coding problem are unique. [Hillar and Sommer, 2011]

We say a dictionary $B$ is **incoherent** if

$$BS_1 = BS_2 \text{ for } k\text{-sparse } s_1, s_2 \in \mathbb{R}^n \rightarrow s_1 = s_2$$

(2)

The ACS Reconstruction Theorem assures us that if there is sparse structure in the data, then this structure can be recovered if we can solve the sparse coding problem optimally.

## 3 Exponential Family Sparse Coding[4]

**Motivation:** Sparse coding with the Gaussian noise distribution assumption may be too restrictive to model the wide variety of inputs that we might encounter in machine learning problems, including point clouds or depth maps, discrete data, etc. To address this problem, we generalized the Gaussian probabilistic model behind sparse coding in a principled way to include most standard distributions. We draw on the widely studied idea of the “exponential family” of distributions. This class of distributions includes the Gaussian, Bernoulli and Poisson distribution, among others, while still providing guarantees useful for efficient learning and inference. Our generalization is analogous to the way in which generalized linear models (GLMs) generalize least squares regression (which relies on a Gaussian assumption) to other kinds of regression, including logistic regression (for binary inputs) or softmax regression (for multivalued inputs). We call our model **exponential family sparse coding**, and to differentiate it from the previous model, we henceforth call that model Gaussian sparse coding.

$$P(x|b, s) = h(x) \exp(\eta^T T(x) - a(\eta)), \quad \eta = \sum_j b_j s_j$$

(3)

where we use the basis vectors $b_j$ and the activations $s_j$ to construct the natural parameter $\eta$ for the family. We can compute the maximum a posteriori estimates of the basis vectors $b_j$ and the activations $s^{(i)}$ as follows:

$$\min_{B, s^{(i)}} \sum_i -\log h(x^{(i)}) - s^{(i)^T} BT(x^{(i)}) + a(Bs^{(i)}) + \beta \sum_{i,j} |s_j^{(i)}|^2$$

(4)

$$s.t. \quad ||b_j||^2 \leq c, \forall j = 1, \cdots, n,$$

Optimization: Since the exponential family guarantees convexity of $\log P(x|\eta)$ with respect to $\eta$, we can show that the above optimization problem is convex w.r.t. $s$ for fixed $B$, and w.r.t. $B$ for fixed $s$ (though it is not jointly convex). This again suggests an alternating minimization procedure intertreating the following two steps till convergence: (1) fix $s$ and compute the optimal base $B$; and (2) fix $B$, and compute the optimal activations $s$. 

3
4 Extension to Deep Learning

4.1 Sparse Deep Belief Networks

4.2 Convolutional Deep Belief Networks

4.3 Convolutional DBNs for Audio Classification

5 Non-local Sparse Models for Image Restoration[5]

\[
\min_{W,H} ||Y - WH||_F^2 + \lambda\psi(H)
\]

In sparse coding, \( \psi \) is usually taken to be the \( l_1 \) norm, i.e. \( \psi(H) = \sum_{i,j} |h_{i,j}| \) and an alternate minimization scheme is typically employed.

If the problem has more structure, one would like to use this structure in order to learn non-local sparsity patterns. Group norm sparsity priors can be used to learn dictionaries of natural images by first clustering the training image patches, and then learning a dictionary where members of the same cluster are encouraged to share sparsity patterns. In particular, they use the \( l_{2,1} \) norm defined as \( \psi(H) = \sum_k ||h^k||_2 \), where \( h^k \) are the elements of \( H \) that are members of the \( k \)-th group.


We formulate the problem of learning motion dictionaries as the one of solving the following optimization problem.

\[
\min_{W,H} ||Y - \text{vec}(W)\text{vec}(H)||_F + \lambda\phi(W) + \eta L_{p,q,r}(W)
\]

s.t. \( \forall i, j, k \ 0 \leq H_{i,j,k} \leq 1 \), \( H_{i,j,k} = H_{i,j+1,k+1} \), \( \forall j \sum_{i,j} H_{i,j,k} \leq \delta_{\text{train}} \)

- \( Y \in \mathbb{R}^{D \times L} \), \( D \) is the dimensionality of the signal, \( L \) is the temporal length of the signal.
- \( W \in \mathbb{R}^{D \times P \times Q} \), encoding temporal and spatial information, \( P \) is the number of primitives, \( Q \) is the length of the primitives.
- \( H \in \mathbb{R}^{Q \times P \times L} \)
- \( \text{vec}(W) \in \mathbb{R}^{D \times PQ} \), \( \text{vec}(H) \in \mathbb{R}^{QP \times L} \)
- \( \phi(W) = \sum_{p=1}^{P} ||\nabla^2 W_{p,:}||_F \)
- \( L_{p,q,r} = \left( \sum_{i=1}^{P} \left( \sum_{j=1}^{Q} \left( \sum_{k=1}^{D} |W_{i,j,k}|^p \right)^{q/p} \right)^{r/q} \right)^{1/r} \)

7 CVPR 2012 Tutorial on Sparse Coding by Kai Yu

Sparse coding offers an effective building block to learn useful features.
Coding: nonlinear mapping data into another feature space.
Better coding methods: sparse coding, RBMs, auto-encoders.

Sparse coding (Olshausen & Field, 1996). Originally developed to explain early visual processing in the brain (edge detection).

Training: given a set of random patches $x$, learning a dictionary of bases $[\Phi_1, \Phi_2]$.

Coding: for data vector $x$, solve LASSO to find the sparse coefficient vector $a$.

$$\min_{a, \phi} \sum_{i=1}^{m} ||x_i - \sum_{j=1}^{k} a_{i,j} \phi_j||^2 + \lambda \sum_{i=1}^{m} \sum_{j=1}^{k} |a_{i,j}|$$  \hspace{1cm} (7)

Alternating optimization:
1. Fix dictionary $\phi_1, \cdots, \phi_k$, optimize $a$. (a standard LASSO problem)
2. Fix activations $a$, optimize dictionary $\phi_1, \cdots, \phi_k$. (a convex QP problem)

Encoding: activation $a = f(x)$ is nonlinear implicit function of $x$.

Decoding: reconstruction $x' = g(a)$ is linear and explicit.

RBM and auto encoders also involve activation and reconstruction, but they have explicit $f(x)$ and not necessarily enforce sparsity on $a$. But if sparsity is enforced on $a$, improved results can be achieved. Therefore, sparse RBMs, sparse auto encoder, even VQ can be viewed as a form of sparse coding.

### 7.1 Sparse activations vs. Sparse models

For a general function learning problem $a = f(x)$

- sparse model: parameters in $f(x)$ are sparse
  1. example: LASSO $f(x) = \langle w, x \rangle$, $w$ is sparse
  2. the goal is **feature selection**: all data selects a common subset of features.

- sparse activations: outputs of $f(x)$ are sparse
  1. example: sparse coding $a = f(x)$, $a$ is sparse
  2. the goal is feature learning: different data points activate different feature subsets

### 7.2 Sparsity vs. Locality

Intuition: similar data should get similar activated features, meaning data in the same neighborhood tend to have shared activated features. While data in different neighborhoods tend to have different features activated. However, sparse coding is not always local.

- If each basis is a “direction”, then each datum is a linear combination of only several bases, resulting in sparsity.
- If each basis is an “anchor point”, then each datum is a linear combination of neighbor anchors. In this case, sparsity is caused by locality. Local sparse coding could be achieved by local anchor points or local subspaces. Classical sparse coding is empirically local.

On MNIST dataset, when $\lambda$ is small, each basis is like a part or direction. $\lambda = 0.05$, then each basis is more like a digit. $\lambda = 0.5$, lie VQ. When sparse coding achieves the best classification accuracy, the learned bases are like digits – each basis has a clear local class association.
7.3 Hierarchical sparse coding

Hierarchical sparse coding provide even better performance than CNN. More amazingly, HSC learns features in unsupervised manner!

8 Convex Sparse Coding, Subspace Learning, and Semi-supervised Extensions[7]

8.1 Unsupervised Representation Learning

Data $X \in \mathbb{R}^{n \times m}$, where $m$ is the number of observations, $n$ is the number of dimensions. The goal is to learn a $n \times k$ dictionary $B$ containing $k$ basis vectors, and a $k \times m$ representation matrix $\Phi$ containing $m$ new feature vectors of length $k$, so that $X$ can be accurately reconstructed from $\hat{X} = B\Phi$. $L(\hat{X}, X)$ is defined to measure the approximation error, such as sum of squared error $||\hat{X} - X||_F^2$ or a sum of Bregman divergences $\sum_j D(\hat{X}_{:,j}||X_{:,j})$. Note that the factorization $\hat{X} = B\Phi$ is invariant to reciprocal rescaling of $B$ and $\Phi$, so to avoid degeneracy their individual magnitudes have to be controlled. We will assume that each column $B_{:,j}$ of $B$ is constrained to belong to a bounded closed convex set $\mathcal{B}$, hence $B \in \mathcal{B}^k$.

Subspace learning methods, such as PCA and variants seek a reconstruction matrix $\hat{X} = B\Phi$ that has reduced rank. Sparse coding, on the other hand, seek a reconstruction where each new feature vector $\Phi_{:,j}$ is sparse; that is, $\hat{X}_{:,j}$ is reconstructed from a small subset of basis vectors chosen from the dictionary $B$. For both, the generic training problem can be expressed as

$$\min_{B \in \mathcal{B}^k} \min_{\Phi} L(B\Phi; X) + \alpha||\Phi||$$

where $k$ is the number of features to be extracted. Specific choices of $L$, $\alpha$, $||\cdot||$ and $\mathcal{B}$ yield standard forms of subspace learning and sparse coding.

- PCA: $L(\hat{X}; X) = ||\hat{X} - X||_F^2$, $\alpha = 0$ and $\mathcal{B} = \{ b : ||b||_2 \leq 1 \}$

- exponential family PCA: set $L$ to Bregman divergence, $\alpha = 0$, and $\mathcal{B} = \{ b : ||b||_2 \leq 1 \}$

For subspace learning(i.e. dimensionality reduction), rather than bounding the number of columns in $B$, we allow it to grow as necessary and drop features implicitly by imposing a $||\Phi||_{2,1}$ regularizer. Such a regularizer will encourage entire rows $\Phi_{:,i}$(features) to become sparse but otherwise only smooth the columns.

For sparse coding, the goal is not to reduce dimension but instead to learn a sparse representation $\Phi$. The standard regularizer used for this purpose has been $||\Phi||_{1,1}$, which encourages entry-wise sparsity on $\Phi$.

8.2 General Formulation

9 Automatic Group Sparse Coding[8]

Different traditional spectral decomposition methods such as PCA and SVD, Sparse Coding (1) is usually additive, which results in a better representation ability; (2) does not require the learned bases to be orthogonal, which allows more flexibility to adapt the representation to the dataset. In
many real world applications, SC achieves state-of-the-art performance.

Traditional sparse coding treated each data instance as an individual and no data group information is considered.

$$\min_{F,G} \|X - FG^T\|_F^2 + \lambda \sum_{i=1}^n \|G_i\|_1$$  \hspace{1cm} (9)

where $X \in \mathbb{R}^{d \times n}, F \in \mathbb{R}^{d \times k}, G \in \mathbb{R}^{n \times k}$. Sometimes it makes more sense to learn a group level sparse representation. Thus Bengio[9] proposed Group Sparse Coding (GSC), which assumes that there are $C$ hidden groups in $X$.

$$\min_{F \geq 0, G \geq 0} \sum_{c=1}^C \|X_c - FG_c^T\|_F^2 + \lambda \sum_{i=1}^{n_c} \|G_{ci}\|_p + \gamma \sum_{j=1}^k \|F_j\|_p$$  \hspace{1cm} (10)

where $F \in \mathbb{R}^{d \times k}$ is a common shared dictionary over all groups, $G_c \in \mathbb{R}^{n_c \times k}$ is the coding coefficient matrix for group $c$. $G_{ci}$ is the $i$-th row of $G_c$, $F_j$ is the $j$-th column of $F$.

GSC learns a sparse representation on group level as well as a shared dictionary. However, GSC assumes the data group identities are pre-given and it can only learn a common dictionary. However, in many real word applications, (1) the data group identities are hidden and (2) we want to know the group-specific dictionaries, as these individual dictionaries can help us to capture the discrimination information contained in different data groups.

$$\min_{F^S \geq 0, G^I \geq 0} \sum_c \|X_c - F^S G^S_c - F^I_c G^I_c\|_F^2 + \sum_c [\gamma^I \phi(G^I_c) + \gamma^S \phi(G^S_c)]$$  \hspace{1cm} (11)

s.t. $F^S \geq 0, \forall c = 1, \cdots, C, F^I_c \geq 0, G^I_c \geq 0, G^S_c \geq 0$

where $G^S_c \in \mathbb{R}^{n_c \times k^S_c}$ is the reconstruction coefficient matrix on the group-shared dictionary $F^S$. $G^I_c$ is the reconstruction coefficient matrix on $c$-th group-specific dictionary $G^I_c \in \mathbb{R}^{n_c \times k^I_c}$.

References


