Neural Computation, submitted March 2007

Neurally plausible sparse coding via thresholding and local competition

Christopher J. Rozell, Don H. Johnson,
Richard G. Baraniuk, Bruno A. Olshausen

Abstract

While evidence indicates that neural systems may be employing sparse approximations to represent sensed stimuli, the mechanisms underlying this ability are not understood. We present a class of neurally plausible *locally competitive algorithms* (LCAs) that correspond to a collection of sparse coding principles minimizing a weighted combination of mean-squared error (MSE) and a coefficient cost function. LCAs use thresholding functions to induce local (usually one-way) inhibitory competitions. In contrast to greedy algorithms that iteratively select the single best element, LCAs allow continual interaction among many units. LCAs produce coefficients with sparsity levels comparable to existing sparse coding algorithms while being plausible for neural implementation. Additionally, LCAs coefficients for video sequences demonstrate inertial properties that are both qualitatively and quantitatively more regular (i.e., smoother and more predictable) than the coefficients produced by greedy algorithms.

1 Introduction

Natural images can be well-approximated by a small subset of elements from an overcomplete dictionary [1]. The process of choosing a good subset of dictionary elements along with the corresponding coefficients to represent a signal is known as *sparse approximation*. Recent theoretical and experimental evidence indicates that many sensory neural systems appear to employ similar sparse representations with their population codes [2–6], encoding a stimulus in the activity of just a few neurons. While sparse coding in neural populations is an intriguing hypothesis, neurally plausible mechanisms capable of efficiently finding sparse approximations are currently unknown. The challenge of collecting simultaneous data from large neural populations makes it difficult to evaluate this hypothesis without testing predictions from a specific proposed mechanism.

Sparse approximation is a difficult problem that is the center of much research in mathematics and signal processing. Researchers have developed a variety of sparse approximation algorithms, including convex relaxation and very efficient *greedy algorithms* that often work well in practice [7]. Greedy algorithms are iterative schemes where each iteration selects the single best dictionary element to represent the residual signal without regard to which other elements will be selected. However, existing algorithms have two significant drawbacks that make them unlikely proposals for neural populations: they would be difficult to implement in the parallel computational primitives used by neural systems, and they do not efficiently handle the time-varying signals critical for biological system behavior.

We introduce and study a new class of neurally plausible sparse approximation algorithms based on the principles of *thresholding* and *local competition* that addresses many of the drawbacks observed in existing methods. In our Locally Competitive Algorithms (LCAs), neurons in a population continually compete with neighboring units using (usually one-way) lateral inhibition to calculate coefficients representing an input using an overcomplete dictionary. Unlike greedy algorithms that irrevocably select the single best dictionary element at each iteration, parallel competition allows many coefficients to become part of the representation simultaneously, perhaps even working together to suppress a coefficient that had previously been selected. Our continuous-time LCA is described by the dynamics of a system of nonlinear ordinary differential equations (ODEs) that govern the internal state (membrane potential) and external communication (short-term

firing rate) of units in a neural population. We show that each LCA corresponds to an optimal sparse approximation problem that minimizes an energy function combining reconstruction mean-squared error (MSE) and a sparsity-inducing cost function.

A sparse coding system built on a realistic neural architecture must possess three critical properties. Most importantly, the system must be stable in some meaningful sense to guarantee that the physical system is well-behaved under normal operating conditions. Next, the system must perform its primary task well, finding a good tradeoff between sparsity and representation error when coding a fixed image. Finally, the system must handle time-varying inputs efficiently. In particular, a neurally plausible system must produce coefficients that reflect the smooth character of natural input signals. In addition to being built on a neurally realistic architecture, we illustrate that LCAs also exhibit these three properties.

Specifically, LCAs can produce representations for static signals with approximately the same sparsity as other known techniques, including greedy algorithms and convex relaxation. Additionally, the LCA coefficients representing time-varying signals are much more regular than the coefficients produced by greedy algorithms. In particular, while greedy algorithms can be very erratic in their coefficient selection at each time step, LCA coefficients display *inertia*. This inertia regularizes the time variation of the sparse coefficients for time-varying inputs by encouraging as many coefficients as possible to retain their state when the input changes. This regularity makes the coefficients much more predictable, making it easier for higher-level structures to identify and understand the changing content in the time-varying stimulus.

This paper develops a neural architecture for LCAs, shows their correspondence to a broad class of sparse approximation problems, and details their advantages over existing sparse coding algorithms (particularly greedy algorithms). While the principles we describe apply to many modalities, we will focus on the visual system and the representation of video sequences. The LCA methods presented here point toward information representation methods possibly used by sensory neural systems employing sparse coding and represent a first step toward a testable neurobiological model for these systems.

2 Background and related work

2.1 Sparse approximation

Given an N-dimensional stimulus $\boldsymbol{s} \in \mathbb{R}^N$ (e.g., an N-pixel image), we seek a representation in terms of a dictionary $\mathcal{D}$ composed of M vectors $\{\boldsymbol{\phi}_m\}$ that span the space $\mathbb{R}^N$. Define the ℓ^p norm of the vector $\boldsymbol{x}$ to be $||\boldsymbol{x}||_p = \left(\sum_m |x_m|^p\right)^{1/p}$ and the inner product between $\boldsymbol{x}$ and $\boldsymbol{y}$ to be $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \sum_m x_m y_m$. Without loss of generality, assume the dictionary vectors are unit-norm, $||\boldsymbol{\phi}_m||_2 = 1$. When the dictionary is overcomplete ($M > N$), there are an infinite number of ways to choose coefficients $\{a_m\}$ such that $\boldsymbol{s} = \sum_{m=1}^{M} a_m \boldsymbol{\phi}_m$. In optimal sparse approximation, we seek the coefficients having the fewest number of non-zero entries by solving the minimization problem

$$\min_{\boldsymbol{a}} ||\boldsymbol{a}||_0 \quad \text{subject to} \quad \boldsymbol{s} = \sum_{m=1}^{M} a_m \boldsymbol{\phi}_m, \tag{1}$$

where the ℓ^0 "norm"[1] denotes the number of non-zero elements of $\boldsymbol{a} = [a_1, a_2, \ldots, a_M]$. Unfortunately, this combinatorial optimization problem is NP-hard [8].

In the signal processing community, two primary approaches are typically used to efficiently find acceptable suboptimal solutions to the optimal sparse approximation problem. The first general approach substitutes alternate sparsity measures to convexify the ℓ^0 norm. One well-known example is Basis Pursuit (BP) [9], which replaces the ℓ^0 norm with the ℓ^1 norm

$$\min_{\boldsymbol{a}} ||\boldsymbol{a}||_1 \quad \text{subject to} \quad \boldsymbol{s} = \sum_{m=1}^{M} a_m \boldsymbol{\phi}_m. \tag{2}$$

[1] While clearly not a norm in the mathematical sense, we will use this terminology prevalent in the literature.

BP is especially important for signal processing applications because (2) is a convex optimization problem equivalent to a linear program, solvable using modern interior point methods. If the signal is sparse compared to the nearest pair of dictionary elements (e.g., $||\boldsymbol{a}||_0 < \min_{m \neq n} \frac{1}{2}\left[1 + 1/\langle \boldsymbol{\phi}_m, \boldsymbol{\phi}_n \rangle\right]$) BP has the same solution as the optimal sparse approximation problem [10].

The perfect reconstruction imposed by (2) may be too strict in many cases, especially in the presence of noise. Instead, we may make a tradeoff between reconstruction mean-squared error (MSE) and sparsity by forming an objective function that is a weighted combination of MSE and the ℓ^1 coefficient norm. For example, Basis Pursuit De-Noising (BPDN) [9], corresponds to the unconstrained optimization problem:

$$\min_{\boldsymbol{a}} \left(\left\| \boldsymbol{s} - \sum_{m=1}^{M} a_m \boldsymbol{\phi}_m \right\|_2^2 + \lambda \left|\left| \boldsymbol{a} \right|\right|_1 \right), \tag{3}$$

where λ is a tradeoff parameter. BPDN provides the ℓ^1-sparsest approximation for a given reconstruction quality. There are many algorithms that can be used to solve the BPDN optimization problem, with interior point-type methods ((3) corresponds to a "perturbed" linear program [9]) being the most common choice. Our use of the term BPDN in this paper will assume that interior point methods are used to find a numerical solution.

The second general approach employed by signal processing researchers uses iterative greedy algorithms to constructively build up a signal representation [7]. The canonical example of a greedy algorithm is known in the signal processing community as Matching Pursuit (MP) [11]. The MP algorithm is initialized with a residual $r_0 = \boldsymbol{s}$. At the k^{th} iteration, MP finds the index of the single dictionary element best approximating the current residual signal, $\theta_k = \arg\max_m |\langle r_{k-1}, \boldsymbol{\phi}_m \rangle|$. The coefficient $d_k = \langle r_{k-1}, \boldsymbol{\phi}_{\theta_k} \rangle$ and index θ_k are recorded as part of the reconstruction, and the residual is updated, $r_k = r_{k-1} - \boldsymbol{\phi}_{\theta_k} d_k$. After K iterations, the signal approximation using MP is given by $\widehat{\boldsymbol{s}} = \sum_{k=1}^{K} \boldsymbol{\phi}_{\theta_k} d_k$. Though they may not be optimal in general, greedy algorithms often efficiently find good sparse signal representations in practice.

2.2 Sparse coding in neural systems

Recent research has found compelling evidence that V1 population responses to natural stimuli may be the result of a sparse approximation. For example, it has been shown that V1 receptive fields [12,13] are consistent with optimizing the coefficient sparsity when encoding natural images [5]. Additionally, simultaneous V1 recordings show activity levels (correspond to the coefficients $\{a_m\}$) becoming more sparse as neighboring units are also stimulated using natural scenes [2]. These populations are typically very overcomplete [4], allowing great flexibility in the coefficients used to represent a stimulus. Using this flexibility to pursue sparse codes might offer many advantages to sensory neural systems, including enhancing the performance of subsequent processing stages, increasing the storage capacity in associative memories, and increasing the energy efficiency of the system [4].

A neural system using sparse coding must employ an algorithm that is computable using networks of parallel computational elements. While existing sparse approximation algorithms are effective, their implementations do not correspond to plausible neural architectures. For example, the BPDN objective function in (3) can be minimized through direct gradient descent by the network implementation mentioned in [5]. However, this implementation has several theoretical and practical shortcomings: it requires continuous inhibition between all units with overlapping receptive fields; direct gradient descent can have difficulty minimizing objective functions that decay steeply for very small coefficients; and it lacks a natural mechanism to make small coefficients identically zero. Similarly, neural circuits implementing MP [14, 15] would require tightly coupled timing of "winner-take-all" circuits [16] (because mistakes cannot be corrected) and separate memory areas to hold the current approximation and the current residual.

Beyond implementation considerations, existing sparse approximation algorithms also do not consider the time-varying stimuli faced by neural systems. A time-varying input signal $\boldsymbol{s}(t)$ would be represented with a set of time-varying coefficients $\{a_m(t)\}$. Most sparse approximation schemes have a single goal: minimize the sparsity of the representation for a fixed signal. However, higher level processing areas would

also desire coefficients sequences that change in a way matching the way the stimulus changes over time. In particular, sparse coefficients should have smooth temporal variations in response to smooth changes in the image. Simply recomputing a new sparse approximation at each time step produces coefficients that contain significant temporal variations due to idiosyncrasies of the particular sparse approximation algorithm used rather than structure contained in the image, particularly when using greedy algorithms.

In Section 3 we develop our LCAs, in which dictionary elements continually fight for the right to represent the stimulus. These LCAs adapt their coefficients continually over time as the input changes without having to build a new representation from scratch at each time step. This evolution induces inertia in the coefficients, regularizing the temporal variations for smoothly varying input signals. In contrast to the problems seen with purely greedy approaches, our LCAs are plausible for neural implementation and encourage both sparsity and smooth temporal variations in the coefficients as the stimulus changes.

2.3 Other related work

There are several sparse approximation methods that do not fit into the two primary approaches of pure greedy algorithms or convex relaxation. For example, Sparse Bayesian Learning (SBL) [17,18] maximizes the posterior distribution on the coefficients using an iterative expectation-maximization procedure [19]. SBL involves computing a matrix inverse at each iteration, making it relatively time-consuming and not amenable to neural circuit implementation. In still another direction, several researchers have parallelized greedy algorithms by selecting a set of coefficients at each iteration and correcting for the resulting correlations between coefficients [20–22]. These extensions do speed up convergence, but selected elements are committed to the representation and cannot be changed as in our LCAs. In another direction, many extensions of MP have been introduced [23, 24] that iteratively commit dictionary elements to the representation but progressively change the coefficient values through a computationally expensive orthogonalization step that also neurally implausible.

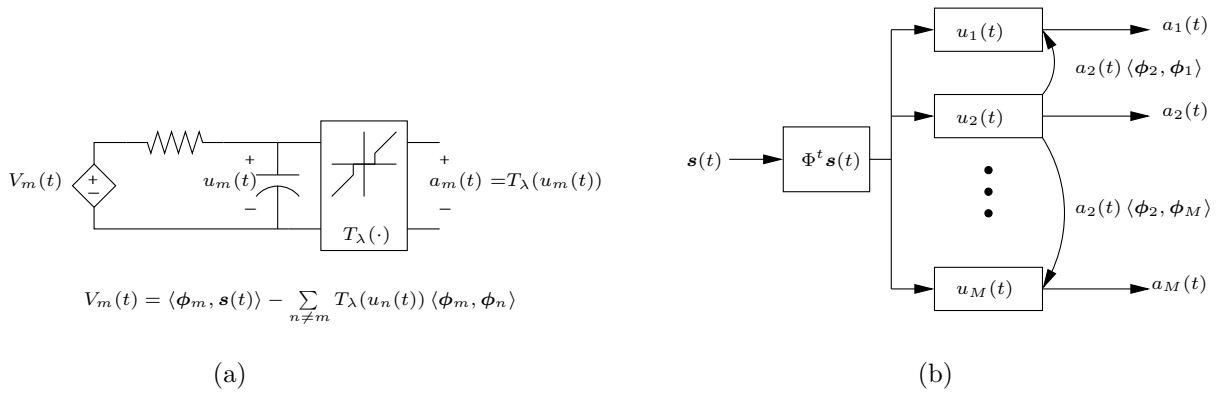
There are also several sparse approximation methods built on a parallel computational framework that are related to our LCAs [25–28]. These approaches will be discussed in more detail after describing the LCA system architecture (see Section 3.5).

3 Locally competitive algorithms for sparse coding

3.1 Architecture of locally competitive algorithms

Sensory neural systems (e.g., V1) represent a stimulus by the collective spiking activity of the neural population. Our LCAs associate each neuron with an element of the dictionary $\boldsymbol{\phi}_m \in \mathcal{D}$. The system is presented with an input image $\boldsymbol{s}(t)$, momentarily assumed to be a constant (time-varying video sequences will be considered in Section 4.3). The collection of nodes evolve according to fixed dynamics (described below) and settle on a collective output $\{a_m(t)\}$, corresponding to the short-term average firing rate of the neurons. The goal is to define the LCA system dynamics so that few coefficients have non-zero values while approximately reconstructing the input, $\widehat{\boldsymbol{s}}\,(t) = \sum_m a_m(t)\,\boldsymbol{\phi}_m \approx \boldsymbol{s}(t)$.

The LCA dynamics follow several properties observed in neural systems: inputs cause the membrane potential to "charge up" like a leaky integrator; membrane potentials over a threshold produce "action potentials" for extracellular signaling; and these super-threshold responses inhibit neighboring units through lateral connections. We represent each unit's sub-threshold value by a time-varying *internal state* $u_m(t)$ that evolves according to the dynamical system equation $\dot{u}_m(t) = \frac{1}{\tau}\left[b_m(t) - u_m(t)\right]$, where τ is the membrane time-constant. The unit's excitatory input current is proportional to how well the image matches with the node's receptive field, $b_m(t) = \langle \boldsymbol{\phi}_m, \boldsymbol{s}(t) \rangle$. With a constant input, these internal state variables converge to the projections of the image onto the dictionary elements. These values represent all of the information in the image, but are not sparse. We must introduce lateral interactions between the units to allow stronger nodes to inhibit weaker nodes, sparsifying the population response.



(a) (b)

Figure 1: (a) LCA nodes behave as a leaky integrators, charging with a speed that depends on how well the input matches the associated dictionary element and the inhibition received from other nodes. (b) A system diagram shows the inhibition signals being sent between nodes. In this case, only node 2 is shown as being active (i.e., having a coefficient above threshold) and inhibiting its neighbors. Since the neighbors are inactive then the inhibition is one-way.

When the internal state u_m of a node becomes significantly large, the node becomes "active" and produces an output signal a_m used to represent the stimulus and inhibit other nodes. This output coefficient is the result of an activation function applied to the membrane potential, $a_m = T_\lambda(u_m)$, parameterized by the system threshold λ. Though similar activation functions have traditionally taken a sigmoidal form, we consider activation functions that operate as thresholding devices — they are essentially zero for values below λ and essentially linear for values above λ. Note that for analytical simplicity we allow positive and negative coefficients, but rectified systems could use two physical units to implement one LCA node.

The nodes best matching the stimulus will have internal state variables that charge at the fastest rates. To achieve the competition allowing these nodes to win over the weaker nodes, we have active nodes inhibit other nodes with an inhibition signal proportional to both the activity level of the active node and the similarity of the nodes' receptive fields. Specifically, the inhibition signal from the active node m to any other node n is proportional to the activity level a_m and to the inner product between the node receptive fields, measured by $G_{m,n} = \langle \boldsymbol{\phi}_m, \boldsymbol{\phi}_n \rangle$. The possibility of unidirectional inhibition gives strong nodes a chance to prevent weaker nodes from becoming active and initiating counter-inhibition, thus making the search for a sparse solution more efficient.

Putting all of the above components together, our LCA node dynamics are expressed by the non-linear ordinary differential equation (ODE)

$$\dot{u}_m(t) = \frac{1}{\tau} \left[b_m(t) - u_m(t) - \sum_{n \neq m} G_{m,n} a_n(t) \right]. \tag{4}$$

This ODE is essentially the same form as the well-known continuous Hopfield network [29]. Figure 1 shows a LCA node circuit schematic and a system diagram illustrating the lateral inhibition. To express the system of coupled non-linear ODEs that govern the whole dynamic system, we represent the internal state variables in the vector $\boldsymbol{u}(t) = [u_1(t), \ldots, u_M(t)]^t$, the active coefficients in the vector $\boldsymbol{a}(t) = [a_1(t), \ldots, a_M(t)]^t = T_\lambda(\boldsymbol{u}(t))$, the dictionary elements in the columns of the $(N \times M)$ matrix $\Phi = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_M]$ and the driving inputs in the vector $\boldsymbol{b}(t) = [b_1(t), \ldots, b_M(t)]^t = \Phi^t \boldsymbol{s}(t)$. The function $T_\lambda(\cdot)$ performs element-by-element thresholding on vector inputs. The stimulus approximation is $\widehat{\boldsymbol{s}}(t) = \Phi \boldsymbol{a}(t)$,

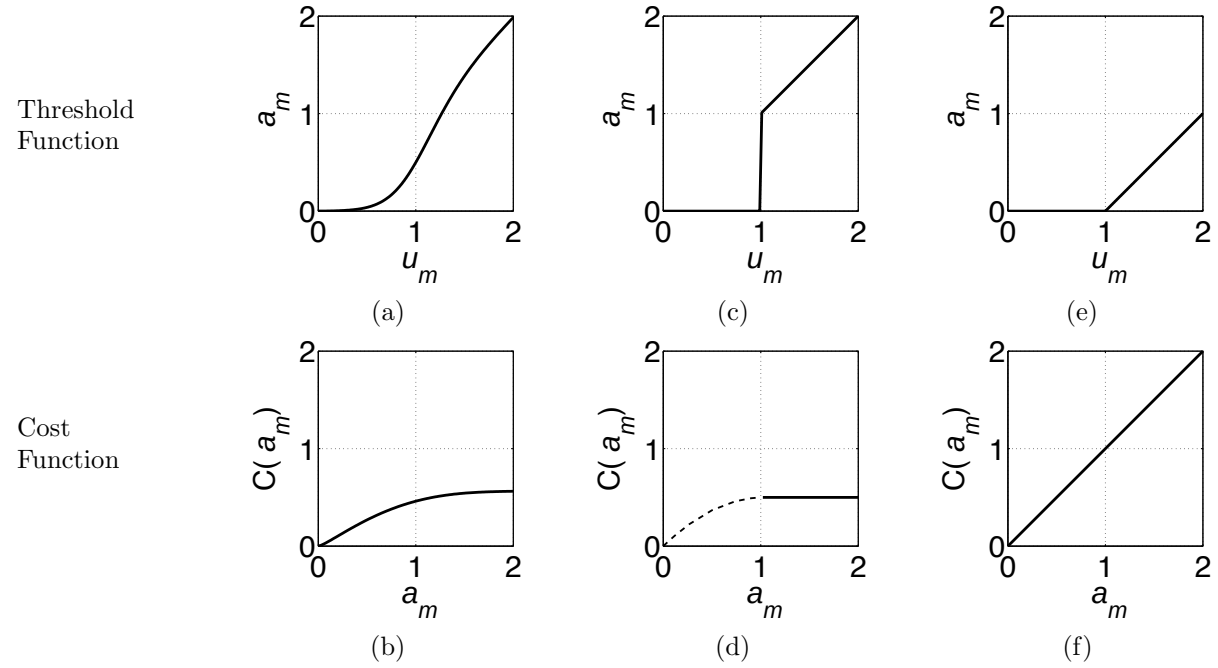


Figure 2: Relationship between the threshold function $T_{(\alpha,\gamma,\lambda)}(\cdot)$ and the sparsity cost function $C(\cdot)$. Only the positive half of the symmetric threshold and cost functions are plotted. (a) Sigmoidal threshold function and (b) cost function for $\gamma = 5$, $\alpha = 0$ and $\lambda = 1$. (c) The ideal hard thresholding function ($\gamma = \infty$, $\alpha = 0$, $\lambda = 1$) and the (d) corresponding cost function. The dashed line shows the limit, but coefficients produced by the ideal thresholding function cannot take values in this range (e) The ideal soft thresholding function ($\gamma = \infty$, $\alpha = 1$, $\lambda = 1$) and the (f) corresponding cost function.

and the full dynamic system equation is

$$\begin{aligned} \dot{\boldsymbol{u}}\left(t\right) &= f(\boldsymbol{u}(t)) = \frac{1}{\tau}\left[\boldsymbol{b}\left(t\right) - \boldsymbol{u}(t) - \left(\Phi^t\Phi - I\right)\boldsymbol{a}\left(t\right)\right], \\ \boldsymbol{a}\left(t\right) &= T_\lambda(\boldsymbol{u}(t))\,. \end{aligned} \tag{5}$$

3.2 Sparse approximation by locally competitive algorithms

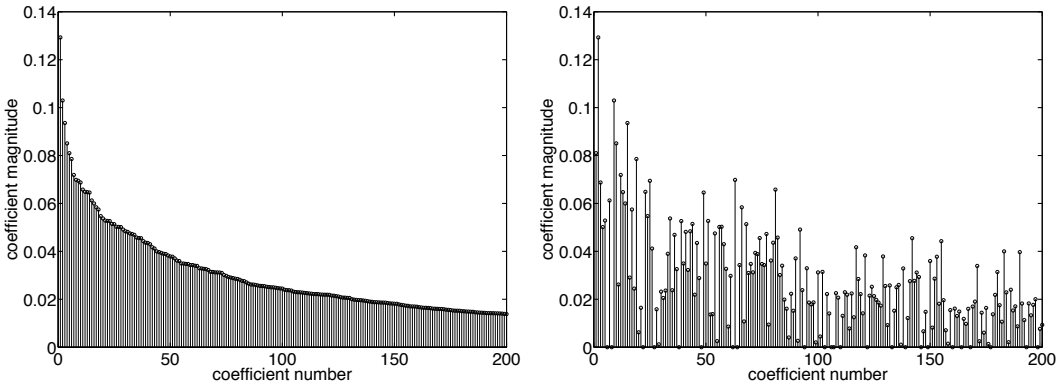
The LCA architecture described by (5) solves a family of sparse approximation problems. Specifically, LCAs descend an energy function that combines the reconstruction MSE and a sparsity-inducing cost penalty $C(\cdot)$,

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

The specific form of the cost function $C(\cdot)$ is determined by the form of the thresholding activation function $T_\lambda(\cdot)$. For a given threshold function, the cost function is specified (up to a constant) by

$$\lambda \frac{dC(a_m)}{da_m} = u_m - a_m = u_m - T_\lambda(u_m). \tag{6}$$

This correspondence between the thresholding function and the cost function can be seen by computing the derivative of E with respect to the active coefficients, $\{a_m\}$ (see Appendix A). If (6) holds, then letting the internal states $\{u_m\}$ evolve according to $\dot{u}_m \propto -\frac{\partial E}{\partial a_m}$ yields the equation for the internal state dynamics



(a) (b)

Figure 3: (a) The top 200 coefficients from a BPDN solver sorted by magnitude. (b) The same coefficients, sorted according to the magnitude ordering of the SLCA coefficients. While there is a gross decreasing trend noticeable, the largest SLCA coefficients are not in the same locations as the largest BPDN coefficients. While the solutions have equivalent energy functions, the two sets of coefficients differ significantly.

above in (4). As long as a_m and u_m are related by a monotonically increasing function, the $\{a_m\}$ will also descend the energy function E. This method for showing the correspondence between a dynamic system and an energy function is essentially the same procedure used by Hopfield [29] to establish network dynamics in associative memory systems.

We focus specifically on the cost functions associated with thresholding activation functions. Thresholding functions limit the lateral inhibition by allowing only "strong" units to suppress other units and forcing most coefficients to be identically zero. For our purposes, thresholding functions $T_\lambda(\cdot)$ have two distinct behaviors over their range: they are essentially linear with unit slope above threshold λ, and essentially zero below threshold. Among many reasonable choices for thresholding functions, we use a smooth sigmoidal function

$$T_{(\alpha,\gamma,\lambda)}(u_m) = \frac{u_m - \alpha\lambda}{1 + e^{-\gamma(u_m - \lambda)}}, \tag{7}$$

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. An example sigmoidal thresholding function is shown in Figure 2a. We are particularly interested in the limit of this thresholding function as $\gamma \to \infty$, a piecewise linear function we denote as the *ideal thresholding function*. In the signal processing literature, $T_{(0,\infty,\lambda)}(\cdot) = \lim_{\gamma\to\infty} T_{(0,\gamma,\lambda)}(\cdot)$ is known as a "hard" thresholding function and $T_{(1,\infty,\lambda)}(\cdot) = \lim_{\gamma\to\infty} T_{(1,\gamma,\lambda)}(\cdot)$ is known as a "soft" thresholding function [30].

Combining (6) and (7), we can integrate numerically to determine the cost function corresponding to $T_{(\alpha,\gamma,\lambda)}(\cdot)$, shown in Figure 2b. For the ideal threshold functions we derive a corresponding *ideal cost function*,

$$C_{(\alpha,\infty,\lambda)}(a_m) = \frac{(1-\alpha)^2\lambda}{2} + \alpha\,|a_m|\,. \tag{8}$$

Details of this derivation are in Appendix B. Note that unless $\alpha = 1$ the ideal cost function has a gap because active coefficients cannot take all possible values, $|a_m| \notin [0, (1-\alpha)\,\lambda]$ (i.e., the ideal thresholding function is not technically invertible).

3.3 Special case: Soft-thresholding locally competitive algorithm (SLCA)

As we see in Section 3.2, a LCA can optimize a variety of different sparsity measures depending on the choice of thresholding function. One special case that we will look at in detail is the soft thresholding function,

corresponding to $\alpha = 1$ and shown graphically in Figures 2e and 2f. The soft-thresholding locally competitive algorithm (SLCA) applies the ℓ^1 norm as a cost function on the active coefficients,

$$C_{(1,\infty,\lambda)}(a_m) = |a_m|.$$

Thus, the SLCA minimizes a combination of the ℓ^2 reconstruction error and ℓ^1 sparsity penalty, meaning that it is simply another solution method for the general BPDN problem described in Section 2. Despite minimizing the same convex energy function, SLCA and BPDN solvers will find different sets of coefficients, as illustrated in Figure 3. The connection between soft-thresholding and BPDN is well-known in the case of orthonormal dictionaries [9], and recent results [31] have given some justification for using soft-thresholding in overcomplete dictionaries. The SLCA provides another formal connection between the soft-thresholding function and the ℓ^1 cost function.

Though BPDN uses the ℓ^1-norm as its sparsity penalty, we often expect many of the resulting coefficients to be identically zero (especially when $M \gg N$). However, like many numerical methods, interior point BPDN solvers will drive coefficients *toward* zero but will never make them identically zero. While an ad hoc threshold could be applied to the results of a BPDN solver, the SLCA has the advantage of incorporating a natural thresholding function that keeps coefficients identically zero during the computation unless they become large enough to go active. In other words, while BPDN solvers often start with many non-zero coefficients and try to force coefficients down, the SLCA starts with all coefficients equal to zero and only lets a few grow up. This advantage is especially important for neural systems that must expend energy for non-zero values throughout the entire computation.

3.4 Special case: Hard-thresholding locally competitive algorithm (HLCA)

Another important special case is the hard thresholding function, corresponding to $\alpha = 0$ and shown graphically in Figures 2c and 2d. Using the relationship in (6), we see that this hard-thresholding locally competitive algorithm (HLCA) applies an ℓ^0-like cost function by using a constant penalty regardless of magnitude,

$$C_{(0,\infty,\lambda)}(a_m) = \frac{\lambda}{2}\, I(|a_m| > \lambda)\,,$$

where $I(\cdot)$ is the indicator function evaluating to 1 if the argument is true and 0 if the argument is false.

As with the SLCA, the HLCA also has connections to known sparse approximation principles. If node m is fully charged, the inhibition signal it sends to other nodes would be exactly the same as the update step when the m^{th} node is chosen in the MP algorithm. However, due to the continuous competition between nodes before they are fully charged, the HLCA will not find the same sparse representation as MP in general.

As a demonstration of the power of competitive algorithms over greedy algorithms, consider a canonical example used to illustrate the shortcomings of greedy algorithms [9,32]. For this example, specify a positive integer $K < N$ and construct a dictionary $\mathcal{D}$ with $M = N + 1$ elements to have the following form:

$$\boldsymbol{\phi}_m = \begin{cases} e_m & \text{if } m \leq N \\ \sum_{n=1}^{K} \kappa e_n + \sum_{n=K+1}^{N} \left(\kappa/(n-K)\right) e_n & \text{if } m = N+1, \end{cases}$$

where e_m is the canonical basis element (i.e., it contains a single 1 in the m^{th} location) and κ is a constant to make the vectors have unit norm. In words, the dictionary is essentially the canonical basis but with one "extra" element that is a decaying combination of all other elements (illustrated in Figure 4, with $N = 20$ and $K = 5$). The input signal is sparsely represented in the first K dictionary elements, $\boldsymbol{s} = \sum_{m=1}^{K} \frac{1}{\sqrt{K}} e_m$. The first MP iteration chooses $\boldsymbol{\phi}_M$, introducing a residual with decaying terms. Even though $\boldsymbol{s}$ has an exact representation in K elements, MP iterates forever trying to atone for its bad initial choice. This pathological example highlights the main danger in purely greedy algorithms such as MP: optimal local choices cannot be corrected even if they induce undesirable behavior throughout the rest of the coefficients. In contrast, the LCA competition allows the system to correct bad choices and eliminate them if they are sufficiently globally undesirable. In this example, the HLCA initially activates the M^{th} node. The collective inhibition from nodes $1, \ldots, K$ causes this extra node to eventually die away leaving the optimal set of coefficients.

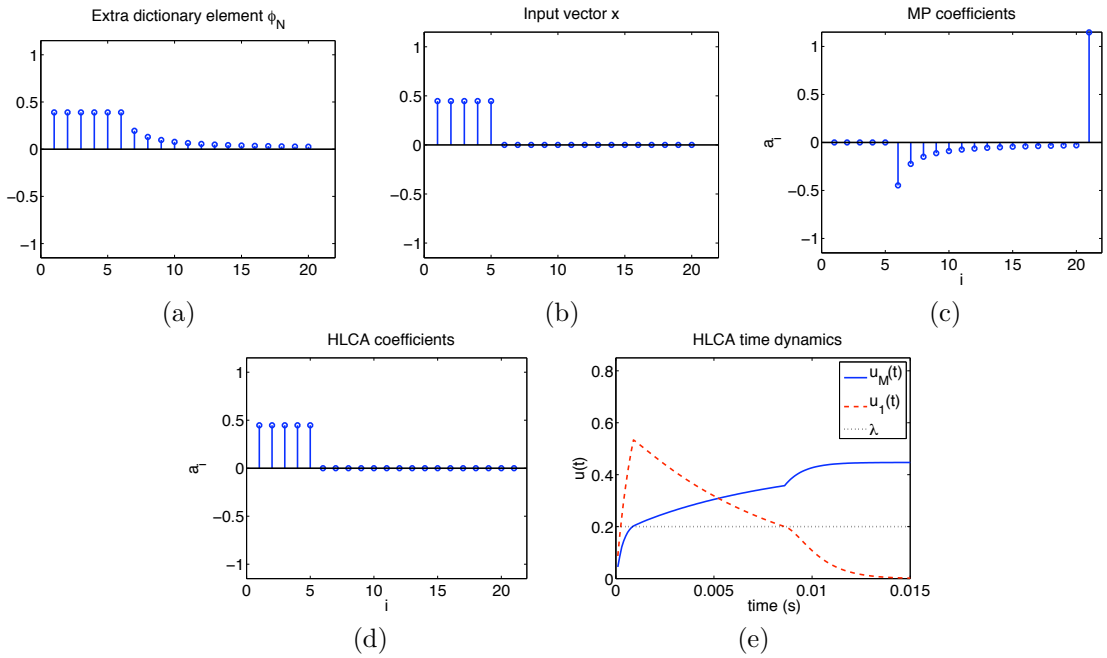


Figure 4: (a) The dictionary in this example has one "extra" element that consists of decaying combinations of all other dictionary elements. (b) The input vector has a sparse representation in just a few dictionary elements. (c) MP initially chooses the "extra" dictionary element, preventing it from finding the optimally sparse representation (coefficients shown after 100 iterations). (d) In contrast, the HLCA system finds the optimally sparse coefficients. (e) The time-dynamics of the HLCA system illustrate its advantage. The "extra" dictionary element is the first node to activate, followed shortly by the nodes corresponding to the optimal coefficients. The collective inhibition of the optimal nodes causes the "extra" node to die away.

3.5 Closely related methods

Fischer, et al. [25] developed an algorithm that has a similar flavor to our LCAs but with significant differences. Their iterative procedure generates new coefficients by selecting a subset of old coefficients and adding adjustments to retain perfect reconstruction. Coefficients are selected by thresholding a separate set of state variables representing coefficient values over previous iterations. While this system does have parallel characteristics, keeping a set of auxiliary state variables not directly related to the current coefficients requires two separate neural populations. Additionally, this algorithm would not map naturally to time-varying inputs because the state variables would be reset with each new input.

Kingsbury and Reeves [26] have described a sparsity-inducing iterative scheme for complex wavelets that is similar to a discrete approximation of the HLCA. Their algorithm is described by a system of equations that has two primary differences from the HLCA system equations. First, instead of the "charging up" behavior used by the HLCA, their algorithm starts with projecting the stimulus onto the basis set and then iteratively altering those coefficients. If this algorithm were initialized with all zero coefficients, then the first iteration would jump directly to the projection coefficients and continue from there. It is not immediately clear if this strategy enables the algorithm to adapt smoothly to tracking stimulus changes. The second significant difference is that their algorithm uses both the dictionary and the dual set (i.e., the analysis and synthesis vectors in the frame representation [33]). Calculating the dual vectors is computationally expensive and recalculation must be performed whenever the dictionary changes. Such a system is less resilient to adaptations or failures. This algorithm is also very similar to an iterative scheme described by Herrity et al. [27], though the Herrity algorithm does not employ the dual dictionary.

Rehn and Sommer [28] have also recently described a neurally plausible sparse coding mechanism that is similar to our approach. This algorithm is described by a system of equations that have three primary differences from the HLCA system equations. First, as with the work in [26], this algorithm also does not use a "charging up" approach and but rather works by modifying the projection coefficients. Second, this system uses a variable threshold for activating each unit, making it easier for neurons to become active when they have a larger driving input (i.e., neurons with a larger input have a lower threshold to become active). Finally, this system uses a constant amount of inhibition that is not related to the current activity level of the active neuron.

4 LCA system properties

To be a neurally plausible sparse coding mechanism, LCAs must exhibit several critical properties: the dynamic systems must remain stable under normal operating conditions, the system must produce sparse coefficients that represent the stimulus with low error, and coefficient sequences must exhibit regularity in response to time-varying inputs. In this section we show that LCAs exhibit good characteristics in each of these three areas. We focus our analysis on the HLCA both because it yields the most interesting results and because it is notationally the cleanest to discuss. In general, the analysis principles we use will apply to all LCAs through straightforward (through perhaps laborious) extensions.

4.1 Stability

Any proposed neural system must remain well-behaved under normal conditions. Linear systems theory has an intuitive and uniform notion of stability: well-behaved systems produce outputs with bounded energy as long as the input also has bounded energy [34]. Unfortunately, no such unifying concept of stability exists for non-linear systems [35]. Instead, non-linear systems are characterized in a variety of ways, including their input-output relationship and their behavior near an equilibrium point $\boldsymbol{u}^*$ where $f(\boldsymbol{u}^*) = 0$.

4.1.1 LCA stability criteria

The various stability notions for the LCA will depend on a common criteria. Define $\mathcal{M}_{\boldsymbol{u}(t)} \subseteq [1, \ldots, M]$ as the set of nodes that are above threshold in the internal state vector $\boldsymbol{u}(t)$, $\mathcal{M}_{\boldsymbol{u}(t)} = \{m : |u_m(t)| \geq \lambda\}$.

We say that the LCA 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 criteria will be important for several later results. In the meantime, it makes some intuitive sense that this condition is important to a LCA: if a collection of linearly dependent nodes are active simultaneously, the nodes could have active (and possibly growing) coefficients with a net effect of no contribution to the reconstruction.

Satisfying the stability criteria comes down to two questions: how likely are small subsets of dictionary elements to be linearly dependent, and how likely is the LCA to activate such a subset? Small subsets of dictionary elements are unlikely to be linearly dependent unless the dictionary is designed with this property. This result has been quantified recently for random dictionaries in work related to the field of "compressive sensing" [36] (c.f. [37] for an equivalent problem). Though these results are for random dictionaries, they do give insight for large ambient signal dimensions and provide analytic tools to explore a specific dictionary.

Regardless of the properties of the dictionary, sparse coding systems are actively trying to select dictionary subsets so that they can use many *fewer* coefficients then the dimension of the signal space, $\left|\mathcal{M}_{\boldsymbol{u}(t)}\right| \ll N \ll M$. While the LCA lateral inhibition signals discourage linear dependent sets from activating, the stability criteria could be violated when a collection of nodes becomes active too quickly, before inhibition can take effect. In practice, this situation could occur when the threshold is too low compared to the system time constant. We expect (and our simulations confirm) that under normal operating conditions with biologically relevant parameters, the LCAs satisfy the stability criteria.

4.1.2 Equilibrium points

In a LCA presented with a static input, we look to the steady-state response (where $\dot{\boldsymbol{u}}\left(t\right)=0$) to determine the coefficients. The character of the equilibrium points $\boldsymbol{u}^*$ ($f(\boldsymbol{u}^*)=0$) and the system's behavior in a neighborhood around an equilibrium point provides one way to ensure that a system is well-behaved. Consider the ball around an equilibrium point $B_\epsilon\left(\boldsymbol{u}^*\right)=\{\boldsymbol{u}:||\boldsymbol{u}-\boldsymbol{u}^*||<\epsilon\}$. The tools of Lyapunov stability [35] are frequently employed to ask an intuitive question: if the system is perturbed within this ball, does it then run away, stay where it is, or get attracted back? Specifically, a system is said to be *locally asymptotically stable* [38] at an equilibrium point $\boldsymbol{u}^*$ if one can specify a $\epsilon>0$ such that

$$\boldsymbol{u}(0)\in B_\epsilon\left(\boldsymbol{u}^*\right) \implies \lim_{t\to\infty}\boldsymbol{u}(t)=\boldsymbol{u}^*.$$

Previous research [39–41] has used the tools of Lyapunov functions [35] to study a Hopfield network [29] similar to the LCA architecture, but all of these analyses make assumptions that do not encompass the ideal thresholding functions (e.g., they are continuously differentiable and/or monotone increasing). In Appendix C.1 we show that as long as the stability criteria is met, 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.

The conditions that hold "almost certainly" are true as long as none of the equilibria have components identically equal to the threshold, $(u_m^*\neq\lambda,\forall m)$, which holds with overwhelming probability. 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. Asymptotic stability also implies a notion of robustness, guaranteeing that the system will remain well-behaved even under perturbations (Theorems 2.8 and 2.9 in [38]).

4.1.3 Input-output stability

In physical systems it is important that the energy of both internal and external signals remain bounded for bounded inputs. One intuitive approach to ensuring output stability is to examine the energy function E. We show in Appendix C.2 that for non-decreasing threshold functions, the energy function is non-increasing

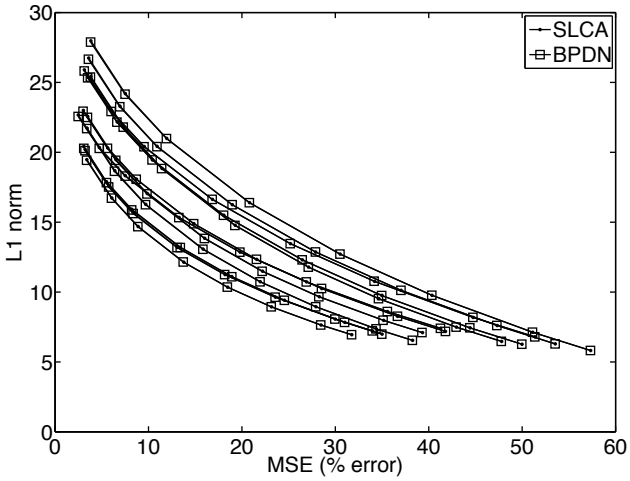


Figure 5: SLCA and BPDN coefficients for a series of standard test images. Each line on the plot indicates the tradeoff between MSE and ℓ^1 coefficient norm as λ is varied. The results for SLCA and BPDN overlap exactly, illustrating that the systems are finding equivalent minima of the energy function.

($\frac{d}{dt}E(t) \leq 0$) for fixed inputs. While this is encouraging, it does not guarantee input-output stability. To appreciate this effect, note that the HLCA cost function is constant for nodes above threshold — nothing explicitly keeps a node from growing without bound once it is active.

While there is no universal input-output stability test for general non-linear systems, we observe that the LCA system equation is linear and fixed until a unit crosses threshold. A branch of control theory specifically addresses these *switched systems* [42]. Results from this field indicate that input-output stability can be guaranteed if the individual linear subsystems are stable, and the system doesn't switch "too fast" between these subsystems [43]. In Appendix C.2 we give a precise mathematical statement of this input-output stability and show that the HLCA linear subsystems are individually stable if and only if the stability criteria are met. Therefore, the HLCA is input-output stable as long as nodes are limited in how fast they can change states. We expect that the infinitely fast switching condition is avoided in practice either by the physical principles of the system implementation or through an explicit hysteresis in the thresholding function.

4.2 Sparsity and representation error

Viewing the sparse approximation problem through the lens of rate-distortion theory [44], the most powerful algorithm produces the lowest reconstruction MSE for a given sparsity. When the sparsity measure is the ℓ^1 norm, there is little variation in the solution quality: solving the convex optimization principle in (3) produces a solution with minimum MSE for that ℓ^1 norm. Therefore, even though SLCA produces different coefficients from an interior point BPDN solver, they will both achieve optimality with regard to ℓ^1 sparsity (demonstrated in Figure 5). The SLCA is unique in that it is the only LCA corresponding to a convex energy function.

Despite the analytic appeal of the ℓ^1 norm as a sparsity measure, many systems concerned with energy minimization (including neural systems) likely have an interest in minimizing the ℓ^0 norm of the coefficients. The HLCA is appealing because of its ℓ^0-like sparsity penalty, but this objective function is not convex and the HLCA may find a local minimum. We will show that while HLCA cannot guarantee the ℓ^0 sparsest solution, it produces coefficients that demonstrate comparable sparsity to MP for natural images.

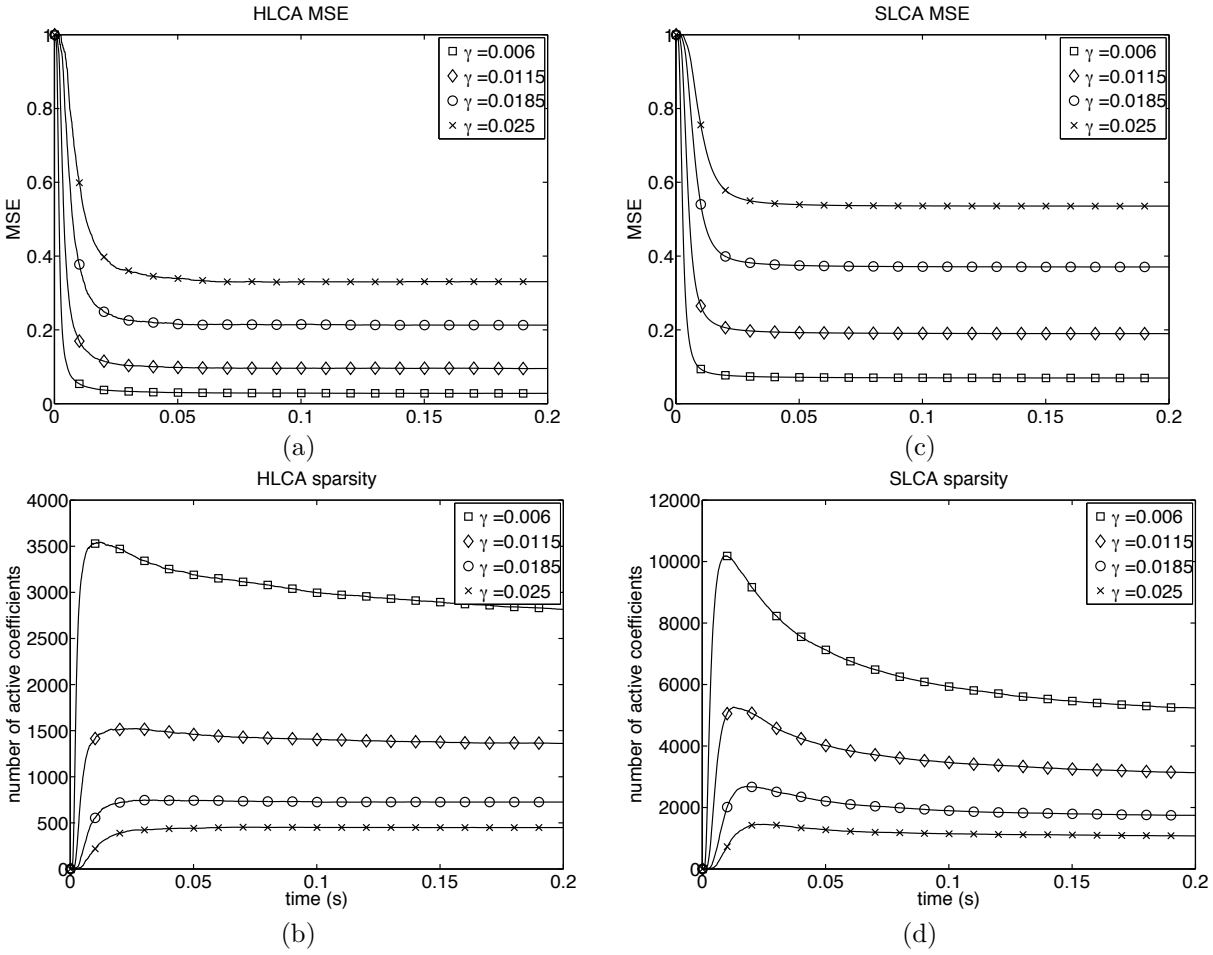
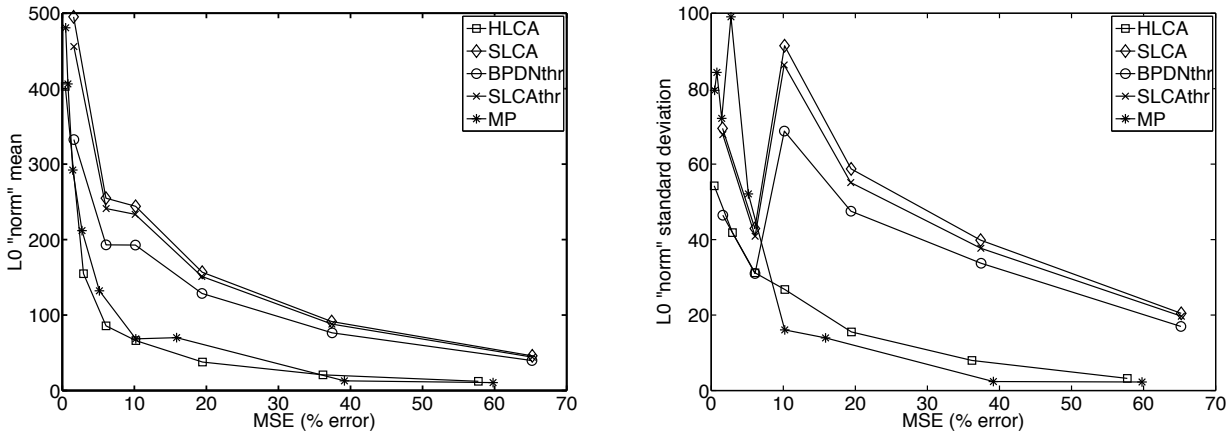


Figure 6: The time response of the HLCA and SLCA ($\tau = 10$ ms) for a single fixed image patch. (a) The MSE decay and (b) the ℓ^0 sparsity for HLCA. (c) The MSE decay and (d) the ℓ^0 sparsity for SLCA. The error converges within 1-2 time constants and the sparsity often approximately converges within 3-4 time constants. In some cases sparsity is reduced with a longer running time.



(a) (b)

Figure 7: Mean tradeoff between MSE and ℓ^0-sparsity for normalized (32×32) patches from a standard set of test images. For a given MSE range, we plot the mean (a) and standard deviation (b) of the ℓ^0 sparsity.

Insight about the HLCA reconstruction fidelity comes from rewriting the LCA system equation

$$\dot{\boldsymbol{u}}\left(t\right) = \frac{1}{\tau}\left[\Phi^t\left(\boldsymbol{s}(t) - \widehat{\boldsymbol{s}}\left(t\right)\right) - \boldsymbol{u}(t) + T_{(\alpha,\infty,\lambda)}(\boldsymbol{u}(t))\right]. \tag{9}$$

For a constant input, HLCA equilibrium points ($\dot{\boldsymbol{u}}\left(t\right) = 0$) occur when the residual error is orthogonal to active nodes and balanced with the internal state variables of inactive nodes.

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

Therefore, when HLCA converges the coefficients will perfectly reconstruct the component of the input signal that projects onto the subspace spanned by the final set of active nodes. Using standard results from frame theory [33], we can bound the HLCA reconstruction MSE in terms of the set of inactive nodes

$$||\boldsymbol{s}(t) - \widehat{\boldsymbol{s}}\left(t\right)||^2 \leq \frac{1}{\eta_{\min}} \sum_{m \notin \mathcal{M}_{\boldsymbol{u}(t)}} |\langle \boldsymbol{\phi}_m, \boldsymbol{s}(t) - \widehat{\boldsymbol{s}}\left(t\right)\rangle|^2 \leq \frac{\left(M - |\mathcal{M}_{\boldsymbol{u}(t)}|\right)\lambda^2}{\eta_{\min}},$$

where $\eta_{\min}$ is the minimum eigenvalue of $(\Phi\Phi^t)$.

Though the HLCA is not guaranteed to find the globally optimal ℓ^0 sparsest solution, we would like to ensure that the system is still being reasonably efficient. While the system nonlinearity makes it impossible to analytically determine the LCA steady-state coefficients, 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. Using linear systems theory we can calculate the steady-state response $\widetilde{\boldsymbol{u}}^{\mathcal{M}} = \lim_{t\to\infty} \boldsymbol{u}(t)$ assuming that $\mathcal{M}$ stays fixed. If $|\widetilde{u}_m^{\mathcal{M}}| < \lambda$ for any $m \in \mathcal{M}$ or if $|\widetilde{u}_m^{\mathcal{M}}| > \lambda$ for any $m \notin \mathcal{M}$, then $\mathcal{M}$ cannot describe the set of active nodes in the steady-state response and we call it *inconsistent*. We show in Appendix D that when the stability criteria are met, the following statement is true for the HLCA: *If $\boldsymbol{s} = \boldsymbol{\phi}_m$, then any set of active coefficients $\mathcal{M}$ with $m \in \mathcal{M}$ and $|\mathcal{M}| > 1$ is inconsistent.* 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 system does not needlessly employ both the optimal and extraneous nodes.

We have also verified numerically that the LCAs achieve a combination of error and sparsity comparable with known methods. In these experiments we simulated the LCAs until steady-state on normalized (32×32) test image patches. We used a biologically plausible membrane time constant of $\tau = 10$ ms [45] and a dictionary consisting of the bandpass band of a steerable pyramid [46] with one level and four orientation bands (i.e., the dictionary is approximately four times overcomplete).

The test image patches are created by filtering out the lowpass and residual bands of the steerable pyramid, so that any given test image can be fully described using the dictionary.[2] Figure 6 shows the time evolution of the reconstruction MSE and ℓ^0 sparsity for SLCA and HLCA responding to an individual image, and Figure 7 shows the mean steady-state tradeoff between ℓ^0 sparsity and MSE. For comparison, we also plotted the results obtained from using 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 the connections between the HLCA and MP were pointed out in Section 3.4, these are very different systems and there is no reason to expect them to produce the same coefficients. Additionally, note that the SLCA is producing coefficients that are nearly as ℓ^0-sparse as what we can achieved by thresholding the results of a BPDN solver even though the SLCA keeps most coefficients zero throughout the calculation.

4.3 Time-varying stimuli

4.3.1 Inertia

Biological sensory systems are faced with constantly changing stimuli due to both external movement and internal factors (e.g., organism movement, eye saccades, etc.). Neurally plausible sparse coding systems must therefore efficiently handle time-varying stimuli such as video sequences. One strategy for applying a typical sparse approximation scheme to a video sequence is to independently find a set of coefficients representing each frame using a 2-D (spatial) dictionary, thereby producing a sequence of coefficients (i.e., a coefficient time series) for each dictionary element. However, optimizing a sparsity criteria at each frame can cause the selected coefficients to vary significantly from one frame to the next even when there are relatively small changes in the input. Greedy algorithms are especially prone to generating such "brittle" representations. This temporal irregularity is undesirable for higher level systems trying to interpret the content of the scene because characteristics of a smoothly varying stimulus would not be reflected in the encoding.

In contrast, LCAs naturally produce smoothly changing outputs in response to smoothly changing time-varying inputs. Assuming that the system time constant τ is faster than the temporal changes in the stimulus, the LCA will evolve to capture the stimulus change and converge to a new representation. While local minima in an energy function are typically problematic, the LCAs can use these local minima to find coefficients that are "close" to their previous coefficients even if they are not optimally sparse. While permitting suboptimal coefficient sparsity, this property allows the LCA to exhibit inertia that smooths the coefficient sequences.

The inertia property exhibited in LCAs can be seen by focusing on a single node in the system equation (9):

$$\dot{u}_m(t) = \frac{1}{\tau} \begin{cases} \langle \boldsymbol{\phi}_m, (\boldsymbol{s}(t) - \widehat{\boldsymbol{s}}\,(t)) \rangle - u_m(t) & \text{when } |u_m(t)\,| < \lambda \\ \langle \boldsymbol{\phi}_m, (\boldsymbol{s}(t) - \widehat{\boldsymbol{s}}\,(t)) \rangle - \alpha\lambda & \text{when } |u_m(t)\,| \geq \lambda. \end{cases}$$

A new residual signal drives the coefficient higher but suffers an additive penalty. Inactive coefficients suffer an increasing penalty as they get closer to threshold while active coefficients only suffer a constant penalty $\alpha\lambda$ that can be very small (e.g., the HLCA has $\alpha\lambda = 0$). This property induces a "king of the hill" effect: when a new residual appears, active nodes move virtually unimpeded to represent it while inactive nodes are penalized until they reach threshold. This inertia encourages inactive nodes to remain inactive unless the active nodes cannot adequately represent the new input.

[2] We eliminate the lowpass band because it accounts for a large fraction of the total image energy in just a few dictionary elements and it is unlikely that much gain could be achieved by sparsifying these coefficients.

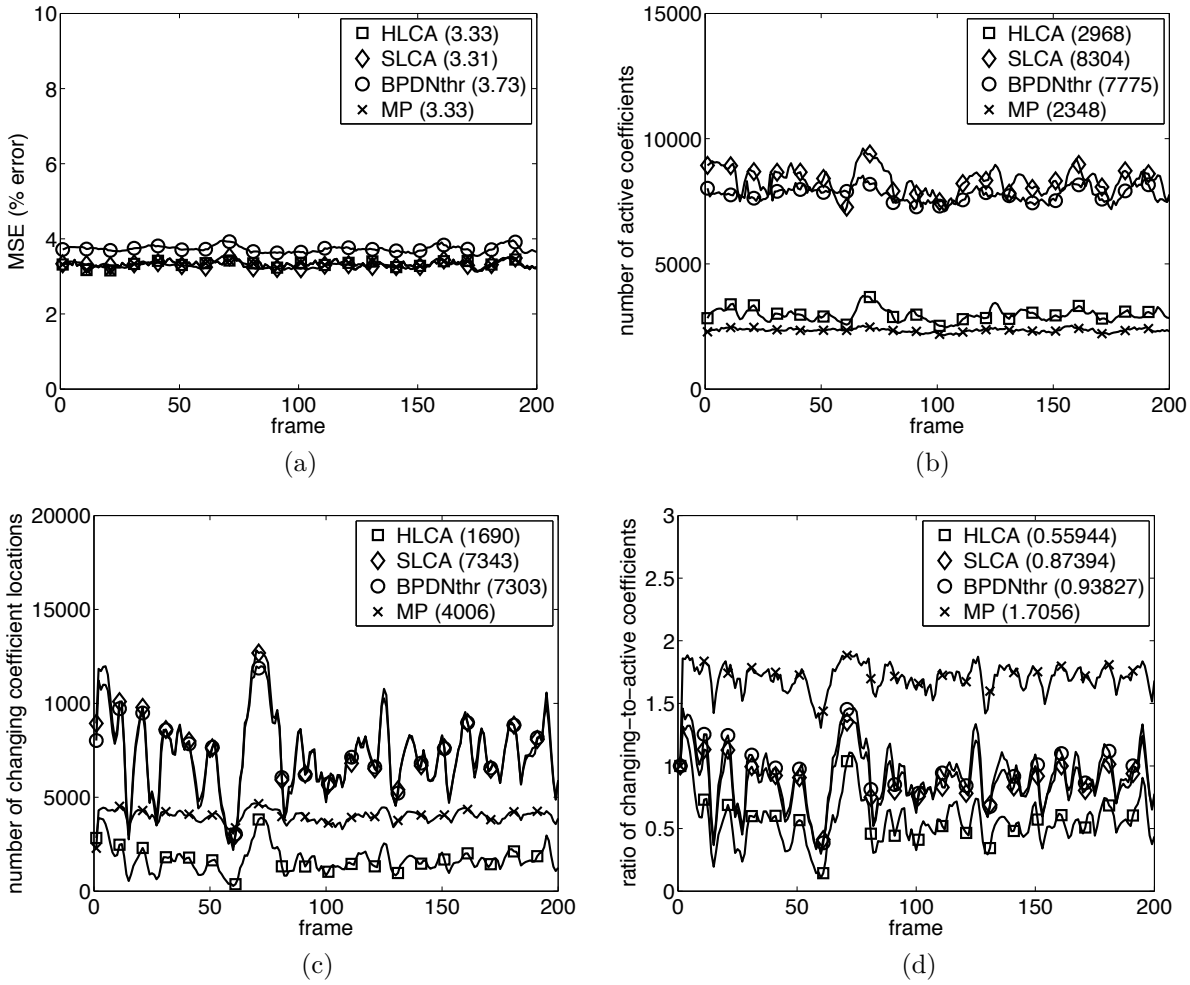


Figure 8: The HLCA and SLCA systems simulated on 200 frames of the "foreman" test video sequence. For comparison, MP coefficients and thresholded BPDN coefficients are also shown. Average values for each system are notated in the legend. (a) Per-frame MSE for each coding scheme, designed to be approximately equal. (b) The number of active coefficients in each frame. (c) The number of changing coefficient locations for each frame, including the number of inactive nodes becoming active and the number of active nodes becoming inactive. (d) The ratio of changing coefficients to active coefficients. A ratio near 2 (such as with MP) means that almost 100% of the coefficient locations are new at each frame. A ratio near 0.5 (such as with HLCA) means that approximately 25% of the coefficients are new at each frame.

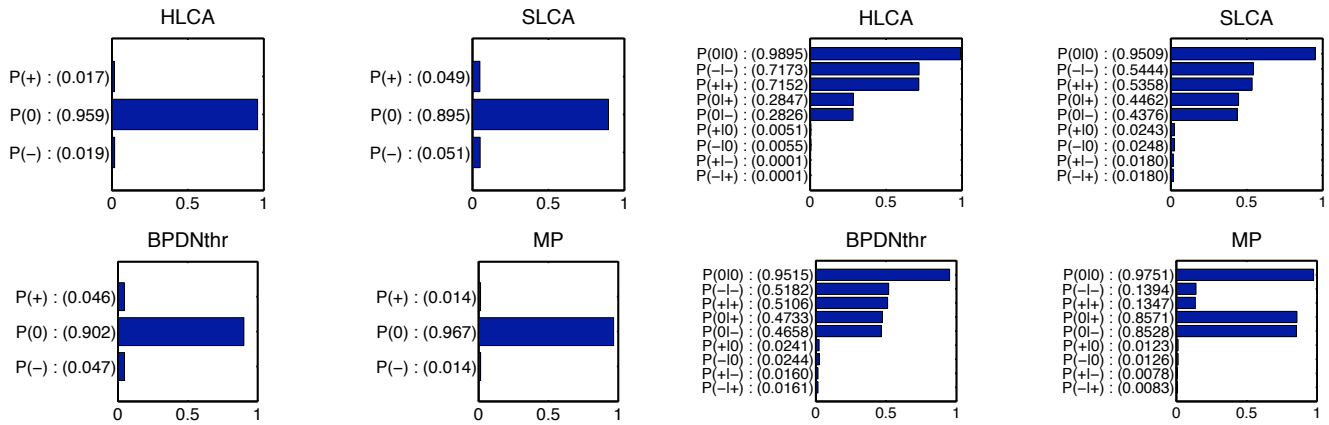


Figure 9: (a) The marginal probabilities denoting the fraction of the time coefficients spent in the three states: negative, zero and positive ($-$, 0, and $+$). (b) The transition probabilities denoting the probability of a node in one state transitioning to another state on the next frame. For example, $P\,(0|+)$ is the probability that a node with an active positive coefficient will be inactive (i.e., zero) in the next frame.

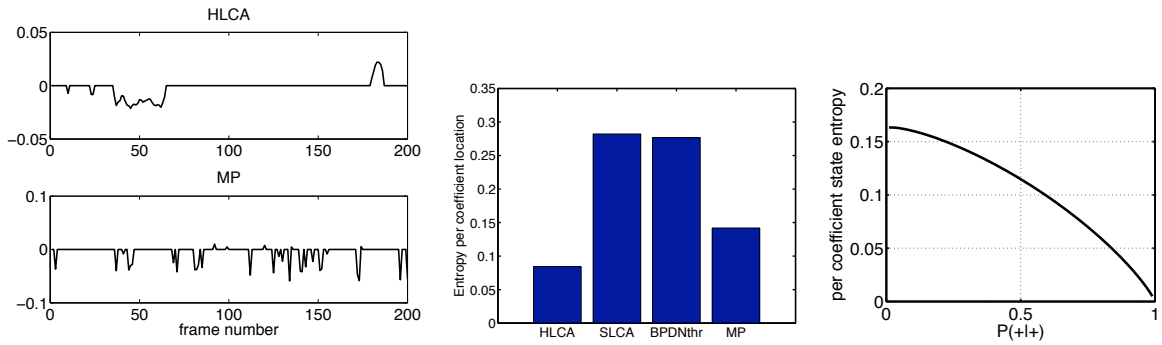
To illustrate this inertia, we applied the LCAs to a sequence of 144×144 pixel, bandpass filtered, normalized frames from the standard "foreman" test video sequence with the same experimental setup described in Section 4.2. The LCA input is switched to the next video frame every (simulated) 1/30 seconds. The results are shown in Figure 8, along with comparisons to MP and BPDN applied independently on each frame. The changing coefficient locations are nodes that either became active or inactive at each frame. Mathematically, the number of changing coefficients at frame n is: $\left|\mathcal{M}_{\boldsymbol{u}(n-1)} \oplus \mathcal{M}_{\boldsymbol{u}(n)}\right|$, where $\oplus$ is the "exclusive OR" operator and $\boldsymbol{u}(n)$ are the internal state variables at the end of the simulation for frame n.

This simulation highlights that the HLCA uses approximately the same number of active coefficients as MP but is much more efficient in how it chooses those coefficients. The HLCA is significantly more likely to re-use active coefficient locations from the previous frame without making significant sacrifices in the sparsity of the solution. This difference is highlighted when looking at the ratio of the number of changing coefficients to the number of active coefficients, $\left|\mathcal{M}_{\boldsymbol{u}(n-1)} \oplus \mathcal{M}_{\boldsymbol{u}(n)}\right| / \left|\mathcal{M}_{\boldsymbol{u}(n)}\right|$. MP has a ratio of 1.7, meaning that MP is finding almost an entirely new set of active coefficient locations for each frame. The HLCA has a ratio of 0.5, meaning that it is changing approximately 25% of its coefficient locations at each frame. SLCA and BPDNthr have approximately the same performance, with regularity falling between HLCA and MP. Though the two systems can calculate different coefficients, the convexity of the energy function appears to be limiting the coefficient choices enough so that SLCA cannot smooth the coefficient time series substantially more than BPDNthr.

4.3.2 Markov state transitions

The simulation results indicate that the HLCA is producing time series coefficients that are much more regular than MP. This regularity is visualized in Figure 10 by looking at the time-series of example HLCA and MP coefficients. Note that though the two coding schemes produce roughly the same number of non-zero entries, the HLCA does much better than MP at clustering the values into consecutive runs of positive or negative values. This type of smoothness better reflects the regularity in the natural video sequence input.

We can quantify this increased regularity by examining the Markov state transitions. Specifically, each



(a) (b) (c)

Figure 10: (a) An example time-series coefficient for the HLCA and MP (top and bottom, respectively) encodings for the test video sequence. HLCA clusters non-zero entries together into longer runs while MP switches more often between states. (b) The empirical conditional entropy of the coefficient states (−,0,+) during the test video sequence. (c) The conditional entropy is calculated analytically while varying $P(+|+)$ and equalizing all other transition probabilities to the values seen in HLCA and MP. The tendency of a system to group non-zero states together is the most important factor in determining the entropy.

coefficient time-series is Markov chain [47] with three possible states at frame n:

$$\sigma_m(n) = \begin{cases} - & \text{if } u_m(n) < -\lambda \\ 0 & \text{if } -\lambda \leq u_m(n) \leq \lambda \\ + & \text{if } u_m(n) > \lambda. \end{cases}$$

Figure 9 shows the marginal probabilities $P(\cdot)$ of the states and the conditional probabilities $P(\cdot|\cdot)$ of moving to a state given the previous state. The HLCA and MP are equally likely to have non-zero states, but the HLCA is over five times more likely than MP to have a positive coefficient stay positive ($P(+|+)$). Also, though the absolute probabilities are small, MP is roughly two orders of magnitude more likely to have a coefficient swing from positive to negative ($P(-|+)$) and vice-versa ($P(-|+)$).

To quantify the regularity of the active coefficient locations we calculate the entropy [48] of the coefficient states at frame n conditioned on the coefficient states at frame (n-1):

$$\begin{aligned} H(\sigma_m(n)\,|\sigma_m(n-1)) = &-P(+)\left[P(-|+) + P(0|+) + P(+|+)\right] \\ &- P(0)\left[P(-|0) + P(0|0) + P(+|0)\right] \\ &- P(-)\left[P(-|-) + P(0|-) + P(+|-)\right], \end{aligned} \tag{10}$$

plotted in Figure 10. This conditional entropy indicates how much uncertainty there is about the status of the current coefficients given the coefficients from the previous frame. Note that the conditional entropy for MP is almost double the entropy for the HLCA, while SLCA is again similar to BPDNthr. The principle contributing factor to the conditional entropy appears to be the probability a non-zero node remains in the same state (i.e., $P(+|+)$ and $P(-|-)$). To illustrate, Figure 10 shows the change in conditional entropy is almost linear with varying $P(+|+)$ (assuming $P(-|-) = P(+|+)$ and all other transition probabilities are kept fixed).

The substantial decrease in the conditional entropy for the HLCA compared to MP quantifies the increased regularity in time-series coefficients due to the inertial properties of the LCAs. The HLCA in particular encourages coefficients to maintain their present state (i.e., active or inactive) if it is possible to find an adequate stimulus representation. While some sparsity may be sacrificed in this strategy, the smoothness induced in the coefficients by grouping active states together in time better reflects the character of the natural time-varying stimuli and could be useful for higher-level computations.

5 Conclusions and future work

Sparse approximation is an important paradigm in neural coding, though the plausible mechanisms to achieve these codes have remained unknown. We have proposed an architecture for a class of locally competitive algorithms that solve a series of sparse approximation problems (including BPDN as a special case). These LCAs are neurally plausible in the sense that they can be implemented using a parallel network of simple elements that match well with the known neurobiology of sensory cortical areas such as V1. Though these LCA systems are non-linear, we have shown that they are well-behaved under nominal operating conditions.

While the LCA systems (other than SLCA) are not generally guaranteed to find a globally optimal solution to their energy function, we have proven that the systems will be efficient in a meaningful sense. In practice, we have seen that the LCAs are very efficient in producing coefficients for natural images. The SLCA system produces coefficients with sparsity levels comparable to BPDN solvers, but uses a more efficient and natural physical implementation than these solvers. Perhaps most interestingly, the HLCA produces coefficients with almost identical sparsity as MP. This is significant because greedy methods such as MP are widely used in signal processing practice because of their efficiency, but HLCA offers a much more natural neural implementation.

LCAs are particularly appropriate for time-varying data such as video sequences. The LCA ODE not only encourages sparsity but also introduces an inertia into the coefficient time-series that we have quantified using both raw counts of changing coefficient location and through the conditional entropy of the coefficient states. By allowing suboptimal sparsity in exchange for more regularity in the set of active coefficients, the LCAs produce smoother coefficient sequences that better reflect the structure of the time-varying stimulus. This property could prove valuable for higher levels of analysis that are trying to interpret the sensory scene from a set of sparse coefficients.

The current limitations of neurophysiological recording mean that exploring the sparse coding hypothesis must rely on testing specific proposed mechanisms. Though the LCAs we have proposed appear to map well to known neural architectures, they still lack the detail necessary to be experimentally testable. We will continue to build on this work by mapping these LCAs to a detailed neurobiological population coding model that can produce verifiable predictions. Furthermore, the combination of sparsity and regularity induced in LCA coefficients may serve as a critical front-end stimulus representation that enables visual perceptual tasks, including pattern recognition, source separation and object tracking.

By using simple computational primitives, LCAs also have the benefit of being implementable in analog hardware. An imaging system using VLSI to implement LCAs as a data collection front end has the potential to be extremely fast and energy efficient. Instead of digitizing all of the sensed data and using digital hardware to run a compression algorithm, analog processing would compress the data into sparse coefficients *before* digitization. In this system, time and energy resources would only be spent digitizing coefficients that are a critical component in the signal representation.

Acknowledgements

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 DSP Leadership University Program.

A Relating cost functions and threshold functions

To see the correspondence between a particular choice of a threshold function $T_\lambda(\cdot)$ and the sparsity-inducing cost function $C(\cdot)$, we begin by assuming we want to minimize an energy function of the form:

$$\begin{aligned} E &= \frac{1}{2}\left|\left|\boldsymbol{s}-\widehat{\boldsymbol{s}}\right|\right|^2 + \lambda\sum_m C(a_m) \\ &= \frac{1}{2}\left(\boldsymbol{s}^t\boldsymbol{s} - 2\boldsymbol{b}^t\boldsymbol{a} + \boldsymbol{a}^t\Phi^t\Phi\boldsymbol{a}\right) + \lambda\sum_m C(a_m)\,. \end{aligned}$$

For simplicity, we suppress the time variable in the notation. To find the changes in the active coefficients $\{a_m\}$ that will most significantly minimize the energy function, we take the derivative of the energy function with respect to the active coefficients,

$$\frac{dE}{da_m} = -b_m + \sum_n G_{m,n}a_n + \lambda\frac{dC(a_m)}{da_m} = -b_m + \sum_{n\neq m} G_{m,n}a_n + a_m + \lambda\frac{dC(a_m)}{da_m}, \tag{11}$$

where we assume the vectors are unit-norm $\left|\left|\boldsymbol{\phi}_m\right|\right|^2 = 1$. Looking back to the dynamic system in (4),

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

we can see that the dynamics on the internal state variables are proportional to the derivative of the energy function in (11), $\dot{u}_m \propto -\frac{dE}{da_m}$, if the active coefficients are related to the internal state variables by

$$u_m = a_m + \lambda\frac{dC(a_m)}{da_m}.$$

B Cost functions corresponding to ideal thresholding functions

The sigmoidal threshold function specified in (7) is invertible, meaning that active coefficients can be related back to their underlying state variables, $u_m = T^{-1}_{(\alpha,\gamma,\lambda)}(a_m)$, though not in closed form. For notational simplicity and without losing generality, we will assume in this section positive coefficients ($a_m > 0$). Though the ideal thresholding functions are not technically invertible, we can find the limit of the inverse function:

$$T^{-1}_{(\alpha,\infty,\lambda)}(a_m) = \lim_{\gamma\to\infty} T^{-1}_{(\alpha,\gamma,\lambda)}(a_m) = \begin{cases} \lambda & \text{if} \quad a_m < (1-\alpha)\lambda \\ \lambda + a_m - (1-\alpha)\lambda & \text{if} \quad a_m \geq (1-\alpha)\lambda. \end{cases}$$

Using the correspondence from Appendix A,

$$\lambda\frac{dC(a_m)}{da_m} = u_m - a_m = T^{-1}_{(\alpha,\gamma,\lambda)}(a_m) - a_m,$$

we integrate to find the ideal cost function

$$\begin{aligned} C(a_m) &= \frac{1}{\lambda}\int_0^{a_m}\left(T^{-1}_{(\alpha,\infty,\lambda)}(x) - x\right)dx \\ &= \frac{1}{\lambda}\left(\int_0^{a_m}(\lambda - x)\,dx + \int_{(1-\alpha)\lambda}^{a_m}(x - (1-\alpha)\lambda)\,dx\right) \\ &= \alpha a_m + \frac{\lambda\,(1-\alpha)^2}{2}. \end{aligned}$$

C Stability of LCAs

C.1 Equilibrium points

For a given set of active and inactive coefficients the LCA system equations are linear and only change when a node crosses threshold (from above or below). A sub-field of control theory specifically addresses these *switched systems* [42]. To express the HLCA system as a switched system, we define $\mathcal{M} \subseteq [1, \ldots, M]$ as the current set of active nodes (i.e., $m \in \mathcal{M}$ if $|u_m(t)| \geq \lambda$). We also define a $(M \times M)$ selection matrix $S_{\mathcal{M}}$ as being all zeros except for ones on the diagonal corresponding to the active nodes,

$$[S_{\mathcal{M}}]_{m,n} = \begin{cases} 1 & \text{if } m = n \text{ and } m \in \mathcal{M} \\ 0 & \text{if } m \neq n \text{ or } m \notin \mathcal{M}. \end{cases}$$

Defining the system matrix $A_{\mathcal{M}} = \frac{1}{\tau}\left[\left(I - \Phi^t\Phi\right) S_{\mathcal{M}} - I\right]$, the HLCA is written as a *switched linear system*,[3]

$$\dot{\boldsymbol{u}}\left(t\right) = \frac{1}{\tau}\Phi^t \boldsymbol{s}(t) + A_{\mathcal{M}}\boldsymbol{u}(t)\,.$$

There are only finitely many possible sets $\mathcal{M}$ which we further limit by only allowing sets satisfying the stability criteria (active nodes must not form linearly dependent subdictionaries). We also assume that a given fixed input $\boldsymbol{s}$ induces equilibrium points $\boldsymbol{u}^*$ that do not have any components identically equal to threshold $u^*_m \neq \lambda$. This condition appears true with overwhelming probability, and implies that there exists $r > 0$ such that $|u^*_m| - r \geq \lambda$ for all $m \in \mathcal{M}$ and $|u^*_m| + r \leq \lambda$ for all $m \notin \mathcal{M}$.

For a given $\mathcal{M}$, linear systems theory indicates that the system

$$\dot{\boldsymbol{u}} = A_{\mathcal{M}}\boldsymbol{u}$$

has a single equilibrium point (i.e., is *asymptotically stable*) only if $A_{\mathcal{M}}$ has negative eigenvalues [34]. The matrix $A_{\mathcal{M}}$ has no positive eigenvalues, so we must show that it is full rank ($A_{\mathcal{M}}\boldsymbol{u} \neq 0, \forall \boldsymbol{u} \in \mathbb{R}^M$). We begin by determining the nullspace $\mathcal{N}\left(\cdot\right)$ of the composite matrix $\Phi^t\Phi S_{\mathcal{M}}$. The nullspace of Φ^t is empty, $\mathcal{N}\left(\Phi^t\right) = \emptyset$, because $\text{span}\{\boldsymbol{\phi}_m\} = \mathbb{R}^N$. Because the collection $\{\boldsymbol{\phi}_m\}_{m\in\mathcal{M}}$ is linearly independent and the matrix $\Phi S_{\mathcal{M}}$ consists of only those selected vectors on the columns, $\mathcal{N}\left(\Phi S_{\mathcal{M}}\right) = \text{span}\{e_m\}_{m\notin\mathcal{M}}$, where e_m are the canonical basis elements. Therefore the composite matrix also has a nullspace of $\mathcal{N}\left(\Phi^t\Phi S_{\mathcal{M}}\right) = \text{span}\{e_m\}_{m\notin\mathcal{M}}$. Without losing generality, we assume that the first $|\mathcal{M}|$ entries are active, $\mathcal{M} = 1, \ldots, |\mathcal{M}|$. Consider first the case when all non-trivial internal state vectors only have non-zero values in the first $|\mathcal{M}|$ positions, $\boldsymbol{u} \in \text{span}\{e_1, \ldots, e_{|\mathcal{M}|}\}$. In this case, $\boldsymbol{u} \notin \mathcal{N}\left(\Phi^t\Phi S_{\mathcal{M}}\right)$, implying that $A_{\mathcal{M}}\boldsymbol{u} = -\Phi^t\Phi S\boldsymbol{u} \neq 0$. Consider next the case when all non-trivial internal state vectors only have non-zero values in the last $(M - |\mathcal{M}|)$ positions, $\boldsymbol{u} \in \text{span}\{e_{|\mathcal{M}|+1}, \ldots, e_M\}$. In this case, $\boldsymbol{u} \in \mathcal{N}\left(\Phi^t\Phi S_{\mathcal{M}}\right)$, meaning that $A_{\mathcal{M}}\boldsymbol{u} = -\boldsymbol{u} \neq 0$. Taking these two cases together, we see that $A_{\mathcal{M}}\boldsymbol{u} \neq 0, \forall \boldsymbol{u} \in \mathbb{R}^M$, implying that $A_{\mathcal{M}}$ only has negative eigenvalues so that the system in question has a single equilibrium point.

Given a particular set of active nodes $\mathcal{M}$, we therefore have a single equilibrium point $\boldsymbol{u}^*$ defined by the system matrix $A_{\mathcal{M}}$. All other points within a neighborhood of this equilibrium point correspond to the same set of active nodes (and therefore the same system matrix). Therefore, since each system matrix has a single equilibrium, there can be no other equilibrium points with coordinates within a neighborhood of $\boldsymbol{u}^*$,

$$|u^*_m - u_m| < r \quad \text{for any} \quad m \implies f(\boldsymbol{u}) \neq 0.$$

We know then that there are a finite number of equilibrium points, and each equilibrium point is isolated because there can be no other equilibrium points infinitely close.

Finally we consider the stability of the system in the neighborhood of the equilibrium point $\boldsymbol{u}^*$. Because we know that the linear subsystem is asymptotically stable, we must show that there exists a $\epsilon > 0$ such

[3] All LCA systems can be written as a similar switched system, but the thresholding functions with additive correction terms require a cumbersome definition of proxy state variables that we omit here.

that for any $\boldsymbol{u}(0) \in B_\epsilon(\boldsymbol{u}^*)$, the set of active nodes $\mathcal{M}$ never changes so $A_\mathcal{M}$ stays fixed. We must therefore ensure that we can specify a $\epsilon > 0$ such that for a fixed $A_\mathcal{M}$ the internal states never change state, $|u_m(t)| > \lambda, \forall m \in \mathcal{M}$ and $|u_m(t)| < \lambda, \forall m \notin \mathcal{M}$. The tools of linear systems theory give this evolution [34]:

$$\begin{aligned}
\boldsymbol{u}(t) &= e^{A_\mathcal{M} t}\boldsymbol{u}(0) + \int_0^t e^{(t-\tau)A_\mathcal{M}}\Phi^t \boldsymbol{s} d\tau \\
&= e^{A_\mathcal{M} t}\boldsymbol{u}(0) + e^{A_\mathcal{M} t}\left(\int_0^t e^{-A_\mathcal{M}\tau} d\tau\right)\Phi^t \boldsymbol{s} \\
&= e^{A_\mathcal{M} t}\boldsymbol{u}(0) + e^{A_\mathcal{M} t}\left(-A^{-1}e^{-A_\mathcal{M} t} + A^{-1}\right)\Phi^t \boldsymbol{s} \\
&= e^{A_\mathcal{M} t}\boldsymbol{u}(0) + -A^{-1}\Phi^t \boldsymbol{s} + e^{A_\mathcal{M} t}A^{-1}\Phi^t \boldsymbol{s} \\
&= e^{A_\mathcal{M} t}\left(\boldsymbol{u}(0) - \boldsymbol{u}^*\right) + \boldsymbol{u}^*,
\end{aligned}$$

where $\lim \boldsymbol{u}(t) = -A^{-1}\Phi^t \boldsymbol{s} = \boldsymbol{u}^*$ for a linear system. From this, we bound the energy of the difference signal

$$||\boldsymbol{u}(t) - \boldsymbol{u}^*|| = \left|\left|e^{A_\mathcal{M} t}\left(\boldsymbol{u}(0) - \boldsymbol{u}^*\right)\right|\right| \leq e^{\mu_{\max} t}\,||\boldsymbol{u}(0) - \boldsymbol{u}^*|| \leq ||\boldsymbol{u}(0) - \boldsymbol{u}^*|| \leq \epsilon,$$

where $\mu_{\max}$ is the largest magnitude eigenvector of $A_\mathcal{M}$. This energy bound also serves as a crude bound on the individual elements of the internal state vector

$$|u_m(0) - u_m^*| \leq ||\boldsymbol{u}(t) - \boldsymbol{u}^*|| \leq \epsilon.$$

We conclude that if $\epsilon < r$, the system will not change state and behaves as a fixed, asymptotically stable linear system. Therefore, the system is locally asymptotically stable around each equilibrium point, $\boldsymbol{u}(0) \in B_r(\boldsymbol{u}^*)$.

C.2 Input-output stability

Consider the time derivative of the energy computed through the chain rule,

$$\frac{d}{dt}E(t) = \frac{dE}{d\boldsymbol{a}}\frac{d\boldsymbol{a}}{d\boldsymbol{u}}\dot{\boldsymbol{u}} = -\left(\frac{dE}{d\boldsymbol{a}}\right)^2\frac{d\boldsymbol{a}}{d\boldsymbol{u}},$$

where the last equality follows from the definition of the LCA dynamics $\dot{\boldsymbol{u}} = -\frac{dE}{d\boldsymbol{a}}$. Therefore, as long as $T_\lambda(\cdot)$ is non-decreasing, $\frac{d\boldsymbol{a}}{d\boldsymbol{u}} \geq 0$, implying that the energy function will be non-increasing with time $\frac{d}{dt}E(t) \leq 0$.

To assess input-output stability, define τ_D to be the average dwell time between changes to the set of active nodes $\mathcal{M}$. For switched linear systems, sufficient conditions for input-output stability require each subsystem to be asymptotically stable and that the system doesn't switch "too often" [43]. Specifically, we restate here a theorem from switched system theory in language corresponding to the LCAs.

Average dwell time theorem (Theorem 2 combined with Lemma 1 in [43]). *Given a collection of system matrices $A_\mathcal{M}$ and a positive constant ω_0 such that $A_\mathcal{M} + \omega_0 I$ is asymptotically stable for all t, then, for any $\omega \in [0, \omega_0]$, there is a finite constant τ_D^* such that as long as $\tau_D \geq \tau_D^*$, the switched system has a bounded internal state for piecewise constant input signals $\boldsymbol{s}(t)$:*

$$\left(\int_0^t e^{2\omega\tau}\,||\boldsymbol{u}(\tau)||^2\,d\tau\right)^{1/2} \leq \kappa_1\left(\int_0^t e^{2\omega\tau}\,||\boldsymbol{s}(\tau)||^2\,d\tau\right)^{1/2} + \kappa_2\,||\boldsymbol{s}(0)||\,,$$

where κ_1 and κ_2 are finite constants. Similar statements can be made using ℓ^∞ norms instead of ℓ^2 norms.

The average dwell time theorem guarantees that the LCA system will remain stable as long as each subsystem is asymptotically stable and τ_D is not too small. Appendix C.1 shows that the system matrix $A_\mathcal{M}$ has only strictly negative eigenvalues. The modified system matrix $\widetilde{A} = A_\mathcal{M} + \omega_0 I$ has a minimum magnitude eigenvalue of $\widetilde{\mu}_{\min} = \mu_{\min} + \omega_0$. Clearly there exists a $\omega_0 > 0$ such that $\widetilde{\mu}_{\min} > 0$ if and only if $\mu_{\min} > 0$. In other words, there exists $\omega_0 > 0$ so that $\widetilde{A}$ is asymptotically stable (thus satisfying the average dwell time theorem) if the stability criteria are met for every subsystem of the switched system.

D Steady-state sparsity of LCA systems

The LCA at steady-state looks like a fixed linear system. If we know *a priori* the set of active nodes corresponding to the steady-state response $\mathcal{M}$ then the steady-state internal state variables are given by

$$\widetilde{\boldsymbol{u}} = \lim_{t\to\infty} \boldsymbol{u}(t) = -\frac{1}{\tau} A_{\mathcal{M}}^{-1} \Phi^t \boldsymbol{s},$$

where $A_{\mathcal{M}}$ is defined as in Appendix C. While we cannot determine the set of active nodes in the limit, we can distinguish sets of nodes that *cannot* be active. When calculating the steady-state values $\widetilde{\boldsymbol{u}}$ assuming a fixed $\mathcal{M}$, if a node not in $\mathcal{M}$ is above threshold in $\widetilde{\boldsymbol{u}}$ (or a node in $\mathcal{M}$ is below threshold), the system matrix would have changed. In this case we call $\mathcal{M}$ *inconsistent*. It is important to note a subtle point: finding an inconsistency does not indicate the correct steady-state active set, but only indicates that it cannot be $\mathcal{M}$.

Given a set of candidate active nodes $\mathcal{M}$, we assume (without losing generality) the active nodes are indexed consecutively from the beginning, $\mathcal{M} = 1, \ldots, |\mathcal{M}|$. We employ the usual canonical basis elements $e_m \in \mathbb{R}^M$ that contain a single non-zero entry in the m^{th} position (e.g., $e_1 = [1, 0, \ldots, 0]^t$). We will also employ what we call the *Grammian basis elements* $v_m \in \mathbb{R}^M$ that contain the inner products of one dictionary element with all the others, $v_m = [\langle \boldsymbol{\phi}_1, \boldsymbol{\phi}_m \rangle, \langle \boldsymbol{\phi}_2, \boldsymbol{\phi}_m \rangle, \ldots, \langle \boldsymbol{\phi}_M, \boldsymbol{\phi}_m \rangle]^t = [G_{1,m}, G_{2,m}, \ldots, G_{M,m}]^t$. The system matrix can be expressed entirely in terms of these basis elements,

$$A_{\mathcal{M}} = \frac{1}{\tau}\left[\left(I - \Phi^t\Phi\right) S_{\mathcal{M}} - I\right] = -\frac{1}{\tau}\left[v_1, \ldots, v_{|\mathcal{M}|}, e_{|\mathcal{M}|+1}, \ldots, e_M\right],$$

where $S_{\mathcal{M}}$ is the corresponding selection matrix (defined in Appendix C.1). The inverse of this system matrix has several important properties. First, for inactive nodes $m \notin \mathcal{M}$, the corresponding canonical basis vector is an eigenvector of the inverse system matrix $A^{-1}e_m = -\tau e_m$. Similarly, for active nodes $m \in \mathcal{M}$, the inverse system matrix transforms the corresponding Grammian basis vector into the canonical basis vector, $A^{-1}v_m = -\tau e_m$. We also note that the set $\left(\{v_m\}_{m=1}^{|\mathcal{M}|} \bigcup \{e_m\}_{m=|\mathcal{M}|+1}^{M}\right)$ is a basis for the space $\mathbb{R}^M$.

For now, let the input signal be proportional to a single dictionary element, $\boldsymbol{s} = \alpha\boldsymbol{\phi}_n$, meaning that $\Phi^t\boldsymbol{s} = \alpha v_n$. We will assume that the scaling coefficient is greater than the chosen threshold, $\alpha > \lambda$, so the signal strength is considered significant. There exists a unique set of coefficients $\{\beta_m\}$ such that

$$\alpha v_n = \beta_1 v_1 + \cdots + \beta_{|\mathcal{M}|} v_{|\mathcal{M}|} + \beta_{|\mathcal{M}|+1} e_{|\mathcal{M}|+1} + \cdots + \beta_M e_M.$$

Looking at each element of this expression in turn is illuminating:

$$\begin{aligned}
\alpha\,[v_n]_1 &= \langle \boldsymbol{\phi}_1, \alpha\boldsymbol{\phi}_n \rangle &&= \langle \boldsymbol{\phi}_1, \left(\beta_1\boldsymbol{\phi}_1 + \cdots + \beta_{|\mathcal{M}|}\boldsymbol{\phi}_{|\mathcal{M}|}\right)\rangle \\
&&&\vdots \\
\alpha\,[v_n]_{|\mathcal{M}|} &= \langle \boldsymbol{\phi}_{|\mathcal{M}|}, \alpha\boldsymbol{\phi}_n \rangle &&= \langle \boldsymbol{\phi}_{|\mathcal{M}|}, \left(\beta_1\boldsymbol{\phi}_1 + \cdots + \beta_{|\mathcal{M}|}\boldsymbol{\phi}_{|\mathcal{M}|}\right)\rangle \\
\alpha\,[v_n]_{|\mathcal{M}|+1} &= \langle \boldsymbol{\phi}_{|\mathcal{M}|+1}, \alpha\boldsymbol{\phi}_n \rangle &&= \langle \boldsymbol{\phi}_{|\mathcal{M}|+1}, \left(\beta_1\boldsymbol{\phi}_1 + \cdots + \beta_{|\mathcal{M}|}\boldsymbol{\phi}_{|\mathcal{M}|} + \beta_{|\mathcal{M}|+1}\boldsymbol{\phi}_{|\mathcal{M}|+1}\right)\rangle \\
&&&\vdots \\
\alpha\,[v_n]_M &= \langle \boldsymbol{\phi}_M, \alpha\boldsymbol{\phi}_n \rangle &&= \langle \boldsymbol{\phi}_M, \left(\beta_1\boldsymbol{\phi}_1 + \cdots + \beta_{|\mathcal{M}|}\boldsymbol{\phi}_{|\mathcal{M}|} + \beta_M\boldsymbol{\phi}_M\right)\rangle.
\end{aligned}$$

The coefficients $\beta_1, \ldots, \beta_{|\mathcal{M}|}$ correspond to the best approximation of $\boldsymbol{s}$ in the subspace spanned by $\{\boldsymbol{\phi}_m\}_{m=1}^{|\mathcal{M}|}$.

Consider first the case when $n \in \mathcal{M}$. The coefficients are optimal: $\beta_n = 1$ and $\beta_m = 0$ for all $m \neq n$. Assuming the fixed system matrix $A_{\mathcal{M}}$, the steady-state internal state variables are given by

$$\widetilde{\boldsymbol{u}} = -\frac{1}{\tau} A_{\mathcal{M}}^{-1} \Phi^t \boldsymbol{s} = -\frac{1}{\tau} A_{\mathcal{M}}^{-1} \alpha v_n = \alpha e_n.$$

If the LCA selects the optimal set of nodes, the coefficient values are optimal. Now consider the case when the vector is not part of the active set, $n \notin \mathcal{M}$. The coefficients $\{\beta_m\}$ correspond to the steady-state values:

$$-\frac{1}{\tau} A_{\mathcal{M}}^{-1} \Phi^t \boldsymbol{s} = \beta_1 e_1 + \cdots + \beta_{|\mathcal{M}|} e_{|\mathcal{M}|} + \beta_{|\mathcal{M}|+1} e_{|\mathcal{M}|+1} + \cdots + \beta_M e_M.$$

Looking at the entries of αv_n, each index not in the active set, $m \notin \mathcal{M}$, has the coefficient

$$\beta_m = \alpha \langle \boldsymbol{\phi}_m, \boldsymbol{\phi}_n \rangle - \left(\beta_1 \langle \boldsymbol{\phi}_1, \boldsymbol{\phi}_{|\mathcal{M}|+1} \rangle + \cdots + \beta_{|\mathcal{M}|} \langle \boldsymbol{\phi}_{|\mathcal{M}|}, \boldsymbol{\phi}_{|\mathcal{M}|+1} \rangle \right).$$

The set $\mathcal{M}$ is consistent only if $\beta_m > \lambda$ for all $m \in \mathcal{M}$, and $\beta_m > \lambda$ for all $m \notin \mathcal{M}$.

These results lead us to several observations that help qualify the sparsity of the LCA solutions:

1. When $n \in \mathcal{M}$, $\widetilde{u}_m = 0 < \lambda$ for all $m \neq n$ means that if the LCA finds the optimal node, it will not include any extraneous nodes.

2. When $n \in \mathcal{M}$, $\widetilde{u}_n = \alpha$ means that if the optimal node is correctly selected by the LCA in the steady state, the system will find the optimal coefficient for that node.

3. When $n \notin \mathcal{M}$, $\beta_m > \lambda$ for all $m \in \mathcal{M}$ and $\beta_m < \lambda$ for all $m \notin \mathcal{M}$ means that the input signal can be represented by dictionary elements $\{\boldsymbol{\phi}_m\}_{m=1}^{|\mathcal{M}|}$ so the residual projection onto any other vector is less than λ. Any set of active nodes that cannot represent the input signal to this accuracy is inconsistent.

To minimize notation, we have only discussed one-sparse input signals. However, the analysis performed here is entirely linear and the same principles apply to input signals containing more than one dictionary component. In particular, a set of active nodes is inconsistent if: it cannot represent every component of the input signal so that the residual projection onto every other dictionary element is less than λ; or it contains every component of the input signal in addition to other extraneous components. Also, if the active set recovers the correct indices, the LCA steady-state coefficients will find the optimal coefficients.

References

[1] E. Candès and D. Donoho. New tight frames of curvelets and optimal representations of objects with piecewise C^2 singularities. *Communications on Pure and Applied Mathematics*, 57(2):219–266, 2004.

[2] W.E. Vinje and J.L. Gallant. Natural stimulation of the nonclassical receptive field increases information transmission efficiency in V1. *Journal of Neuroscience*, 22:2904–2915, 2002.

[3] M.S. Lewicki. Efficient coding of natural sounds. *Nature Neuroscience*, 5:356–363, 2002.

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

[5] B. Olshausen and D. Field. Emergence of simple cell receptive properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996.

[6] B. Delgutte, B.M. Hammond, and P.A. Cariani. *Psychophysical and Physiological Advances in Hearing*, chapter Neural coding of the temporal envelope of speech: Relation to modulation transfer functions, pages 595–603. Whurr Publishers, Ltd., 1998.

[7] J. Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Trans. Info. Th.*, 50(10):2231–2242, 2004.

[8] B.K. Natarajan. Sparse approximate solutions to linear systems. *SIAM J. Comp.*, 24(2):227–234, April 1995.

[9] S.S. Chen, D.L. Donoho, and M.A. Saunders. Atomic decomposition by basis pursuit. *J. Sci. Comp.*, 43(1):129–159, 2001.

[10] D.L. Donoho and M. Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ^1 minimization. *Proceedings of the National Academy of Sciences of the United States of America*, 100(5):2197–2202, March 2003.

[11] S. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Trans. Sig. Proc.*, 41(12):3397–3415, December 1993.

[12] D.H. Hubel and T.N. Wiesel. Receptive fields and functional architecture of monkey striate cortex. *Journal of Physiology*, 195:215–243, 1968.

[13] R.L. De Valois and K.K. De Valois. *Spatial Vision*. Oxford University Press, New York, 1988.

[14] L. Perrinet, M. Samuelides, and S. Thorpe. Sparse spike coding in an asynchronous feed-forward multi-layer neural network using matching pursuit. *Neurocomputing*, 57:125–134, 2004.

[15] L. Perrinet, M. Samuelides, and S. Thorpe. Coding static natural images using spiking event times: Do neurons cooperate? *IEEE Transactions on neural networks*, 15(5):1164–1175, September 2004.

[16] J.A. Feldman and D.H. Ballard. Connectionist models and their properties. *Cognitive Science: A Multidisciplinary Journal*, 6(3):205–254, 1982.

[17] D.P. Wipf and B.D. Rao. Sparse Bayesian learning for basis selection. *IEEE Transactions on Signal Processing*, 52(8):2153–2164, August 2004.

[18] M. Tipping. Sparse bayesian learning and the relevance vector machine. *The Journal of Machine Learning Research*, 1:211–244, 2001.

[19] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 39(1):1–38, 1977.

[20] D. Donoho, Y. Tsaig, I. Drori, and J. Starck. Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit. 2006.

[21] A. Pece and N. Petkov. Fast atomic decomposition by the inhibition method. In *In Proceedings of the 15th International Conference on Pattern Recognition*, 2000.

[22] H. Feichtinger, A. Türk, and T Strohmer. Hierarchical parallel matching pursuit. In *Proceedings of SPIE*, volume 2302, pages 222–232, San Diego, CA, July 1994.

[23] G. Davis, S. Mallat, and Z. Zhang. Adaptive time-frequency decompositions with matching pursuit. *Opt. Eng.*, 33(7), July 1994.

[24] L. Rebollo-Neira and D. Lowe. Optimized orthogonal matching pursuit approach. *IEEE Signal Processing Letters*, 9(4):137–140, April 2002.

[25] S. Fischer, G. Cristóbal, and R. Redondo. Sparse overcomplete Gabor wavelet representation based on local competitions. *IEEE Transactions on Image Processing*, 15(2):265–272, 2004.

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

[27] K. Herrity, A.C. Gilbert, and J. Tropp. Sparse approximation via iterative thresholding. In *In Proceedings of the 2006 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Toulouse, France, May 2006.

[28] 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.

[29] J.J. Hopfield. Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences of the United States of America*, 81(10):3088–3092, May 1984.

[30] D.L. Donoho. Denoising by soft-thresholding. *IEEE Trans. Info. Th.*, 41(3):613–627, May 1995.

[31] M. Elad. Why simple shrinkage is still relevant for redundant representations? *IEEE Trans. Info. Th.*, 52(12):5559–5569, 2006.

[32] R.A. DeVore and V.N. Temlyakov. Some remarks on greedy algorithms. *Advances in Computational Mathematics*, 5:173–187, 1996.

[33] O. Christensen. *An Introduction to Frames and Riesz Bases*. Birkhauser, Boston, MA, 2002.

[34] G.F. Franklin, J.D. Powell, and A. Emami-Naeini. *Feedback Control of Dynamic Systems*. Addison-Wesley Publishing Company, Reading, MA, 1986.

[35] H.K. Khalil. *Nonlinear Systems*. Prentice Hall, Upper Saddle Tiver, NJ, third edition, 2002.

[36] J.A. Tropp. Random subdictionaries of general dictionaries. Submitted manuscript, 2006.

[37] D.L. Donoho. Neighborly polytopes and sparse solutions of underdetermined linear equations. Preprint., 2005.

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

[39] M.A. Cohen and S. Grossberg. Absolute stability of global pattern formation and parallel memory storage by competitive neural networks. *IEEE Transactions on Systems, Man, and Cybernetics*, 13(5):815–825, September–October 1983.

[40] H. Yang and T.S. Dillon. Exponential stability and oscillation of Hopfield graded response neural network. *IEEE Transactions on Neural Networks*, 5(5):719–729, September 1994.

[41] J.H. Li, A.N. Michel, and W. Porod. Qualitative analysis and synthesis of a class of neural networks. *IEEE Transactions on Circuits and Systems*, 35(8):976–986, August 1988.

[42] R.A. Decarlo, M.S. Cranicky, S. Pettersson, and B. Lennartson. Perspectives and results on the stability and stabilizability of hybrid systems. *Proceedings of the IEEE*, 88(7):1069–1082, 2000.

[43] J.P. Hespanha and A.S. Morse. Stability of switched systems with average dwell time. In *Proceedings of the 38th Conference on Decision and Control*, December 1999.

[44] T. Berger. *Rate Distortion Theory*. Prentice Hall, Englewood Cliffs, NJ, 1971.

[45] P. Dayan and A.F. Abbott. *Theoretical Neuroscience*. MIT Press, Cambridge, MA, 2001.

[46] 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.

[47] J.R. Norris. *Markov Chains*. Cambridge University Press, New York, 1997.

[48] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. John Wiley & Sons, Inc., New York, NY, 1991.