Stable and Efficient Representation Learning with Nonnegativity Constraints

Tsung-Han Lin* THLIN@EECS.HARVARD.EDU
H. T. Kung KUNG@HARVARD.EDU
School of Engineering and Applied Sciences, Harvard University, Cambridge MA, USA

Abstract

Orthogonal matching pursuit (OMP) is an efficient approximation algorithm for computing sparse representations. However, prior research has shown that the representations computed by OMP may be of inferior quality, as they deliver suboptimal classification accuracy on several image datasets. We have found that this problem is caused by OMP's relatively weak stability under data variations, which leads to unreliability in supervised classifier training. We show that by imposing a simple nonnegativity constraint, this nonnegative variant of OMP (NOMP) can mitigate OMP's stability issue and is resistant to noise overfitting. In this work, we provide extensive analysis and experimental results to examine and validate the stability advantage of NOMP. In our experiments, we use a multi-layer deep architecture for representation learning, where we use K-means for feature learning and NOMP for representation encoding. The resulting learning framework is not only efficient and scalable to large feature dictionaries, but also is robust against input noise. This framework achieves the state-of-the-art accuracy on the STL-10 dataset.

1. Introduction

We consider computing high-level image representations with which we can more easily classify images. Such high-level representations are typically derived by encoding low-level image descriptors into a suitable feature space based on a feature dictionary. Much work has been devoted to unsupervised feature dictionary learning over the past years (see Bengio et al., 2013). Recently, it has been shown that the K-means algorithm is usually sufficient for this task (Coates & Ng, 2011a), providing a very efficient solution for dictionary learning.

On the contrary, efficient encoder design for computing data representations based on learned dictionaries has received less attention. A good encoder usually finds representations that are *sparse*, with the hope that the new representations are linearly separable in the feature space and will simplify classifier training. Imposing this sparse prior, however, often invokes a considerable amount of computations. For example, the classical approach to sparse coding involves solving an expensive ℓ_1 minimization problem (Lee et al., 2006; Raina et al., 2007), which is less applicable for large-scale machine learning problems.

There have been several attempts to use efficient approximation algorithms for sparse encoding (see Coates & Ng, 2011a). One example is the soft-threshold encoder, which finds sparse representations by simply dropping entries smaller than a certain threshold (Nair & Hinton, 2010; Kavukcuoglu et al., 2010). Such encoder has been shown to work well in benchmarks containing abundant labeled training samples. In contrast, efficient greedy algorithms, such as Orthogonal Matching Pursuit (OMP) (Pati et al., 1993), are less successful in computing effective representations. OMP is reported to deliver suboptimal classification accuracy on popular benchmarks.

In this work, we show that OMP in fact is *not* a poor encoder. We have found that a key to making OMP perform well is to introduce *nonnegativity constraints*. Nonnegativity constraints have long been exploited for learning sparse, additive features. For example, nonnegative matrix factorization (NMF) has been shown to learn parts-based representations (Lee & Seung, 1999). By further including sparseness constraints into NMF, it has been observed that Gabor-like low-level features can be learned (Hoyer, 2004). In addition, nonnegativity contraints are biologically plausible for modeling human vision systems in computational neuroscience research (Hoyer, 2003). However, despite the large corpus of nonnegative feature learning algorithms in the literature, little is known about the utility of nonnegativity constraints in encoding sparse representations.

We found that imposing nonnegativity constraints can

Proceedings of the 31^{st} International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32.

*T.-H. Lin is now with Intel Labs, Santa Clara CA, USA.

largely alleviate a stability issue of OMP, namely that OMP may fail to find nearby representations for data with small variations (Donoho et al., 2006; Rozell et al., 2008). The instability of the computed representations can lead to confusions in classifier training and inferior classification accuracy. We argue that under nonnegativity constraints, OMP's stability is enhanced, and in addition, will increase with pairwise separation among dictionary atoms. This means that with a better trained dictionary where atoms are well separated, the encoder can be much more stable.

We have validated the effectiveness of the nonnegative OMP encoder (NOMP), a variant of OMP that is as efficient, through experiments on the CIFAR-10, CIFAR-100 (Krizhevsky & Hinton, 2009), and STL-10 datasets (Coates et al., 2011). We present two major findings:

- The proposed NOMP encoder outperforms the prior OMP encoder in classification accuracy by large margins. Like prior sparsity-seeking encoders such as OMP, NOMP can tackle datasets containing a small amount of labeled training data. In contrast, to achieve comparable accuracy performance, other fast feed-forward encoders, such as the soft-threshold encoder, would have to use supervised classifier training involving substantially more labeled training data.

- With a moderate amount of labeled training samples, NOMP is competitive in classification accuracy with the state-of-the-art deep neural networks, and is much faster and easier to train.

2. Related Work

Sparse coding is a promising method for object classification (e.g., Ranzato et al., 2007). Coates and Ng (2011a) point out that the effectiveness of sparse coding is contributed largely by its encoding capability that finds sparse data representations. However, to encourage sparsity in representations, solving the related ℓ_1-minimization problem can be computationally expensive. A considerable amount of work is dedicated to designing efficient ℓ_1-minimization algorithms (e.g., Lee et al., 2006).

For computational efficiency, researchers have also developed fast nonlinear encoders, such as the $tanh$ function, to compute sparse solutions. In particular, these nonlinear encoders may be trained to approximate solutions computed by ℓ_1-sparse coding (Kavukcuoglu et al., 2008; Gregor & LeCun, 2010). Moreover, it has been shown that the simple soft-threshold encoder, $\max(0, \mathbf{D}^T\mathbf{x} - \alpha)$ for some small $\alpha > 0$, can be competitive in some cases (Nair & Hinton, 2010; Kavukcuoglu et al., 2010; Coates & Ng, 2011a). In this work, we take a different path in which OMP is employed to encode sparse representations.

Nonnegative matrix factorization (Paatero & Tapper, 1994; Lee & Seung, 1999) and nonnegative sparse coding (Hoyer, 2004) are related feature extraction methods that enjoy much empirical success. While their use is often motivated by the nonnegative nature of applications (for example, document analysis), theoretical studies suggest that nonnegativity constraints themselves can be powerful. It has been shown that nonnegativity constraints can ensure a unique sparse solution without ℓ_1-regularization (Donoho & Stodden, 2003; Bruckstein et al., 2008). Slawski and Hein (2011) further show that thresholded nonnegative least squares can be resistant to overfitting of noise even in underdetermined sparse recovery.

Following this line of research, we show that nonnegativity constraints can be useful when efficient approximation algorithms such as OMP are used in encoding sparse representations, especially for image classification purposes. We are not the first to propose a nonnegative variant for OMP. In fact, nonnegative extensions have been repeatedly proposed in the literature (Bruckstein et al., 2008; Sindhwani & Ghoting, 2012). However, as far as we know, we are the first to identify and analyze the stability advantage of nonnegativity constraints for OMP.

3. Encoding Sparse Representation with Nonnegativity Constraints

Suppose that we are given a feature dictionary of n atoms (column vectors) and a data vector. OMP encodes data representations by selecting a small number k of the atoms, such that their linear combination best approximates the data vector. Its selection procedure only needs k successive iterations: in each iteration, the atom that can maximally reduce the residual error is selected. Such a greedy iterative solver, however, can be sensitive to data variations. The greedy selection process can amplify small differences in data and lead to large deviations in their representations.

In this section, we introduce a variant of OMP, named nonnegative OMP (NOMP), and show its improved stability in computing data representations. Throughout this work, we learn feature dictionaries using the spherical K-means algorithm (also known as "gain shape" vector quantization) (Coates & Ng, 2011a) unless otherwise noted.

3.1. Nonnegative OMP

Given a nonnegative dictionary $\mathbf{D} \in \mathbb{R}^{m \times n}$ and a nonnegative data vector $\mathbf{x}$, NOMP finds an approximate solution to the following nonnegatively constrained problem:

$$\min_{\mathbf{z}} \|\mathbf{x} - \mathbf{D}\mathbf{z}\|_2 \ s.t. \ \|\mathbf{z}\|_0 \leq k, \ z_i \geq 0 \ \forall i \qquad (1)$$

That is, we would like to find sparse nonnegative coefficients $\mathbf{z} \in \mathbb{R}^n$ that can approximately reconstruct the data

$\mathbf{x}$ using the corresponding k dictionary atoms, where k is a relatively small integer. NOMP iterates the following steps for up to k rounds:

1. Initialize the residual vector $\mathbf{r}^{(0)} = \mathbf{x}$ and round number $l = 1$. Select the atom $\mathbf{d}_{i_l}$ that has the highest positive correlation with the residual, $i_l = \arg\max_i \left\langle \mathbf{d}_i, \mathbf{r}^{(l-1)} \right\rangle$. Terminate if the largest correlation is less than or equal to zero.

2. Approximate the coefficients of the selected atoms by nonnegative least squares.
$\mathbf{z}^{(l)} = \arg\min_{\mathbf{z}} \left\| \mathbf{x} - \sum_{h=1}^{l} \mathbf{d}_{i_h} z_{i_h} \right\|_2 \; s.t. \; z_{i_h} \geq 0$

3. Compute the new residual $\mathbf{r}^{(l)} = \mathbf{x} - \mathbf{D}\mathbf{z}^{(l)}$. Increment l by 1.

While following the high-level iterative structure of OMP, NOMP uses two special mechanisms. First, NOMP selects the atom that has the highest *positive* correlation with the residual, in contrast to OMP which considers both positive and negative correlations. NOMP may exit iterations early if there are no more atoms with positive correlations. Second, NOMP computes the sparse code using nonnegative least squares instead of conventional unconstrained least squares. Note that solving nonnegative least squares is considerably more expensive than solving its unconstrained variant. Empirically, we usually find it sufficient to approximate the solution by solving unconstrained least squares and truncating any resulting negative coefficients to zero.[1]

Given the structural similarity between NOMP and OMP, existing efficient OMP implementations, such as batch OMP (Rubinstein et al., 2008), can easily be adopted by NOMP. These implementations usually exploit both the sparsity in coefficients and incremental updates between iterations. With a large dictionary and small k, the overall computation required is dominated by computing a single round of atom correlations $\mathbf{D}^T\mathbf{x}$. Note that the computation of least squares is not the dominating cost. In this case, NOMP has a running time comparable to other similar encoders, including OMP and soft-threshold encoders.

3.2. Nonnegative OMP as an Encoder

To use NOMP as an encoder, we need to ensure the nonnegativity of both dictionary and input.[2] We define a nonlinear mapping $S : \mathbb{R}^{\frac{m}{2}} \rightarrow \mathbb{R}^{m}_{\geq 0}$ that transforms the input data $\mathbf{x}^{in} \in \mathbb{R}^{\frac{m}{2}}$ into a nonnegative vector $\mathbf{x}$ that is double-sized,

[1] Although truncating the negative coefficients may result in the residual vector having nonzero correlations with the selected atoms, these correlations must be negative. The selected atoms thus will not be re-selected in later iterations, and NOMP's convergence property is not affected.

[2] Although pixel intensities are nonnegative, data preprocessing such as mean subtraction can generate negative values.

Table 1. A comparison in data dimensionality between unconstrained encoders and NOMP to compute a length-n feature vector from a length-$m/2$ data vector.

	$\mathbf{x}^{in}$	$\mathbf{x}$	$\mathbf{D}$	$\mathbf{z}$
UNCNSTRN. ENCODERS	$\frac{m}{2}$	$\frac{m}{2}$	$\frac{m}{2} \times \frac{n}{2}$	$\frac{n}{2}$
NOMP	$\frac{m}{2}$	m	$m \times n$	n

$S(\mathbf{x}^{in}) = [\max(0, \mathbf{x}^{in}), \max(0, -\mathbf{x}^{in})]$ where 0 denotes the zero vector with all its components being zero. For example, a length-2 data vector $[1, -1]$ is transformed to a length-4 vector $[1, 0, 0, 1]$. This transformation has been used in modeling the receptive fields in human vision systems (Hoyer, 2003). Given nonnegative data, the K-means algorithm ensures a nonnegative dictionary will be learned.

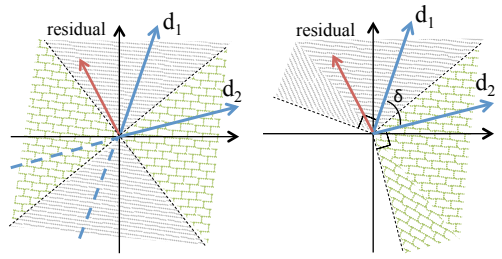
Interestingly, prior research has observed that applying this sign splitting transformation with other unconstrained encoders leads to improved classification results (Ngiam et al., 2011; Coates & Ng, 2011a). This splitting, however, is applied *after* the encoding step, for weighting the positive and negative feature vector values differently in a classifier. In this case, unconstrained encoders can be viewed as nonnegative encoders with a dictionary $[\mathbf{D} \; -\mathbf{D}]$. NOMP generalizes this formulation by using a double-sized nonnegative dictionary that has no such special symmetric structures, and can be expected to be more powerful in classification. Nevertheless, we will see that the advantage of NOMP is beyond this generalization. Table 1 compares the data dimensionality in unconstrained encoders and NOMP.

Note that the nonnegative formulation allows only additive features, and cannot express cancellation between features efficiently. For image data, this limitation is less of a problem. The classical NMF result suggests that the nonnegativity constraint can lead to parts-based representations (Lee & Seung, 1999). For deep, high-level representations, the nonnegativity constraint in fact is preferred, since nonzero entries in the representations correspond to activations of low-level features, and the cancellation between low-level features would be less meaningful.

3.3. Stability of Nonnegative OMP Under Noisy Data

To use sparse representations for classification, it is important that the data representations are stable under expected small data variations. Unstable data representations can confuse supervised classifier training and result in poor classification performance. In this section, using noise as a proxy for small data variations, we assess the robustness of an encoder, and argue that a robust encoder is more stable under these data variations.

OMP is known to obey a local stability under noise (Donoho et al., 2006). That is, OMP can tolerate suffi-



(a) OMP (b) NOMP

Figure 1. Grey(green)-shaded area denotes where the residual vector sits that the algorithm would select $d_1(d_2)$ as the next atom. NOMP can tolerate larger variations in the residual.

ciently small data noise and still find a sparse representation with the same support (the same nonzero entries). Figure 1(a) illustrates this local stability. Suppose we have two atoms $\mathbf{d}_1$ and $\mathbf{d}_2$ in the dictionary. Given the residual vector shown in the figure, OMP would select $\mathbf{d}_1$ as the next atom because the projection of the residual vector onto $\mathbf{d}_1$ is larger than its projection onto both $\mathbf{d}_2$ and $-\mathbf{d}_2$. This selection procedure allows the residual to be affected by small noise. If this deviation is small enough such that the deviated residual does not fall out of the shaded area, the same atom $\mathbf{d}_1$ will still be selected by OMP. However, a slightly larger noise may cause OMP to select $-\mathbf{d}_2$ as the next atom, and subsequently the computed representation may differ by a large error due to a different support set.

In contrast, NOMP can tolerate a larger noise as illustrated in Figure 1(b). In NOMP, only the projections of the residual onto positive $\mathbf{d}_1$ and $\mathbf{d}_2$ are considered, giving a larger noise-tolerant area. Denoting the angle separating $\mathbf{d}_1$ and $\mathbf{d}_2$ as δ and considering the same residual vector, the noise-tolerant area for NOMP to choose $\mathbf{d}_1$ spans an angle of $\pi/2 + \delta/2$, larger than OMP's $\pi/2$. This also suggests that NOMP's noise-tolerant region grows when the two dictionary atoms are further separated, while OMP's noise-tolerant region has a fixed size no matter how the angle between atoms is varied.

Formally, the following theorem shows that NOMP can tolerate sufficiently small noise in data and computes representations with the same support.

Theorem 1. *Suppose a data vector* $\mathbf{x}$ *has a nonnegative* k*-sparse representation* $\mathbf{z}$ *using a nonnegative dictionary* $\mathbf{D}$*, i.e.,* $\mathbf{x} = \mathbf{D}\mathbf{z}$*. Given a noisy data vector* $\mathbf{x} + \mathbf{n}$*, NOMP finds a sparse representation that has the same support as* $\mathbf{z}$ *if the noise* $\mathbf{n}$ *satisfies*

$$\|\mathbf{n}\|_2 < \tfrac{\sqrt{2}}{2}(1 - \mu k) z_{min} \tag{2}$$

where μ *is the coherence of the dictionary, or the maximum correlation between any two atoms in the dictionary, and* z_{min} *is the smallest nonzero entry in* $\mathbf{z}$*.*

Proof. We begin the proof by considering the first iteration in NOMP. Assuming the $\mathbf{z}$'s k nonzeros are located in the first k entries in descending order of magnitudes, for NOMP to select a correct nonzero entry, we need

$$\max_{1 \le h \le k} \langle \mathbf{x} + \mathbf{n}, \mathbf{d}_h \rangle > \max_{h > k} \langle \mathbf{x} + \mathbf{n}, \mathbf{d}_h \rangle \tag{3}$$

We can bound both sides of (3):

$$\begin{aligned} \langle \mathbf{x} + \mathbf{n}, \mathbf{d}_1 \rangle &= z_1 + \sum\nolimits_{i=2}^{k} z_i \langle \mathbf{d}_i, \mathbf{d}_1 \rangle + \langle \mathbf{n}, \mathbf{d}_1 \rangle \\ &\ge z_1 + \langle \mathbf{n}, \mathbf{d}_1 \rangle \end{aligned} \tag{4}$$

$$\begin{aligned} \langle \mathbf{x} + \mathbf{n}, \mathbf{d}_h \rangle &= \sum\nolimits_{i=1}^{k} z_i \langle \mathbf{d}_j, \mathbf{d}_h \rangle + \langle \mathbf{n}, \mathbf{d}_h \rangle \\ &\le z_1 \mu k + \langle \mathbf{n}, \mathbf{d}_h \rangle \end{aligned} \tag{5}$$

Combining (3)-(5) yields

$$\langle \mathbf{n}, \mathbf{d}_h \rangle - \langle \mathbf{n}, \mathbf{d}_1 \rangle < z_1(1 - \mu k) \tag{6}$$

Note that all the atoms are nonnegative. This allows us to further bound the left-hand side of (6).

$$\langle \mathbf{n}, \mathbf{d}_h \rangle - \langle \mathbf{n}, \mathbf{d}_1 \rangle \le \|\mathbf{n}\|_2 \|\mathbf{d}_h - \mathbf{d}_1\|_2 \le \sqrt{2} \|\mathbf{n}\|_2 \tag{7}$$

Swapping (7) into (6) gives us a bound for the noise that NOMP selects a correct nonzero entry in the first iteration.

$$\|\mathbf{n}\|_2 < \tfrac{\sqrt{2}}{2}(1 - \mu k) z_1 \tag{8}$$

We can repeatedly apply the same procedure to derive bounds in later NOMP iterations. In the l-th iteration, we can find the following bound analogous to (8).

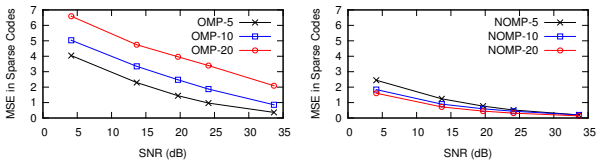
$$\|\mathbf{n}\|_2 < \tfrac{\sqrt{2}}{2}(1 - \mu k) z_l \tag{9}$$

Therefore, satisfying the k conditions derived in the k NOMP iterations guarantees finding the correct support set, and (2) suffice to satisfy all k conditions. □

This upper bound for tolerable noise is larger than OMP's bound by a factor of $\sqrt{2}$, previously derived using the same technique (Donoho et al., 2006). We note that this bound is loose and empirically NOMP can tolerate even larger noise. We will provide empirical results in Section 4.

3.4. Improving Stability with Multiple Dictionaries

We have seen that NOMP enjoys a stronger stability than OMP. However, fundamentally, the stability of greedy pursuit algorithms is limited by the *coherence* of feature dictionaries, as strongly correlated atoms in the dictionary can cause more unstable atom selections. In practice, it is not easy to ensure all dictionary atoms to be equally separated, suggesting that NOMP's encoding will be particularly unstable to data related to strongly correlated atoms. A simple



(a) OMP, MSE of codes (b) NOMP, MSE of codes

Figure 2. Impact of noise on the stability of OMP and NOMP. NOMP is more stable under noise and shows less overfitting.

strategy to mitigate this problem is to make use of multiple separate dictionaries such that it is unlikely that a data input will have unstable representations across the majority of the dictionaries. One can then train multiple separate classifiers, each corresponding to one dictionary, and use a majority vote to combine the predictions.[3][4]

4. Empirical Validation of NOMP's Stability

In this section, we empirically validate NOMP's stability for data under noise, and under variations. We use a dictionary learned from 6×6 images patches from CIFAR-10.

For noisy data, we sample 10,000 image patches $\mathbf{x}$ from CIFAR-10, and generate 2,500 noisy versions $\mathbf{x}^*$ of each patch with different Gaussian noise. Both clean and noisy samples, $\mathbf{x}$ and $\mathbf{x}^*$, are encoded to sparse codes $\mathbf{z}$ and $\mathbf{z}^*$, respectively. We measure the mean-squared-error (MSE) of sparse codes $\|\mathbf{z}^* - \mathbf{z}\|_2$ to assess the stability of the encoders. The sparsity bound k of both OMP and NOMP are varied during the experiments, denoted as (N)OMP-k.

As shown in Figure 2(a), the MSE of sparse codes computed by OMP grows with a larger k. Using more atoms to approximate the input runs the risk of overfitting to data noise, and consequently leads to more unstable sparse codes. In contrast, in Figure 2(b), NOMP finds sparse codes with smaller MSE across all SNRs. In fact, the MSE even *drops* with a larger k due to the fact that NOMP would terminate by itself when no more additive atoms can be found, effectively reducing overfitting. This result suggests NOMP-20 potentially is a better encoder than OMP-5.

Next we test the stability of sparse codes for images of grating under small rotations. We generate 8,000 6×6 images of grating and rotate each image by some small angle.[5] For each pair of grating and its rotation, we compare the similarity between their sparse codes with the normalized correlation. As shown in Table 2, the codes computed by NOMP are more stable under small rotations while the codes computed by OMP quickly become very different.

Table 2. Stability of the codes of grating images under rotations.

ANGLE	0	0.01π	0.02π	0.03π	0.04π
OMP-5	1.00	0.71	0.54	0.43	0.34
NOMP-20	1.00	0.92	0.80	0.68	0.57

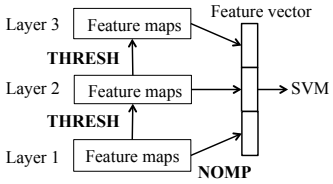


Figure 3. The learning architecture adopted in this work. Note that we use different encoders to compute representations for classifier (NOMP) and for higher-level encoding (soft-threshold).

5. A Multi-Layer Learning Framework for Classification with NOMP

We adopt a popular architecture that stacks multiple layers of convolutional feature encoders (Lee et al., 2009; Coates & Ng, 2011b). At each layer, overlapping patches from the input feature maps are encoded using a feature dictionary. The computed representations are then pooled (max or average) over a small neighborhood to generate feature maps for further encoding in the next layer, or pooled over the whole image to form an image representation.

Standard preprocessing steps are applied on image data to generate data vectors for layer-1. These include mean subtraction, contrast normalization , and ZCA-whitening, followed by sign-splitting as described in Section 3.2.

However, unlike the popular architecture, we use different encoding methods to compute feature vectors for classification, and compute feature maps for higher-layer encoding, as illustrated in Figure 3. In particular, we use NOMP to compute sparse representations for classification, and a soft-threshold function to generate sparse codes for higher layers.[6] We do so because feature maps computed with NOMP are very sparse and are difficult to be further encoded efficiently, and therefore use the soft-threshold function for less-sparse representations. Empirically, we found that using a soft-threshold function that truncates 90% coefficients works well. Note that between layers, only a fea-

[3]Note that this technique does not improve the classification result when using the soft-threshold encoder. The soft-threshold encoder is stable regardless of the properties of dictionaries.

[4]In our implementation, we learn separate dictionaries by using different initializations to K-means. The predictions reported in Section 6 are combined from 7 dictionaries and classifiers.

[5]The grating is generated by $I(x, y) = b + a\sin(\omega(x\cos\theta + y\sin\theta - \phi))$ where ω is the spatial frequency, θ is the orientation, and ϕ is the phase (Berkes & Wiskott, 2005). We set $\omega = \{0.5, 1, 1.5, 2\}$ and $\phi = 0$ to π with a step size $\pi/20$.

[6]Note that this adds only very little computation, as $\mathbf{D}^T\mathbf{x}$ is already computed by NOMP.

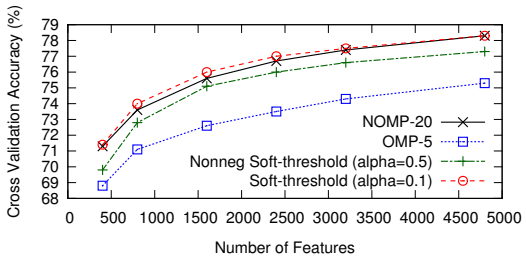


Figure 4. Single-layer classification accuracy on full CIFAR-10 with abundant training samples (5000 labeled samples per class).

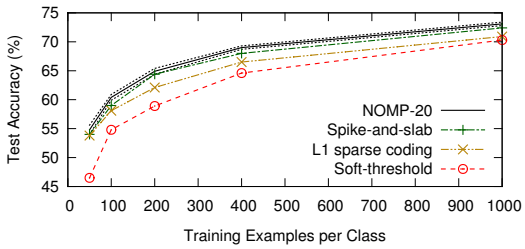


Figure 5. Single-layer classification accuracy on CIFAR-10 with fewer training samples (less than 1000 labeled samples per class). In this experiment, we use a dictionary of 3200 features. Only NOMP's standard error is shown for the readability of the figure.

ture vector normalization step is performed. This makes the learning framework very simple as compared to other existing frameworks, which require some form of data whitening (Coates & Ng, 2011b; Hui, 2013).

For even faster computation, the nonnegativity constraint allows us to enforce a *sparse* high-layer dictionary, given that high-layer inputs are sparse and only additive atoms are allowed. By exploiting the sparsity in computations, both training and encoding can be made significantly faster. A simple strategy to enforce sparse dictionaries is to drop entries with small values. We typically set atoms to have only 10% nonzeros for layer-2 and above. This shows no harm to classification accuracy in our experiments. Finally, the representations computed at different layers are concatenated as a image feature vector for use in classification, for which we employ a linear classifier (L2-SVM).[7]

6. Validating NOMP with Classification

6.1. Performance on the CIFAR-10 Dataset

6.1.1. Single Layer Performance

We first evaluate NOMP using the full CIFAR-10 dataset with a single layer encoder. CIFAR-10 is a dataset with abundant labeled training samples (5000 for each class). We encode 6×6 patches, pool the sparse codes over the four quadrants of an image, and concatenate the four representations. For comparison purposes, this architecture is identical to those used in the literature (Coates & Ng, 2011a; Goodfellow et al., 2012).[8] We compare NOMP with both OMP and the soft-threshold encoder, one of the best known encoders for CIFAR-10. We choose k as 20 and 5 for NOMP and OMP, respectively, in the experiments.[9]

As shown in Figure 4, OMP achieves the worst accuracy despite being a sparse encoder. As a point for comparison, (Coates & Ng, 2011a) uses ℓ_1-sparse coding and reports a 78.5% accuracy with 3200 features (compared to 74.5% achieved by OMP). This suggests that OMP, although very efficient, does not find good representations. With nonnegativity constraints, NOMP is able to approach the accuracy of ℓ_1-sparse coding (77.4%). This makes NOMP an attractive encoder, especially for its high computational efficiency as compared to ℓ_1-sparse coding.

Second, NOMP achieves accuracy comparable with the soft-threshold encoder in the full CIFAR-10 dataset. Classification accuracy with soft-threshold-encoded representations, however, is only competitive under a large amount of labeled training samples (Coates & Ng, 2011a). In contrast, representations encoded by sparse encoders such as NOMP do not need as many samples. In Figure 5, we reduce the amount of labeled training samples in CIFAR-10, and compare NOMP to other encoders using the accuracy numbers reported by (Goodfellow et al., 2012). In this case, NOMP achieves the highest accuracy of the group when there are less than 1000 labeled samples per class, outperforming the soft-threshold encoder, the ℓ_1-sparse coding, and even the more sophisticated spike-and-slab sparse coding.

6.1.2. Quantifying the Impact of Nonnegative Training and Nonnegative Encoding

Having shown that NOMP not only delivers classification accuracy higher than OMP but also competitive with other well-known encoders, we seek to understand why NOMP performs well. NOMP differs with OMP in two ways: (1) Nonnegative training learns a more flexible dictionary by separating positive and negative channels (see Section 3.2). (2) Nonnegative encoding enjoys a stronger stability than

[7] The feature vectors are standardized by rescaling the values to [0, 1] for each dimension. Note that this preserves the sparsity of feature vectors due the nonnegativity.

[8] Accuracy is optimized over max- and average-pooling; generally, NOMP performs best with max-pooling, and the soft-threshold encoder with average pooling.

[9] This choice is cross-validated over $k = \{1, 3, 5, 10, 20\}$. In general, choosing k is not difficult. For NOMP, overfitting is less of a problem, as the algorithm would terminate early by itself. As such, it is safe to use a large k, and this tends to improve performance. For OMP, setting k is trickier due to the trade-off between representational power and overfitting. Usually a small k (such as 5 or 10) works best for OMP. See Section 6.1.2.

Table 3. Single-layer classification accuracy of CIFAR-10 using various training and encoding methods. We report accuracy from the 5-fold cross validation on the training set.

TRAIN	ENCODING			
	OMP-5	OMP-20	NOMP-5	NOMP-20
R	69.2	67.3	69.2	74.6
RP	72.9	71.6	74.6	77.1
UNSPLIT	74.5	73.6	74.9	76.3
SPLIT	73.5	73.2	75.7	77.4

OMP (see Section 3.3). To measure individual impact of nonnegative training versus nonnegative encoding, we construct a dictionary $\mathbf{D_n}$ using a dictionary $\mathbf{D}$ trained from unconstrained K-means, and pair this dictionary with nonnegative encoding. $\mathbf{D_n}$ is constructed as follows to have a special symmetric structure as we do with unconstrained encoding:

$$\mathbf{D_n} = \begin{bmatrix} \max(0, \mathbf{D}) & \max(0, -\mathbf{D}) \\ \max(0, -\mathbf{D}) & \max(0, \mathbf{D}) \end{bmatrix} \tag{10}$$

In this experiment, we also include dictionaries formed by random numbers (R) and randomly selected patches (RP) for comparisons. The sparsity k of both OMP and NOMP for encoding is also varied to examine the impact of encoder stability on classification accuracy.[10]

We can make several observations from the results shown in Table 3. First, nonnegative encoding seems to contribute to most of the success of NOMP, and nonnegative training, in contrast, plays a minor role. We see higher classification accuracy even when nonnegative training is replaced by unconstrained training (76.3%). Across all dictionary training methods, encoding with NOMP-20 consistently improves classification accuracies by a significant margin.

Second, the stability of the encoders is strongly correlated with classification accuracy, and this explains why nonnegative encoding is particularly beneficial. In Section 4, we observed that a large k in OMP results in unstable representations. Correspondingly, we see OMP-20 delivers lower accuracy than OMP-5. In contrast, with NOMP, a larger k gives more stable representations, and we see the accuracy improves from NOMP-5 to NOMP-20. Note that these observations hold true regardless of the employed dictionary training method, suggesting that nonnegative encoding for improved encoder stability is of fundamental importance.

6.1.3. MULTI-LAYER PERFORMANCE

We next examine NOMP's performance in a multi-layer, deep architecture as described in Section 5. For comparison, the settings of the architecture are identical to (Coates & Ng, 2011b), where the authors stack multiple layers of soft-threshold encoders.[11]

Table 4. CIFAR-10 test accuracy in a multi-layer architecture.

	400 EX/CLASS	FULL DATA
OMP-5 (1 LAYER)	67.2 ± 0.3	75.2
OMP-5 (2 LAYERS)	67.9 ± 0.4	76.4
NOMP-20 (1 LAYER)	69.0 ± 0.3	78.0
NOMP-20 (2 LAYERS)	71.3 ± 0.4	80.9
NOMP-20 (3 LAYERS)	71.7 ± 0.3	81.4
NOMP-20 (3 LAYERS + MULTI DICT)	$\mathbf{72.2 \pm 0.4}$	**82.9**
RF LERANING (3 LAYERS) (COATES & NG, 2011B)	70.7 ± 0.7	82

As shown in Table 4, we can see that with NOMP, the accuracy can be improved effectively by simply adding more layers (78.0% to 81.4% in the full dataset, and 69.0% to 71.7% in the reduced dataset). We note that the only other known higher classification accuracy for the reduced CIFAR-10 is 72.6% (Hui, 2013), in which the accuracy is attained by exploiting view-invariant features rather than a better encoder design. Finally, exploiting multiple dictionaries can further improve the classification accuracy.

The accuracy improvement in stacked OMP, however, is relatively small. This may suggest that the nonnegative constraint is also advantageous for high-layer feature encoding. To evaluate the impact of nonnegative encoding in higher layers, we run another experiment that uses only the representations derived at layer-2 for classification.[12] In addition, we include a case where the nonnegative constraint is *only* enforced in layer-2. This allows us to isolate the impact of nonnegativity on layer-2.

Table 5 shows that two-layer OMP alone in fact achieves very poor accuracy (67.3%). Surprisingly, by only adding nonnegativity on layer-2, the accuracy can be drastically improved (77.2%) to almost the same as that in two-layer NOMP (77.3%). We hypothesize that the improvement is a result of avoiding unwanted cancellations between high-level features. A nonzero value in a high-level feature can be interpreted as the "presence" of the corresponding low-level feature. Therefore, cancellations between positive and negative values is less meaningful. Adding the nonnegativity constraint eliminates this possibility and again, prevents overfitting in the model.

We note that the current state-of-the-art accuracy of full CIFAR-10 is achieved by deep neural networks (e.g.,

[10] In these experiments we use features that have $3200 \times 4 = 12800$ dimensions for the SVM.

[11] The patch sizes and features sizes are 6×6, 9×9, 15×15, and 3200, 6400, 6400 for the three layers, respectively.

[12] Instead of concatenating representations found at all layers as described in Section 5.

Table 5. Accuracy from 5-fold cross validation on full CIFAR-10 training set, using only representations constructed in layer-2. T denotes soft-threshold encoding, and NT denotes soft-thresholding with nonnegative sign splitting.

	ACCURACY
LAYER-1 (T) + LAYER-2 (OMP)	67.3
LAYER-1 (T) + LAYER-2 (NOMP)	77.2
LAYER-1 (NT) + LAYER-2 (NOMP)	77.3

Table 6. Classification accuracy of CIFAR-100.

	TEST ACC.
OMP-5 (1 LAYER)	49.0
NOMP-20 (1 LAYER)	53.3
NOMP-20 (3 LAYERS)	57.7
NOMP-20 (3 LAYERS + MULTI DICT)	60.8
STOCHASTIC POOLING (ZEILER & FERGUS, 2013)	57.5
MAXOUT (GOODFELLOW ET AL., 2013)	**61.4**

Goodfellow et al., 2013). The power of such methods, however, depends on the amount of available labeled training samples. As we will see in the next section, NOMP is very competitive when labeled training data is limited.

6.2. Performance on the CIFAR-100 Dataset

The strength of NOMP lies in its ability to tackle datasets with limited labeled training data. The CIFAR-100 dataset is one of such datasets: it has many more classes (100 classes), and fewer labeled training samples per class (500 samples for each class), as compared to the CIFAR-10 dataset (5000 samples per class). We use the same hyper-parameters in this experiment as used in the CIFAR-10 experiments. As shown in Table 6, 3-layer NOMP achieves a very competitive accuracy (57.7%). Further, exploiting multiple dictionaries has a big impact. The accuracy can be largely improved (60.8%) and approaches the state-of-the-art accuracy achieved by maxout networks, an advanced extension of deep neural networks with dropout training.

6.3. Performance on the STL-10 Dataset

Finally, we evaluate NOMP on the STL-10 dataset, which features very few labeled training examples (100 examples for each of the 10 classes) and larger 96×96 images. Due to its relatively large image size, much prior research chose to downsample the images to 32×32. We examine NOMP's performance on both the downsampled and original-sized dataset.[13] The results are shown in Table 7.

[13]For the downsampled dataset, we use the same setting for the multi-layer framework as in CIFAR-10. For the original dataset, we use 10×10, 19×19, and 38×38 patches for layer-1, layer-2 and layer-3, respectively. 2×2 max-pooling is inserted between layers. For both experiments, we use feature size 3200, 6400, and 6400 for the three layers, respectively.

Table 7. Classification accuracy of STL-10.

	TEST ACC.
THRESH (1 LAYER, DOWNSAMPLED)	54.8 ± 0.4
OMP-5 (1 LAYER, DOWNSAMPLED)	58.1 ± 0.5
NOMP-20 (1 LAYER, DOWNSAMPLED)	59.0 ± 0.5
NOMP-20 (2 LAYERS, DOWNSAMPLED)	60.4 ± 0.5
NOMP-20 (1 LAYER)	64.6 ± 0.6
NOMP-20 (3 LAYERS)	67.5 ± 0.5
NOMP-20 (3 LAYERS + MULTI DICT)	$\mathbf{67.9 \pm 0.6}$
SPARSE CODING (1 LAYER, DOWNSAMPLED) (COATES & NG, 2011A)	59.0 ± 0.8
RF LERANING (3 LAYERS, DOWNSAMPLED) (COATES & NG, 2011B)	60.1 ± 1
HMP (BO ET AL., 2012)	64.5 ± 1

First, we note that the classification accuracy follows similar trends as in CIFAR-10. With a single layer, NOMP achieves accuracy similar to ℓ_1-sparse coding. Using two layers of NOMP, the accuracy is also slightly better than that of three layers of stacked soft-threshold encoders.

Second, we found that image size has a huge impact on classification accuracy. Using the original image size, single-layer NOMP achieves an accuracy (64.6%) higher than all of the previously reported numbers. With 3 layers, NOMP achieves 67.5% accuracy, a new state-of-the-art accuracy. This result highlights the importance of efficient and scalable training and robust encoding algorithms.

7. Conclusion

In this work, we have studied greedy sparse encoders for use in unsupervised sparse representation learning. We have found that the stability of OMP, known to be relatively weak, is the cause of its suboptimal classification accuracy. We have shown that this issue can be largely alleviated by simply adding a nonnegativity constraint. The proposed NOMP encoder is not only very efficient, but also delivers competitive accuracy to other best known encoders, including deep neural networks, when the amount of labeled training samples is limited. This makes NOMP very attractive to building large-scale image classification systems.

Acknowledgments

This material is based on research sponsored in part by the Intel Corporation. Tsung-Han Lin is supported by the Siebel Scholars Program.

References

Bengio, Y., Courville, A., and Vincent, P. Representation learning: A review and new perspectives. *IEEE Trans.*

Pattern Anal. Machine Intell., 35(8):1798–1828, 2013.

Berkes, P. and Wiskott, L. Slow feature analysis yields a rich repertoire of complex cell properties. *Journal of Vision*, 5(6):579–602, 2005.

Bo, L., Ren, X., and Fox, D. Unsupervised feature learning for RGB-D based object recognition. In *ISER*, 2012.

Bruckstein, A., Elad, M., and Zibulevsky, M. On the uniqueness of nonnegative sparse solutions to underdetermined systems of equations. *IEEE Trans. Inform. Theory*, 54(11):4813–4820, 2008.

Coates, A. and Ng, A. The importance of encoding versus training with sparse coding and vector quantization. In *ICML*, 2011a.

Coates, A. and Ng, A. Selecting receptive fields in deep networks. In *NIPS*, 2011b.

Coates, A., Ng, A., and Lee, H. An analysis of single-layer networks in unsupervised feature learning. In *AISTATS*, 2011.

Donoho, D. and Stodden, V. When does non-negative matrix factorization give a correct decomposition into parts? In *NIPS*, 2003.

Donoho, D., Elad, M., and Temlyakov, V. Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Trans. Inform. Theory*, 52(1):6–18, 2006.

Goodfellow, I, Courville, A, and Bengio, Y. Large-scale feature learning with spike-and-slab sparse coding. In *ICML*, 2012.

Goodfellow, I., Warde-Farley, D., and Mirza, M. Maxout Networks. In *ICML*, 2013.

Gregor, K. and LeCun, Y. Learning fast approximations of sparse coding. In *ICML*, 2010.

Hoyer, P. Modeling receptive fields with non-negative sparse coding. *Neurocomputing*, 52:547–552, 2003.

Hoyer, P. Non-negative matrix factorization with sparseness constraints. *The Journal of Machine Learning Research*, 5:1457–1469, 2004.

Hui, K.-Y. Direct Modeling of Complex Invariances for Visual Object Features. In *ICML*, 2013.

Kavukcuoglu, K., Ranzato, M., and LeCun, Y. Fast inference in sparse coding algorithms with applications to object recognition. *CBLL-TR-2008-12-01, NYU*, 2008.

Kavukcuoglu, K., Sermanet, P., Boureau, Y-L., Gregor, K., Mathieu, M., and LeCun, Y. Learning convolutional feature hierarchies for visual recognition. In *NIPS*, 2010.

Krizhevsky, A. and Hinton, G. E. Learning multiple layers of features from tiny images. *Technical report, University of Toronto*, 2009.

Lee, D. D. and Seung, H. S. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755): 788–791, 1999.

Lee, H., Battle, A., Raina, R., and Ng, A. Efficient sparse coding algorithms. In *NIPS*, 2006.

Lee, H., Grosse, R., Ranganath, R., and Ng, A. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *ICML*, 2009.

Nair, V. and Hinton, G. E. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.

Ngiam, J., Koh, P.-W., Chen, Z., Bhaskar, S., and Ng, A. Sparse filtering. In *NIPS*, 2011.

Paatero, P. and Tapper, U. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126, 1994.

Pati, Y. C., Rezaiifar, R., and Krishnaprasad, P. S. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Asilomar Conference on Signals, Systems and Computers*. IEEE, 1993.

Raina, R., Battle, A., Lee, H., Packer, B., and Ng, A. Self-taught learning: transfer learning from unlabeled data. In *ICML*, 2007.

Ranzato, M., Huang, F., Boureau, Y-L., and Lecun, Y. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *CVPR*. IEEE, 2007.

Rozell, C., Johnson, D., Baraniuk, R., and Olshausen, B. A. Sparse coding via thresholding and local competition in neural circuits. *Neural computation*, 20(10):2526–2563, 2008.

Rubinstein, R., Zibulevsky, M., and Elad, M. Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit. *CS Technion*, 2008.

Sindhwani, V. and Ghoting, A. Large-scale distributed non-negative sparse coding and sparse dictionary learning. In *KDD*, 2012.

Slawski, M. and Hein, M. Sparse recovery by thresholded non-negative least squares. In *NIPS*, 2011.

Zeiler, M. and Fergus, R. Stochastic pooling for regularization of deep convolutional neural networks. *ICLR*, 2013.