LOCALLY COMPETITIVE ALGORITHMS FOR SPARSE APPROXIMATION

Christopher Rozell, Don Johnson, Richard Baraniuk

Electrical & Computer Engineering Department
Rice University

Bruno Olshausen

Helen Wills Neuroscience Institute
University of California, Berkeley

ABSTRACT

Practical sparse approximation algorithms (particularly greedy algorithms) suffer two significant drawbacks: they are difficult to implement in hardware, and they are inefficient for time-varying stimuli (e.g., video) because they produce erratic temporal coefficient sequences. We present a class of *locally competitive algorithms* (LCAs) that correspond to a collection of sparse approximation principles minimizing a weighted combination of reconstruction MSE and a coefficient cost function. These systems use thresholding functions to induce local nonlinear competitions in a dynamical system. Simple analog hardware can implement the required nonlinearities and competitions. We show that our LCAs are stable under normal operating conditions and can produce sparsity levels comparable to existing methods. Additionally, these LCAs can produce coefficients for video sequences that are more regular (i.e., smoother and more predictable) than the coefficients produced by greedy algorithms.

Index Terms— Approximation methods, visual system, image coding, video coding, nonlinear systems

1. INTRODUCTION

Sparsity has become an important concept in many signal and image processing paradigms. The connection between sparsity and applications such as denoising [1] and compression [2] is well-established. Recent advances in reconstruction from highly undersampled measurements (often referred to as *compressed sensing*) [3] have again brought attention to sparse approximation as a valuable signal processing tool. Furthermore, recent theoretical and experimental evidence indicates that many sensory neural systems appear to employ similar sparse representations, encoding a stimulus with the activity of a small subset of neurons in a population [4].

Optimal sparse approximation is computationally intractable, leading practitioners to often employ two alternate strategies: *convex relaxation* and *greedy algorithms*. Convex relaxation substitutes a convex penalty on the coefficients in place of simply counting the non-zero coefficients [5]. Greedy algorithms iteratively select the single best dictionary element to represent the current residual signal. While not optimal in any sense, greedy algorithms often perform well in practice [6]. However, existing algorithms have two significant drawbacks: they are difficult to implement directly in hardware, and they do not efficiently handle time-varying signals (e.g, video). In particular, these algorithms often produce erratic temporal coefficients even when presented with a smoothly varying input.

Motivated by neurally plausible sparse coding mechanisms, we introduce and study a new class of sparse approximation algorithms based on the principles of *thresholding* and *local competition* that addresses many of drawbacks observed in existing methods. In our Locally Competitive Algorithms (LCAs), each dictionary element is assigned to a node that may continually compete with neighboring nodes. Node dynamics are described by a set of non-linear ordinary differential equations (ODEs) that correspond to simple analog hardware components. Unlike greedy algorithms that irrevocably select a single dictionary element at each iteration, LCAs allow many coefficients to simultaneously enter or leave the representation.

This paper develops an architecture for LCAs and shows their correspondence to a broad class of sparse approximation problems that minimize a combination of reconstruction mean-squared error (MSE) and a sparsity-inducing cost function. We show that a specific LCA displays several critical properties: it is stable, it produces image coefficients with comparable sparsity to greedy algorithms, and it produces time-varying video coefficients that are significantly more regular than greedy algorithms.

2. BACKGROUND AND RELATED WORK

2.1. Sparse approximation

Given an N-pixel image $\boldsymbol{s} \in \mathbb{R}^N$, we seek a representation in terms of a dictionary $\mathcal{D}$ composed of M (unit-norm) vectors $\{\boldsymbol{\phi}_m\}$ that span the space $\mathbb{R}^N$. When the dictionary is overcomplete ($M > N$), there are an infinite number of ways to choose coefficients $\{a_m\}$ such that $\boldsymbol{s} = \sum_{m=1}^{M} a_m \boldsymbol{\phi}_m$. Optimal sparse approximation seeks the fewest number of non-zero coefficients (known as the ℓ^0 quasi-norm of the coefficients) representing $\boldsymbol{s}$ to a specified fidelity, but is an NP-hard optimization [7].

Two suboptimal approaches to solving the optimal sparse approximation problem are typically employed. The first approach of *convex relaxation* is typified by Basis Pursuit De-Noising (BPDN) [5]. BPDN finds the coefficients having minimum ℓ^1 norm and can be solved using modern interior point-type methods. The second approach of using *greedy algorithms* is typified by Matching Pursuit (MP) [8]. MP is initialized with a residual $r_0 = \boldsymbol{s}$. At the k^{th} iteration, MP finds the index of the dictionary element best approximating the current residual signal, $\theta_k = \arg\max_m |\langle r_{k-1}, \boldsymbol{\phi}_m\rangle|$. The resulting coefficient is recorded and the residual is updated, $r_k = r_{k-1} - \boldsymbol{\phi}_{\theta_k} d_k$. Though globally suboptimal, greedy algorithms often efficiently find good sparse signal representations [6].

2.2. Related work

While there are many other sparse approximation methods, the most interesting for our purposes are recent papers developing algorithms that also combine hard thresholding and simple linear op-

This work was funded by grants NGA MCA 015894-UCB, NSF IIS-06-25223 and CCF-0431150, DARPA/ONR N66001-06-1-2011 and N00014-06-1-0610, ONR N00014-06-1-0769 and N00014-06-1-0829, AFOSR FA9550-04-1-0148, and the Texas Instruments Leadership University Program. Correspondence to: {`crozell,dhj,richb`}`@rice.edu` and `baolshausen@berkeley.edu`.

erations [9–11]. While these models have many differences in their details (especially the homogeneity of the threshold in time and over the node index), our LCAs have two primary deviations. First, the LCAs have a "charging up" behavior due to the leaky integrator that allows continual competition and helps to smooth the time-varying coefficients. Second, each LCA can be exactly related to a sparsity cost function that is being (locally) minimized regardless of the dictionary structure (including tight and non-tight frames). It is not clear how the performance of these methods directly compares.

3. LOCALLY COMPETITIVE ALGORITHMS (LCAS)

We begin by drawing from knowledge of sensory neural system architectures to develop a system that can be easily implemented in analog hardware. Our LCAs use a parallel set of nodes where each node is associated with an element of the dictionary $\boldsymbol{\phi}_m \in \mathcal{D}$. The system is presented with an input image $\boldsymbol{s}(t)$, and node state variables $u_m(t)$ begin "charging up" like a leaky integrator. When state variables reach an activation threshold λ, the node also produces a significant non-zero output coefficient a_m that inhibits the driving input of neighboring nodes. Each coefficient is related to the internal state through an activation function $a_m = T_\lambda(u_m)$, that is essentially zero for values below λ and linear for values above λ.

Our LCA node dynamics are expressed by the non-linear ODEs

$$\dot{u}_m(t) = f(u_m(t)) = \frac{1}{\tau}\left[b_m(t) - u_m(t) - \sum_{n \neq m} G_{m,n} a_n(t)\right], \tag{1}$$

where the node's input is $b_m(t) = \langle \boldsymbol{\phi}_m, \boldsymbol{s}(t)\rangle$. The nodes best matching the stimulus will have internal state variables that charge at the fastest rates. If node m crosses the threshold and becomes active, it inhibits the driving input for the n node by an amount proportional to their similarity $G_{m,n} = \langle \boldsymbol{\phi}_m, \boldsymbol{\phi}_n\rangle$. The possibility of unidirectional inhibition gives strong nodes a chance to prevent weaker nodes from becoming active and initiating inhibition.

For a fixed input (i.e., an image), the steady state set of active coefficients $\{a_m(t)\}$ represent the input. If the changes in a video sequence (i.e., the frame rate) are slower than the system time constant, the coefficients will also reach steady-state for each change. The goal is to define the LCA system dynamics (including the activation function) so that few coefficients non-zero values while approximately reconstructing the input, $\widehat{\boldsymbol{s}}(t) = \sum_m a_m(t)\,\boldsymbol{\phi}_m$.

In addition to being implementable in hardware, the LCA architecture solves a family of sparse approximation problems. We have shown [12] that LCAs descend an energy function combining the reconstruction MSE and a sparsity-inducing cost penalty $C(\cdot)$,

$$E(t) = \frac{1}{2}\left|\left|\boldsymbol{s}(t) - \widehat{\boldsymbol{s}}(t)\right|\right|^2 + \lambda \sum_m C(a_m(t))\,.$$

The specific form of the cost function is determined by the form of the activation function $T_\lambda(\cdot)$ according to the relationship

$$\lambda \frac{dC(a_m)}{da_m} = u_m - a_m = u_m - T_\lambda(u_m). \tag{2}$$

This correspondence can be seen by computing the derivative of E with respect to the active coefficients and then allowing the internal state dynamics to descend this gradient, $\dot{u}_m \propto -\frac{dE}{d\,a_m}$.

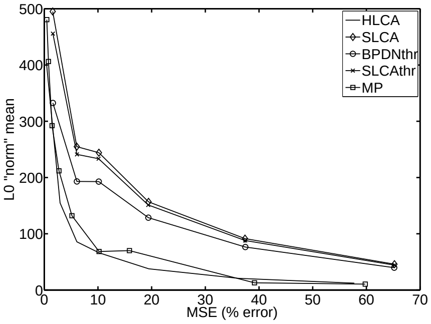


Fig. 1. Mean tradeoff between MSE and ℓ^0-sparsity for normalized (32×32) patches from a standard set of test images.

We focus specifically on the cost functions associated with *thresholding* activation functions that set small values identically to zero. Thresholding limits the competition to only the "strong" nodes. We choose a smooth sigmoidal thresholding function

$$T_\lambda(u_m) = \frac{u_m - \alpha\lambda}{1 + e^{-\gamma(u_m - \lambda)}}, \tag{3}$$

where γ is a parameter controlling the speed of the threshold transition and $\alpha \in [0, 1]$ indicates what fraction of an additive adjustment is made for values above threshold. We are particularly interested in the limit as $\gamma \to \infty$, making this thresholding function become discontinuous. We focus primarily on the special case $\alpha = 0$, known as a "hard" thresholding function. Using (2), this hard-thresholding locally competitive algorithm (HLCA) applies an ℓ^0-like cost function by using a constant penalty regardless of coefficient magnitude,

$$C(a_m) = \frac{\lambda}{2}\, I(|a_m| > \lambda)\,,$$

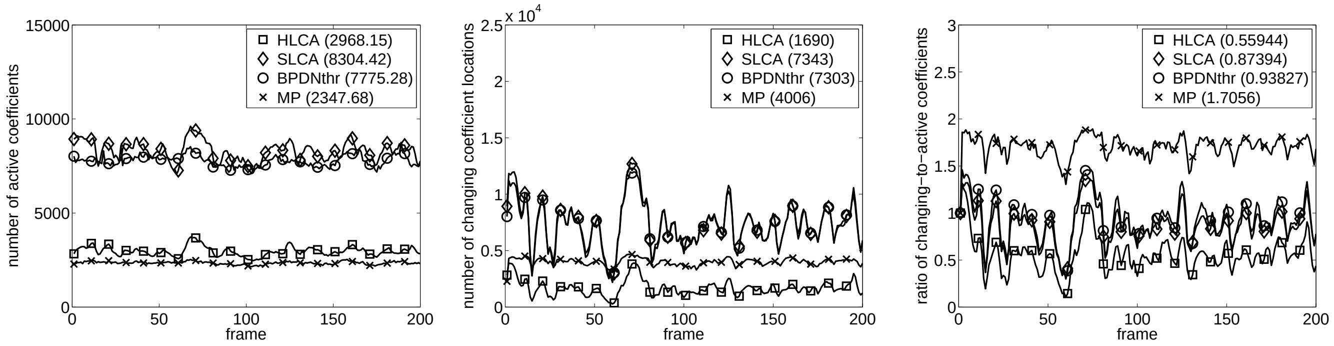
where $I(\cdot)$ is the indicator function evaluating to 1 if the argument is true and 0 if the argument is false. While the ℓ^0-like energy function is very appealing, this energy function is not convex. This normally poses an unwanted danger of only finding a local minima, but we will show that this property works to our advantage by inducing inertia in time-varying coefficients. It is worth noting that the soft thresholding locally competitive algorithm (SLCA) with $\alpha = 1$ corresponds to an ℓ^1 coefficient penalty (i.e., the BPDN objective function).

4. HLCA SYSTEM PROPERTIES

We require that LCAs exhibit three critical properties: stability under normal operating conditions, sparse coefficients for fixed images, and smooth (or "regular") coefficient sequences for video inputs. While we focus on the HLCA, our analysis generally applies to all LCAs through straightforward (perhaps messy) extensions.

4.1. Stability

Non-linear systems are often characterized in terms of their input-output relationship and their behavior near an equilibrium point $\boldsymbol{u}^*$, $f(\boldsymbol{u}^*) = 0$. To describe HLCA system stability, we first define $\mathcal{M}_{\boldsymbol{u}(t)} \subseteq [1, \ldots, M]$ as the set of nodes above threshold, $\mathcal{M}_{\boldsymbol{u}(t)} = \{m : |u_m(t)| \geq \lambda\}$. We say that the HLCA meets the *stability criteria* if for all time t the set of active vectors $\{\boldsymbol{\phi}_m\}_{m \in \mathcal{M}_{\boldsymbol{u}(t)}}$ is linearly independent. This condition ensures that node competitions do not balance to have zero net effect, and we will draw on it to show several different properties. Under normal operating conditions (i.e., typical thresholds, time constants, and dictionaries), the HLCA should satisfy the stability criteria.



(a) (b) (c)

Fig. 2. Encodings of the “Foreman” test video sequence. Average values are noted in the legend. *(a)* The number of active coefficients in each frame. *(b)* The number of changing coefficient locations for each frame. *(c)* The ratio of changing coefficients to active coefficients.

We first consider local behavior in the ball around an equilibrium point, $B_\epsilon(\boldsymbol{u}^*) = \{\boldsymbol{u} : ||\boldsymbol{u} - \boldsymbol{u}^*|| < \epsilon\}$. A system is *locally asymptotically stable* [13] at an equilibrium point $\boldsymbol{u}^*$ if you can specify $\epsilon > 0$ such that $\boldsymbol{u}(0) \in B_\epsilon(\boldsymbol{u}^*) \implies \lim_{t\to\infty} \boldsymbol{u}(t) = \boldsymbol{u}^*$. This local stability notion also implies that the system will remain well-behaved under perturbations (Theorems 2.8 and 2.9 in [13]). We have shown [12] that if the stability criteria is met, then the HLCA:

- has a finite number of equilibrium points;
- has equilibrium points that are almost certainly isolated (no two equilibrium points are arbitrarily close together); and
- is almost certainly locally asymptotically stable for every equilibrium point.

With a finite number of isolated equilibria, we can be confident that the HLCA steady-state response is a distinct set of coefficients representing the stimulus. Regarding input-output behavior, we have also shown that the HLCA output coefficients have bounded energy for bounded energy inputs if the stability criteria are met and if the system nodes don’t cross threshold “too often” [12]. We expect that infinitely fast switching can be avoided either by the physical principles of the implementation or through an explicit hysteresis.

4.2. Sparsity and representation error

To understand the HLCA reconstruction fidelity, we note from rewriting equation (1) that (for a constant input) the HLCA equilibrium points ($\dot{\boldsymbol{u}}(t) = 0$) occur when the residual error is orthogonal to active nodes and balanced with the inactive nodes,

$$\langle \boldsymbol{\phi}_m, \boldsymbol{s}(t) - \widehat{\boldsymbol{s}}(t) \rangle = \begin{cases} u_m(t) & \text{if } |u_m| \le \lambda \\ 0 & \text{if } |u_m| > \lambda \end{cases}.$$

Therefore, the HLCA will perfectly reconstruct the component of the input that projects onto the subspace spanned by the active nodes.

Though the HLCA may not find the globally optimal solution, we must ensure that it is being reasonably efficient. We cannot determine the LCA steady-state coefficients for arbitrary starting points, but it is possible to rule out some sets as *not* being possible. For example, let $\mathcal{M} \subseteq [1, \ldots, M]$ be an arbitrary set of active coefficients. We have shown [12] that when the stability criteria are met, the following statement is true for the HLCA: *If $\boldsymbol{s} = \boldsymbol{\phi}_m$, any set of active coefficients $\mathcal{M}$ with $m \in \mathcal{M}$ and $|\mathcal{M}| > 1$ cannot be a steady-state response.* In other words, the HLCA may use the m^{th} node or a collection of other nodes to represent $\boldsymbol{s}$, but it cannot use a combination of both. This result extends intuitively beyond one-sparse signals: each component in an optimal decomposition is represented by either the optimal node or another collection of nodes, but not both. While not necessarily finding the optimal representation, the HLCA does not needlessly employ both the optimal and extraneous nodes.

We have verified numerically that the HLCA achieves a sparsity comparable with greedy algorithms. We simulated the HLCA on normalized (32×32) bandpass patches from a standard set of test images using $\tau = 10^{-3}$ and a dictionary of four orientation bands in the bandpass level of a steerable pyramid [14] (i.e., the dictionary is approximately four times overcomplete). Figure 1 shows the tradeoff between ℓ^0 sparsity and MSE for HLCA, MP, a standard BPDN solver followed by thresholding to enforce ℓ^0 sparsity (denoted “BPDNthr”) and SLCA with the same threshold applied (denoted “SLCAthr”). Most importantly, note that the HLCA and MP are almost identical in their sparsity-MSE tradeoff. Though there are connections between HLCA and MP (the competition signal for a fully charged node is the same as the MP update step), the resulting coefficients can be very different. In fact, HLCA can produce optimal coefficients in pathological cases where MP runs forever [12].

4.3. Time-varying inputs

The temporal irregularity observed when applying sparse approximation methods to successive video frames introduces additional uncertainty that hinders both encoding and computer vision tasks. While some methods for correcting this problem have been employed (e.g., motion prediction and spatio-temporal dictionaries), these tactics would be difficult to implement in hardware. In contrast, the HLCA exhibits inertia that smooths the coefficient time series. The HLCA seeks a local energy function minima, with sparse coefficients that are “near” the coefficients from the previous frame. To illustrate the increased regularity, we simulated the HLCA (and SLCA) on (144×144) bandpass, normalized frames from the standard “Foreman” test video sequence (using the setup described in Section 4.2). The LCA input is switched to the next video frame every (simulated) 1/30 seconds. Figure 2 shows comparisons to MP and BPDN applied to each frame. Changing coefficient locations are nodes that either became active or inactive at each frame.

This simulation shows that the HLCA uses approximately the same number of active coefficients as MP but is much more efficient

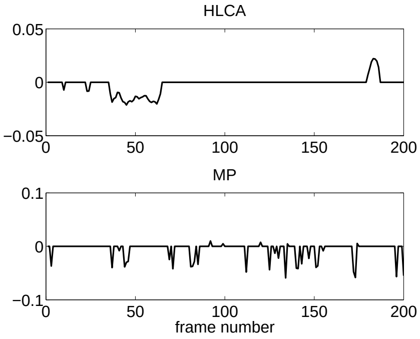


Fig. 3. Example time-series coefficients for the test video sequence.

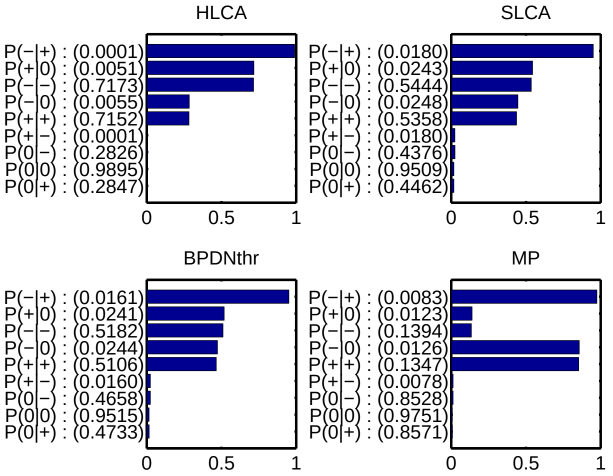


Fig. 4. Transition probabilities for coefficient states.

in how it chooses the coefficient locations. This difference is quantified by the ratio of the number of changing coefficients to the number of active coefficients. MP has a ratio of 1.7, meaning that MP is finding almost an entirely new set of active coefficient locations at each frame. In contrast, the HLCA ratio of 0.5 means that it is changing approximately 25% of its coefficient locations. This difference can be seen in an example coefficient time-series (Figure 3).

We can quantify the increased predictability of the active coefficient locations by calculating their conditional entropy. At frame n, each coefficient can be classified as being in one of three possible states: negative, zero and positive, $\sigma_m(n) \in (-, 0, +)$. Viewing each coefficient time-series as a Markov chain we can calculate the conditional probabilities $P\left(\sigma_m(n)\,|\sigma_m(n-1)\right)$ of moving to a state given the previous state (shown in Figure 4). While the HLCA and MP are equally likely to have non-zero states, the HLCA is over five times more likely than MP to have non-zero coefficients retain their state. The conditional entropy indicates how much uncertainty there is about the state of the current coefficients given the coefficient states from the previous frame. The HLCA and MP conditional entropies are 0.7 and 0.14 bits, respectively, further confirming that HLCA coefficients are much more predictable than MP coefficients.

5. CONCLUSIONS AND FUTURE WORK

We have proposed a class of locally competitive algorithms that solve a series of sparse approximation problems and address some of the drawbacks of common sparse approximation algorithms (especially greedy methods). In addition to being implementable using a parallel network of simple hardware elements, the HLCA exhibits stability, achieves sparsity levels comparable to MP, and produces video coefficient sequences that are substantially smoother than MP.

As data collection rates increase, we will require faster and more efficient processing strategies. LCAs offer an opportunity to rethink the traditional (and somewhat inefficient) paradigm of sampling followed by compression on a single CPU. Instead, LCAs offer a fast and energy efficient method for compressing signals in a parallel and analog computational platform *before* digitization. We anticipate that the sparse and smooth LCA representations could produce efficient video coders as well as improving many computer vision applications relating to scene understanding.

6. REFERENCES

[1] D.L. Donoho, “Denoising by soft-thresholding,” *IEEE Trans. Info. Th.*, vol. 41, no. 3, pp. 613–627, May 1995.

[2] R.A. DeVore, B. Jawerth, and B.J. Lucier, “Image compression through wavelet transform coding,” *IEEE Trans. Info. Th.*, vol. 38, no. 2, pp. 719?–746, March 1992.

[3] E. Candès, J. Romberg, and T. Tao, “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information,” *IEEE Trans. Info. Th.*, vol. 52, no. 2, pp. 489–509, February 2006.

[4] B. Olshausen and D. Field, “Sparse coding of sensory inputs,” *Cur. Op. Neur.*, vol. 14, pp. 481–487, 2004.

[5] S.S. Chen, D.L. Donoho, and M.A. Saunders, “Atomic decomposition by basis pursuit,” *J. Sci. Comp.*, vol. 43, no. 1, pp. 129–159, 2001.

[6] J. Tropp, “Greed is good: Algorithmic results for sparse approximation,” *IEEE Trans. Info. Th.*, vol. 50, no. 10, pp. 2231–2242, 2004.

[7] B.K. Natarajan, “Sparse approximate solutions to linear systems,” *SIAM J. Comp.*, vol. 24, no. 2, pp. 227–234, April 1995.

[8] S. Mallat and Z. Zhang, “Matching pursuits with time-frequency dictionaries,” *IEEE Trans. Sig. Proc.*, vol. 41, no. 12, pp. 3397–3415, December 1993.

[9] N. Kingsbury and T. Reeves, “Redundant representation with complex wavelets: How to achieve sparsity,” in *Proc. Intl. Conf. on Image Proc.*, 2003.

[10] L. Mancera and J. J. Portilla, “L0-norm-based sparse representation through alternate projections,” in *Proc. Intl. Conf. Image Proc.*, Atlanta, GA, 2006, pp. 1749–1752.

[11] M. Rehn and T. Sommer, “A network that uses few active neurones to code visual input predicts the diverse shapse of cortical receptive fields,” *J. Comp. Neuro.*, 2006.

[12] C.J. Rozell, D.H Johnson, R.G. Baraniuk, and B.A. Olshausen, “Neurally plausible sparse coding via thresholding and local competition,” *Neur. Comp.*, January 2007, Submitted.

[13] A. Bacciotti and L. Rosier, *Liapunov functions and stability in control theory*, Springer, New York, 2001.

[14] E.P. Simoncelli and W.T. Freeman, “The steerable pyramid: A flexible architecture for multi-scale derivative computation,” in *Proc. Intl. Conf. Image Proc.*, 1995.