FOCUSS-Based Dictionary Learning Algorithms

Kenneth Kreutz-Delgado and Bhaskar D. Rao *Electrical and Computer Engineering
Jacobs School of Engineering
University of California, San Diego
La Jolla, California 92093-0407, USA

February 24, 2002

Abstract

Algorithms for data-driven learning of domain-specific overcomplete dictionaries are developed to obtain maximum likelihood and maximum a posteriori dictionary estimates based on the use of Bayesian models with concave/Schur-concave (CSC) negative log-priors. Such priors are appropriate for obtaining sparse representations of environmental signals within an appropriately chosen (environmentally matched) dictionary. The elements of the dictionary can be interpreted as 'concepts,' 'features' or 'words' capable of succinct expression of events encountered in the environment (the source of the measured signals). This is a generalization of vector quantization in that one is interested in a description involving a few dictionary entries (the proverbial '25 words or less'), but not necessarily as succinct as one entry. To learn an environmentally-adapted dictionary capable of concise expression of signals generated by the environment, we develop algorithms that iterate between a representative set of sparse representations found by variants of FOCUSS, an affine scaling transformation (AST)-like sparse signal representation algorithm recently developed at UCSD, and an update of the dictionary using these sparse representations.

1 INTRODUCTION

FOCUSS[1, 2, 3, 4, 5, 6, 7] stands for "FOCal Underdetermined System Solver" and is an algorithm designed to obtain suboptimally (and, at times, maximally[1]) sparse solutions to the $m \times n$, underdetermined linear inverse problem[2]

$$Ax = y\,. \tag{1}$$

Since our initial investigations into its properties as a algorithm for providing sparse solutions to linear inverse problems in relatively noise-free environments,[1, 2, 3, 4, 5, 6] we now better understand the behavior of FOCUSS in noisy environments[9, 10] and as an interior point-like optimization algorithm for optimizing concave functionals subject to linear constraints.[11, 12, 13, 14, 15, 16, 17] In this paper, we will briefly discuss how the use of concave (and Schur concave) functionals enforces sparse solutions to (1). We also discuss the choice of the matrix, A, in (1) and its relationship to the set of signal vectors y for which we hope to obtain sparse representations. In this vein, we present algorithms capable of *learning* an environmentally adapted dictionary, A, given a sufficiently large and statistically representative sample of signal vectors, y.[18, 19, 20]

*Authors can be contacted at {kreutz,brao}@ece.ucsd.edu. Research supported by National Science Foundation Grant No. CCR-9902961

[1] A measure of *sparseness* (equivalently, *succinctness* or *concentration*) is provided by the so-called *numerosity*[8],

$$\#(x) = \text{number of nonzero elements of } x.$$

A maximally sparse solution is one for which the sparsity is a minimum as measured by the numerosity.

[2] For notational simplicity, in this paper we consider the real case only. The extension to the complex case is straightforward.

We refer to the columns of the full row-rank $m \times n$ matrix A,

$$A = [a_1, \cdots, a_n] \in \mathsf{R}^{m \times n}, \quad n \gg m\,, \tag{2}$$

as a *dictionary* and they are assumed to be a set of vectors capable of providing a highly succinct representation for *most* (and, ideally, all) statistically representative signal vectors $y \in \mathsf{R}^m$. Note, that with the assumption that rank$(A) = m$, every vector y has a representation; the question at hand is whether this representation is likely to be sparse. We call the statistical generating mechanism for signals, y, the *environment* and a dictionary, A, within which such signals can be sparsely represented an *environmentally adapted* dictionary.

Environmentally generated signals typically have significant statistical structure, and can be represented by a set of basis vectors spanning a lower dimensional submanifold of meaningful signals [21, 22]. These environmentally-meaningful representation vectors can be obtained by maximizing the mutual information between the set of these vectors (the dictionary) and the signals generated by the environment [23, 24, 25, 26, 27, 28]. This procedure can be viewed as a natural generalization of Independent Component Analysis (ICA) [23, 25]. As initially developed, this procedure usually results in obtaining a *minimal* spanning set of spanning vectors (i.e., a true basis). More recently, the desirability of obtaining "overcomplete" sets of vectors (or "dictionaries") has been noted [26, 29, 30, 31, 8, 6]. For example, projecting measured noisy signals onto the signal submanifold spanned by a set of dictionary vectors results in noise reduction and data compression [8, 32]. These dictionaries can be structured as a set of true bases from which a single basis is to be selected to represent the measured signal(s) of interest [30], or as a large set of individual vectors [31], perhaps even unorganized in any particular way [26, 29, 6].

The problem of determining a representation from an overcomplete dictionary, $A = [a_1, ..., a_n]$, $n >> m$, for a specific signal measurement, y, is equivalent to solving an underdetermined inverse problem, $Ax \approx y$. The standard least squares solution to this problem has the (at times) undesirable feature of involving *all* the dictionary vectors in the solution[3] (the "spurious artifact" problem), and does not allow for the extraction of a categorically or physically meaningful solution. That is, it is not generally the case that a least-squares solution yields a concise representation allowing for a precise semantic meaning[4]. If the dictionary is large and rich enough in representational power, a measured signal can be matched to a very few (perhaps even just one) dictionary words. In this manner we can obtain concise semantic content about objects or situations encountered in natural environments [21]. Thus, there has been a significant interest in finding "sparse" solutions, x, (solutions having a minimum number of nonzero elements) to the signal representation problem. Interestingly, matching a *specific* signal to a sparse set of dictionary words/vectors can be related to entropy *minimization* as a means of elucidating statistical structure [33]. Finding a sparse representation (based on the use of a "few" code/dictionary words) can also be viewed as a generalization of vector quantization where a match to a single "code vector" (word) is always sought (taking "code book" = "dictionary"). Indeed, we can refer to a sparse solution, x, as a sparse coding of the signal instantiation, y.

1.1 Stochastic Models

It is well known[34] that the stochastic generative model

$$y = Ax + \nu\,, \tag{3}$$

can be used to develop algorithms enabling coding of $y \in \mathsf{R}^m$ via solving the inverse problem for a sparse solution $x \in \mathsf{R}^n$ for the undercomplete ($n < m$) and complete ($n = m$) cases. In recent years there has been a great deal of interest in obtaining sparse codings of y via this procedure for the *overcomplete* ($n > m$) case [31, 21]. In our earlier work we have shown that given an overcomplete dictionary, A, (with the columns of A comprising the dictionary

[3] This fact comes as no surprise when the solution is interpreted within a Bayesian framework, using a gaussian (maximum entropy) prior.

[4] Taking "semantic" here to mean categorically or physically interpretable.

vectors) a MAP estimate of the source vector, x, will yield a sparse coding of y in the low-noise limit if the negative log-prior, $-\log(P(x))$, is Concave/Schur-Concave (CSC)[7, 16]. For $P(x)$ factorizable into a product of marginal probabilities, the resulting code is also known to provide an Independent Component Analysis (ICA) representation of y. More generally, a CSC prior results in a sparse representation even in the non-factorizable case (with x then forming a "Dependent Component Analysis," or DCA, representation) [12, 17].

Given iid data, $Y = Y^N = (y_1, ..., y_N)$, which is assumed to be generated by the model (3), a maximum likelihood estimate, $\widehat{A}_{\text{ML}}$, of the unknown (but nonrandom) dictionary A can be determined as[26, 29]

$$\widehat{A}_{\text{ML}} = \arg\max_A P(Y; A)\,.$$

This requires integrating out the unobservable iid source vectors, $X = X^N = (x_1, ..., x_N)$, in order to compute $P(Y; A)$ from the (assumed) known probabilities $P(x)$ and $P(\nu)$. In essence X is formally treated as a set of nuisance parameters which, in principle, can be removed via integration. However, because the prior $P(x)$ is generally taken to be supergaussian, this integration is intractable or computationally unreasonable. Thus approximations to this integration are performed which result in an approximation to $P(Y; A)$ which is then maximized with respect to Y. A new, better, approximation to the integration can then be made and this process is iterated until the estimate of the dictionary A has (hopefully) converged [26]. We refer to the resulting estimate as an Approximate Maximum Likelihood (AML) estimate of the dictionary A (denoted here by $\widehat{A}_{\text{AML}}$). No formal proof of the convergence of this algorithm to the true maximum likelihood estimate, A_{ml}, has been given in the prior literature, but it appears to perform well in various test cases [26]. Below, we discuss the problem of dictionary learning within the framework of our recently developed CSC log-prior model-based sparse source vector learning approach which for a *known* overcomplete dictionary can be used to obtain sparse codes, both for the ICA (factorial code) and DCA (nonfactorial code) cases [7, 12, 13, 14, 11, 16]. Such sparse codes can be found using FOCUSS, an affine scaling transformation (AST)-like iterative algorithm which finds a sparse locally optimal MAP estimate of the source vector x for an observation y. Using these results, we can develop dictionary learning algorithms, both within the Approximate Maximum Likelihood framework mentioned above and for obtaining a MAP-like estimate, $\widehat{A}_{\text{MAP}}$, of the (now assumed random) dictionary, A, assuming in the latter case that the dictionary belongs to a compact submanifold corresponding to unit Frobenius norm. Under certain conditions, convergence to a local minimum of a MAP-loss function which combines functions of the discrepancy $e = (y - Ax)$ and the degree of sparsity in x can be rigorously proved.

2 FOCUSS: Sparse Solutions for Known Dictionaries

2.1 Known Dictionary Model.

A Bayesian interpretation is obtained from the generative signal model (3) by assuming that x has the parameterized (generally nongaussian) pdf,

$$P_p(x) = Z_p^{-1} e^{-\gamma_p\, d_p(x)}\,, \quad Z_p = \int e^{-\gamma_p\, d_p(x)} dx\,, \tag{4}$$

with parameter vector p. Similarly, the noise ν is assumed to have a parameterized (possibly nongaussian) density $P_q(\nu)$ of the same form as (4) with parameter vector q. It is assume that x and ν have zero means and that their densities obey the property $d(x) = d(|x|)$, for $|\cdot|$ defined component-wise. This is equivalent to assuming that the densities are symmetric with respect to sign changes in the components of x, $x[i] \leftarrow -x[i]$, and therefore that the skews of these densities are zero. We also assume that $d(0) = 0$. With a slight abuse of notation, we allow the differing subscripts q and p to indicate that d_q and d_p may be *functionally* different as well as parametrically different. We refer to densities like (4), for suitable additional constraints on $d_p(x)$, as Hypergeneralized Gaussian Distributions [16, 17].

If we treat A, p, and q as *known* parameters, then x and y are jointly distributed as $P(x,y) = P(x,y;p,q,A)$. Bayes' rule yields,

$$P(x|y;p,A) = \frac{1}{\beta}P(y|x;p,A)\cdot P(x;p,A) = \frac{1}{\beta}P_q(y-Ax)\cdot P_p(x) \tag{5}$$

$$\beta = P(y) = P(y;p,q,A) = \int P(y|x)\cdot P_p(x)dx\,. \tag{6}$$

Usually the dependence on p and q is notationally suppressed and we write $\beta = P(y;A)$, etc. Given an observation, y, maximizing (5) with respect to x yields the MAP estimate $\widehat{x}$. This ideally results in a sparse coding of the observation, a requirement which places functional constraints on the probability density functions. Note that β is independent of x and can be ignored when optimizing (5) with respect to the unknown source vector x.

The MAP estimate equivalently is obtained from minimizing the the negative logarithm of $P(x|y)$, which is,

$$\widehat{x} = \arg\min_x d_q(y-Ax) + \lambda\, d_p(x)\,, \tag{7}$$

where $\lambda = \gamma_p/\gamma_q$, and $d_q(y-Ax) = d_q(Ax-y)$ by our assumption of symmetry. The quantity $\frac{1}{\lambda}$ is interpretable as a signal-to-noise ratio (SNR). Furthermore, assuming that $d_{\,q}$ and d_p are CSC, the term $d_q(y-Ax)$ in (7) encourages sparse residuals, $e = y - A\widehat{x}$, while the term $d_p(x)$ encourages sparse source-vector estimates, $\widehat{x}$. A given value of λ then determines a trade-off between residual and source vector sparseness.

Note that $\lambda \to 0$ as $\gamma_p \to 0$ which (consistent with the generative model (3)) we refer to as the *low noise limit*. Because the mapping A is assumed to be onto, in the low noise limit the optimization (7) is equivalent to the linearly constrained problem,

$$\widehat{x} = \arg\min d_p(x) \quad \text{subject to} \quad Ax = y\,. \tag{8}$$

In the low-noise limit, no sparseness constraint is placed on the residuals $e = y - A\widehat{x}$. It is evident that the structure of $d_p(\cdot)$ is critical for obtaining a sparse coding, $\widehat{x}$, of the observation y[12, 11]. The quantity $d_{\,p}(x)$ is always assumed to be CSC (enforcing sparse solutions to the inverse problem (3)). Later, during the development of dictionary learning algorithms, ν will be assumed to be Gaussian ($q = 2$).

2.2 The FOCUSS Algorithm.

Locally optimal solutions to the problems (8) and (7) are given by the FOCUSS algorithm. This is an Affine-Scaling Transformation (AST)-like (interior point) algorithm originally proposed for the low noise case[6, 12, 11] and extended via regularization to the non-trivial noise case[9, 20]. For a known dictionary estimate, A, a solution to the optimization problem is provided by repeated iteration of the form,

$$\widehat{x}_k \overset{\Pi^{-1}(\widehat{x}_k)}{\leftarrow} A^T\left(\beta(\widehat{x}_k)I + A\Pi^{-1}(\widehat{x}_k)A^T\right)^{-1} y_k\,, \tag{9}$$

$k = 1,\cdots,N$. The diagonal positive-definite matrix, $\Pi(x)$, is defined as in equation (28) given below and comes from a factorization of the gradient of the sparsity-inducing regularizing function $d_{\,p}(x)$. This factorization is key to understanding FOCUSS as a sparsity-inducing interior-point (AST-like) optimization algorithm[12, 13, 11]. Taking $\beta \equiv 0$ yields the original low-noise FOCUSS algorithm. The case $\beta \neq 0$ yields the regularized FOCUSS algorithm[7, 20]. More computationally robust variants of (9) are discussed elsewhere[4, 9].

For regularized FOCUSS, $\beta(\widehat{x}_k) = \lambda\alpha(x)$, where λ is the regularization parameter in (7), which in general is a function of x_k, y_k and the iteration number. Methods for choosing λ include the quality-of-fit criteria, the sparsity critera, and the *L-curve* [35]. The quality-of-fit criteria attempts to minimize the residual error $y - Ax$ [2] which can be shown to converge to a sparse solution [11]. The sparsity critera requires that a certain number of elements of each x_k be non-zero.

The L-curve method adjusts λ to optimize the trade-off between the residual and sparsity of x_k. The plot of $d_p(x_k)$ versus $d_q(y_k - Ax_k)$ has an L-shape, the corner of which provides the best trade-off. The corner of the L-curve is the point of maximum curvature, and can be found by a one-dimensional maximization of the curvature function [36].

A hybrid approach known as the *modified L-curve method* combines the L-curve method on a linear scale and the quality-of-fit critera, which is used to place limits on the range of λ that can be choosen by the L-curve [35]. The modified L-curve method was shown to have good performance, but it requires a one-dimensional numerical optimization step for each x_k at each iteration, which can be computationally expensive for large vectors.

2.3 Independent Component Analysis (ICA) and Sparsity Inducing Priors.

An important class of densities is given by the *generalized gaussians* for which

$$d_p(x) = \|x\|_p^p = \sum_{k=1}^{n} |x[k]|^p \,, \tag{10}$$

for $p > 0$ [37]. This is a special case of the larger ℓ_p class (the "p-class") of functions which allows p to be negative in value[11, 12]. Note that this function has the special property of *separability*,

$$d_p(x) = \sum_{k=1}^{n} d_p(x[k]) \,,$$

which corresponds to *factorizability* of the density $P_p(x)$,

$$P_p(x) = \prod_{k=1}^{n} P_p(x[k]) \,,$$

and hence to *independence of the components of* x. It is now well-known that the assumption of independent components allows the problem of solving the generative model (3) for x to be interpreted as an Independent Component Analysis (ICA) problem [23, 38, 26, 39]. It is of interest, then, to consider the development of a large class of parameterizable separable functions $d_p(x)$ consistent with the ICA assumption[11, 12]. Note that given such a class, it is natural to examine the issue of finding a best fit within this class to the "true" underlying prior density of x. This is a problem of parametric density estimation of the true prior where one attempts to find an optimal choice of the model density $P_p(x)$ by an optimization over the parameters p which define the choice of a prior from within the class. This is, in general, a difficult problem which may require the use of Monte-Carlo, evolutionary programming, and/or stochastic search techniques.

Can the belief that supergaussian priors, $P_p(x)$, are appropriate for finding sparse solutions to (3)[21, 26] be clarified or made rigorous? It is well known that the generalized gaussian distribution arising from the use of (10) yields supergaussian distributions (positive kurtosis) for $p < 2$ and subgaussian (negative kurtosis) for $p > 2$. However, one can argue that the condition for obtaining sparse solutions in the low noise limit is the stronger requirement that $p \leq 1$, in which case the separable function $d_p(x)$ is *concave and Schur-concave*. This indicates that supergaussianess (positive kurtosis) alone is *necessary* but *not sufficient* for inducing sparse solutions. Rather, sufficiency is given by the requirement that $-\log P_p(x) \approx d_p(x)$ be Concave/Schur-Concave(CSC).

We have seen that the function $d_p(x)$ has an interpretation as a (negative logarithm of) a Bayesian prior *or* as a penalty function enforcing sparsity in (7) where $d_p(x)$ should serve as a "relaxed counting function" on the nonzero elements of x. Our perspective emphasizes the fact that $d_p(x)$ serves *both* of these goals simultaneously. Thus, good regularizing functions, $d_p(x)$, should be flexibly parameterizable so that $P_p(x)$ can be optimized over the parameter vector p to provide a good parametric fit to the underlying environmental probability density function, *and* the functions should also have analytical properties consistent with the goal of enforcing sparse solutions. Such properties are discussed in the next section.

2.4 Majorization and Schur-Concavity [40]

Schur-Concave Functions. A measure of the sparsity of the elements of a solution vector x (or the lack thereof, which we refer to as the *diversity* of x) is given by a partial ordering on vectors known as the *Lorentz order*. For any vector in the positive orthant, $x \in R_+^n$, define the *decreasing rearrangement*

$$x \doteq (x_{\lfloor 1 \rfloor}, \cdots, x_{\lfloor n \rfloor}), \quad x_{\lfloor 1 \rfloor} \geq \cdots \geq x_{\lfloor n \rfloor} \geq 0$$

and the *partial sums* [40, 41],

$$S_x[k] = \sum_{i=1}^{k} x_{\lfloor n \rfloor}, \quad k = 1, \cdots, n.$$

We say that y *majorizes* x, $y \succ x$, iff for $k = 1, \cdots, n$,

$$S_y[k] \geq S_x[k]; \quad S_y[n] = S_x[n].$$

The vector y is more concentrated, or less *diverse*, than x. This partial order defined by majorization then defines the Lorentz order.

We are interested in scalar-valued functions of x which are consistent with majorization. Such functions are known as *Schur-Concave* functions, $d(\cdot) : R_+^n \to R$. They are defined to be precisely the class of functions which are *consistent with the Lorentz order*,

$$y \succ x \quad \Rightarrow \quad d(y) < d(x).$$

In words, if y is *less diverse than* x (according to the Lorentz order) then $d(y)$ is *less than* $d(x)$ for $d(\cdot)$ Schur-concave. We assume that Schur-Concavity is a *necessary condition* for $d(\cdot)$ to be a good *measure of diversity* (*anti-sparsity*).

Concavity yields sparse solutions. Recall that a function $d(\cdot)$ is *concave* on the positive orthant R_+^n iff [42]

$$d\left((1-\gamma)x + \gamma y\right) \geq (1-\gamma)d(x) + \gamma d(y),$$

$\forall x, y \in R_+^n, \forall \gamma, 0 \leq \gamma \leq 1$. In addition, a scalar function is said to be permutation invariant if its value is independent of rearrangements of its components. An important fact is that for permutation invariant functions *concavity is a sufficient condition for Schur-Concavity*:

$$\text{Concavity + Permutation Invariance } \Rightarrow \text{ Schur-Concavity.}$$

Now consider the low-noise sparse inverse problem (8). It is well known that subject to linear constraints, a concave function on R_+^n takes its minima on the *boundary* of R_+^n [42], and as a consequence these minima are therefore *sparse*. We take concavity to be a *sufficient condition* for $d(\cdot)$ to be a measure of diversity and we obtain sparsity as constrained minima of $d(\cdot)$. More generally, a diversity measure should be somewhere between Schur-concave and concave. In this spirit, one can define *almost concave* functions,[12] which are Schur-concave and (locally) concave in all n directions but one, which also are good measures of diversity.

Separability, Schur-Concavity, and ICA. The simplest way to ensure that $d(x)$ be permutation invariant (a necessary condition for Schur-concavity) is to use functions that are *separable*. Recall that separability of $d_p(x)$ corresponds to *factorizability* of $P_p(x)$. Thus *separability* of $d(x)$ corresponds to the assumption of *independent components* of x under the model (3). We see that from a Bayesian perspective, separability of $d(x)$ corresponds to a generative model for y that *assumes a source, x, with independent components.* With this assumption, we are working within the framework of Independent Component Analysis (ICA) [43, 38, 39]. We have developed effective algorithms for solving the optimization problem (8) for sparse solutions when $d_p(x)$ is separable and concave [12, 11].

It is now evident that relaxing the restriction of separability generalizes the generative model to the case were the source vector, x, has *dependent components*. We can reasonably call an approach based on a non-separable

diversity measure $d(x)$ a *Dependent Component Analysis* (DCA). Unless care is taken, this relaxation can significantly complicate the analysis and development of optimization algorithms. However, one can solve the low-noise DCA problem provided appropriate choices of non-separable diversity functions are made.[12, 11]

2.5 Supergaussian Priors and Sparse Coding

The P-class of diversity measures for $0 < p \leq 1$ result in sparse solutions to the low-noise coding problem (8). These separable and concave (and thus Schur-concave) diversity measures correspond to supergaussian priors, consistent with the "folk theorem" that supergaussian priors are sparsity enforcing priors. However, taking $1 \leq p < 2$ results in supergaussian priors which are *not* sparsity enforcing. Taking p to be between 1 and 2 yields a $d_p(x)$ which is *convex*, and therefore *not* concave. This is consistent with the well-known fact that for this range of p, the p^{th}-root of $d_p(x)$ is a norm. Minimizing $d_p(x)$ in this case drives x towards the origin, favoring "concentrated" rather than "sparse" solutions. We see that if a sparse coding is to be found based on obtaining a MAP estimate to the low-noise generative model (3) then, in a sense, supergaussianess is a necessary but not sufficient condition for a prior to be sparsity enforcing. A sufficient condition for obtaining a sparse MAP coding is that the negative log-prior be Concave/Schur-concave (CSC).

3 Dictionary Learning

3.1 Unknown, Nonrandom Dictionaries.

The Maximum Likelihood Estimation framework treats parameters to be estimated as unknown but deterministic. In this spirit we take the dictionary, A, to be the set of unknown parameters to be estimated from the observation set $Y = Y^N$. In particular, given Y^N the maximum likelihood estimate $\widehat{A}_{\text{ML}}$ is found from maximizing the likelihood function $L(A\,|Y^N) = P(Y^N; A)$. Under the assumption that the observations are iid, this corresponds to the optimization,

$$\widehat{A}_{\text{ML}} = \arg\max_A \prod_{k=1}^{N} P(y_k; A)\,, \tag{11}$$

$$P(y_k; A) = \int P(y_k, x; A)dx = \int P(y_k|x; A) \cdot P_p(x)dx = \int P_q(y_k - Ax) \cdot P_p(x)dx\,. \tag{12}$$

Defining the sample average of a function $f(y)$ over the sample set $Y^N = (y_1, \cdots, y_N)$ by

$$\langle f(y)\rangle_N = \frac{1}{N}\sum_{k=1}^{N} f(y_k)\,,$$

the optimization (11) can be equivalently written as

$$\widehat{A}_{\text{ML}} = \arg\min_A - \langle \log(P(y; A))\rangle_N\ . \tag{13}$$

Note that $P(y_k; A)$ is equal to the normalization factor β encountered earlier above, but now with the dependence of β on A and the particular sample, y_k, made explicit. The integration in (12) in general is intractable, and various approximations have been proposed to obtain an Approximate Maximum Likelihood estimate, $\widehat{A}_{\text{AML}}$ [26, 29].

Recently the following approximation has been proposed:[26]

$$P_p(x) \approx \delta(x - \widehat{x}_k(\widehat{A}))\,, \tag{14}$$

where

$$\widehat{x}_k(\widehat{A}) = \arg\max_x P(y_k, x; \widehat{A})\,, \tag{15}$$

for $k = 1, \cdots, N$, assuming a current estimate, $\widehat{A}$, for A. This approximation corresponds to assuming that x_k for which $y_k = Ax_k$ is known and equal to $\widehat{x}_k(\widehat{A})$. With this approximation, the optimization (13) becomes,

$$\widehat{A}_{\text{AML}} = \arg\min_A \left\langle d_q(y - \widehat{A}\widehat{x}) + \lambda\, d_p(\widehat{x}) \right\rangle_N , \tag{16}$$

which is an optimization over the sample average of the functional (7) encountered earlier. Updating our estimate for the dictionary,

$$\widehat{A} \leftarrow \widehat{A}_{\text{AML}} , \tag{17}$$

we can iterate the procedure (15)–(16) until $\widehat{A}_{\text{AML}}$ has converged, hopefully (at least in the limit of large N) to $\widehat{A}_{\text{ML}} = \widehat{A}_{\text{ML}}(Y^N)$ as the maximum likelihood estimate $\widehat{A}_{\text{ML}}(Y^N)$ has well-known desirable asymptotic properties in the limit $N \to \infty$.

Performing the optimization in (16) for the $q = 2$ iid gaussian measurement noise case (ν gaussian with known covariance $\sigma^2 \cdot I$) corresponds to taking

$$d_q(y - \widehat{A}\widehat{x}) = \frac{1}{2\sigma^2}\|y - \widehat{A}\widehat{x}\|^2 , \tag{18}$$

in (16). In the Appendix it is shown that we can readily obtain the unique 'batch' solution,

$$\widehat{A}_{\text{AML}} = \Sigma_{y\hat{x}}\Sigma_{\hat{x}\hat{x}}^{-1} , \tag{19}$$

$$\Sigma_{y\hat{x}} = \frac{1}{N}\sum_{k=1}^{N} y_k \widehat{x}_k^T , \quad \Sigma_{\hat{x}\hat{x}} = \frac{1}{N}\sum_{k=1}^{N} \widehat{x}_k \widehat{x}_k^T . \tag{20}$$

The Appendix actually derives the maximum likelihood estimate of A for the ideal case of *known* source vectors X,

$$\text{Known Source Vector Case: } \; A_{\text{ML}} = \Sigma_{yx}\Sigma_{xx}^{-1} ,$$

which is, of course, actually not computable since the source vectors $X = (x_1, \cdots, x_N)$ are assumed to be 'hidden.' Instead, as mentioned earlier, we pretend we know X by making the approximation $X \approx \widehat{X}(\widehat{A})$.

As an alternative to using the explicit solution (19), which requires an often prohibitive $n \times n$ inversion, we can obtain A_{AML} iteratively via gradient descent on (16)/(18),

$$\begin{aligned} \widehat{A}_{\text{AML}} &\leftarrow \widehat{A}_{\text{AML}} - \mu\,\frac{1}{N}\sum_{k=1}^{N} e_k \widehat{x}_k^T , \\ e_k &= \widehat{A}\widehat{x}_k - y_k , \quad k = 1, \cdots, N , \end{aligned} \tag{21}$$

for an appropriate choice of the (possibly adaptive) positive step-size parameter μ. Note the distinction in (21) between $\widehat{A}$ and $\widehat{A}_{\text{AML}}$, the latter hopefully converging to the batch estimate (19). The iteration (21) can be initialized as $\widehat{A}_{\text{AML}} = \widehat{A}$.

A general iterative dictionary learning procedure is obtained by nesting the iteration (21) entirely within the iteration defined by repeatedly solving (15) every time a new estimate, $\widehat{A}_{\text{AML}}$, of the dictionary becomes available. However performing the optimization required in (15) is generally nontrivial.[26, 29] Recently we have shown how the use of FOCUSS results in an effective algorithm for performing the optimization required in (15) for the case when ν is gaussian.[9, 20] This approach solves (15) using the Affine-Scaling Transformation (AST)-like algorithms recently proposed for the low noise case[6, 12, 11] and extended via regularization to the non-trivial noise case[9, 20]. For a current dictionary estimate, $\widehat{A}$, a solution to the optimization problem is provided by repeated iteration of the form,

$$\widehat{x}_k \leftarrow \Pi^{-1}(\widehat{x}_k)\widehat{A}^T \left(\beta(\widehat{x}_k)I + \widehat{A}\Pi^{-1}(\widehat{x}_k)\widehat{A}^T\right)^{-1} y_k , \tag{22}$$

$k = 1, \cdots, N$, with $\Pi(x)$ defined as in equation (28) given below. This is the regularized FOCUSS algorithm[7, 20] which has an interpretation as an AST-like concave function minimization algorithm. The proposed dictionary learning algorithm *alternates* between the iteration (22) and the iteration (21) (or the direct batch solution given by (19), if the inversion is tractable). Extensive simulations show the ability of the AST-based algorithm to completely recover an unknown 20×30 dictionary matrix A[20].

3.2 Unknown, Random Dictionaries.

We now generalize to the case where the dictionary, A, and the source vector set $X = X^N = (x_1, \cdots, x_N)$ are jointly random. We add the requirement that the dictionary is known to obey the constraint,

$$A \in \mathcal{A} = \text{compact submanifold of } \mathsf{R}^{m \times n}.$$

A compact submanifold of $\mathsf{R}^{m \times n}$ is necessarily closed and bounded. On the constraint submanifold the dictionary A has the prior probability density function $P(A)$, which in the sequel we assume has the simple (uniform on $\mathcal{A}$) form,

$$P(A) = c\,\mathcal{X}(A \in \mathcal{A})\,, \tag{23}$$

where $\mathcal{X}(\cdot)$ is the indicator function and c is a positive constant chosen to ensure that

$$P(\mathcal{A}) = \int_{\mathcal{A}} P(A)\, dA = 1\,.$$

The dictionary A and the elements of the set X are also all assumed to be mutually independent,

$$P(A, X) = P(A)\, P(X) = P(A) P_p(x_1) \cdots P_p(x_N)\,.$$

With the set of iid noise vectors, $(\nu_1, \cdots, \nu_N)$ also taken to be jointly random with, and independent of, A and X, the observation set $Y = Y^N = (y_1, \cdots y_N)$ is assumed to be generated via the model (3). With these assumptions we have

$$\begin{aligned} P(A, X|Y) &= P(Y|A, X)\, P(A, X)/P(Y) \\ &= c\,\mathcal{X}(A \in \mathcal{A})\, P(Y|A, X)\, P(X)/P(Y) \\ &= \frac{c\,\mathcal{X}(A \in \mathcal{A})}{P(Y)} \prod_{k=1}^{N} P(y_k|A, x_k)\, P_p(x_k) \\ &= \frac{c\,\mathcal{X}(A \in \mathcal{A})}{P(Y)} \prod_{k=1}^{N} P_q(y - Ax_k)\, P_p(x_k)\,, \end{aligned} \tag{24}$$

using the facts that the observations are conditionally independent and $P(y_k|A, X) = P(y_k|A, x_k)$.

The *jointly* Maximum A Posteriori (MAP) estimates

$$(\widehat{A}_{\text{MAP}}, \widehat{X}_{\text{MAP}}) = (\widehat{A}_{\text{MAP}}, \widehat{x}_{1,\text{MAP}}, \cdots, \widehat{x}_{1,\text{MAP}})$$

are found by maximizing *a posteriori* probability density $P(A, X|Y)$ simultaneously with respect to $A \in \mathcal{A}$ and X. This is equivalent to minimizing the negative logarithm of $P(A, X|Y)$, yielding the optimization problem,

$$(\widehat{A}_{\text{MAP}}, \widehat{X}_{\text{MAP}}) = \arg \min_{A \in \mathcal{A},\, X} \; \langle d_q(y - Ax) + \lambda\, d_p(x) \rangle_N \;. \tag{25}$$

Note that this is a *joint* minimization of the sample average of the functional (7), and as such is a natural generalization of the single (with respect to the set of source vectors) optimization previously encountered in (16). By finding joint

MAP estimates of A and X, we obtain a problem that is much more tractable than the one of finding the single MAP estimate of A (which involves maximizing the marginal posterior density $P(A|Y)$).

The requirement that $A \in \mathcal{A}$, where $\mathcal{A}$ is a compact and hence *bounded* subset of $\mathsf{R}^{m\times n}$, is sufficient for the optimization problem (25) to avoid the degenerate solution, [5]

$$\text{for } k = 1, \cdots, N\,, \;\; y_k = Ax_k\,, \text{ with } \|A\| \to \infty \text{ and } \|x_k\| \to 0\,. \tag{26}$$

This solution is possible for unbounded A because $y = Ax$ is almost always solvable for x since learned overcomplete A's are (generically) onto and for any solution pair (A, x) the pair $(\frac{1}{\alpha}A, \alpha\, x)$ is also a solution. This fact shows that the inverse problem of finding a solution pair (A, x) is generally ill-posed *unless* A is constrained to be bounded (as we've explicitly done here) or the cost functional is chosen to ensure that bounded A's are learned (e.g., by adding a term monotonic in the matrix norm $\|A\|$ to the cost function in (25)).

For the iid $q = 2$ gaussian measurement noise case of (18), algorithms that provably converge (in the low step-size limit) to a local minimum of (25) can be readily developed for the case,

$$\mathcal{A} = \{A \mid \|A\|_F = 1\} \subset \mathsf{R}^{m\times n}\,, \tag{27}$$

where $\|A\|_F$ denotes the Frobenius norm of the matrix A,

$$\|A\|_F^2 = \text{trace}\,(A^T A)\,,$$

and it is assumed that the prior $P(A)$ is uniformly distributed on $\mathcal{A}$ as per condition (23).

Following the gradient factorization procedure[12, 11], we factor the gradient of $d(x)$ as

$$\nabla d(x) = \alpha(x)\Pi(x)x\,, \quad \alpha(x) > 0\,, \tag{28}$$

where it is assumed that $\Pi(x)$ is diagonal and positive-definite for all nonzero x. We also define $\beta(x) = \lambda\alpha(x)$. A learning law which provably converges to a minimum of (25) on the manifold (27) is then given by,

$$\begin{aligned} \frac{d}{dt}\widehat{x}_k &= -\Omega_k \left\{ \left(\widehat{A}^T\widehat{A} + \beta(\widehat{x}_k)\Pi(\widehat{x}_k) \right) \widehat{x}_k - \widehat{A}^T y_k \right\}\,, \\ \frac{d}{dt}\widehat{A} &= -\mu \left(\delta\widehat{A} - \text{trace}\,(\widehat{A}^T\delta\widehat{A})\widehat{A} \right)\,, \;\; \mu > 0\,, \end{aligned} \tag{29}$$

for $k = 1, \cdots N$, where $\widehat{A}$ is initialized to $\|\widehat{A}\|_F = 1$, Ω_k are $n \times n$ positive definite matrices, and the "error" $\delta\,\widehat{A}$ is

$$\delta\widehat{A} = \left\langle e(\widehat{x})\widehat{x}^T \right\rangle_N = \frac{1}{N}\sum_{k=1}^{N} e(\widehat{x}_k)\widehat{x}_k^T\,, \quad e(\widehat{x}_k) = \widehat{A}\widehat{x}_k - y_k\,, \tag{30}$$

and which can be rewritten in the perhaps more illuminating form (cf. equations (19) and (20)),

$$\delta\widehat{A} = \widehat{A}\,\Sigma_{\hat{x}\hat{x}} - \Sigma_{y\hat{x}}\,. \tag{31}$$

A formal convergence proof is given in the Appendix, where it is also shown that the right-hand-side of the second equation in (29) corresponds to projecting the error term $\delta\,\widehat{A}$ onto the tangent space of $\mathcal{A}$ thereby ensuring that the derivative of $\widehat{A}$ lies in the tangent space. Convergence of the algorithm to a local optimum of (25) is formally proved by interpreting the loss functional as a Lyapunov function whose time derivative along the trajectories of the adapted parameters $(\widehat{A}, \widehat{X})$ is guaranteed to be negative-definite by the choice of parameter time derivatives shown in (29). As a consequence of the La Salle invariance principle, the loss functional will decrease in value and the parameters will converge to the largest invariant set for which the time derivative of the loss functional is identically zero [44].

[5] $\|A\|$ is any suitable matrix norm on A.

Note that (except for the trace term) the dictionary learning update in (29) is of the same form as for the AML update law given earlier in (21). The key difference is the additional trace term in (29). This difference corresponds to a projection of the update onto the tangent space of the manifold (27), thereby ensuring a unit Frobenius norm (and hence boundedness) of the dictionary estimate at all times and avoiding the ill-posedness problem indicated in (26). It is also of interest to note that choosing Ω_k to be

$$\Omega_k = \eta_k \left(\widehat{A}^T \widehat{A} + \beta(\widehat{x}_k) \Pi(\widehat{x}_k) \right)^{-1} , \ \eta_k > 0 \tag{32}$$

in (29), followed by some matrix manipulations, yields the alternative algorithm,

$$\frac{d}{dt} \widehat{x}_k = -\eta_k \left\{ \widehat{x}_k - \Pi^{-1}(\widehat{x}_k) \widehat{A}^T \left(\beta(\widehat{x}_k) I + \widehat{A} \Pi^{-1}(\widehat{x}_k) \widehat{A}^T \right)^{-1} y_k \right\} \tag{33}$$

with $\eta_k > 0$. In any event (regardless of the specific choice of the positive definite matrices Ω_k as shown in the Appendix), the proposed algorithm outlined here converges to a solution which satisfies the implicit and nonlinear relationships,

$$\begin{aligned} \widehat{x}_k &= \Pi^{-1}(\widehat{x}_k) \widehat{A}^T \left(\beta(\widehat{x}_k) I + \widehat{A} \Pi^{-1}(\widehat{x}_k) \widehat{A}^T \right)^{-1} y_k \, , \\ \widehat{A} &= \Sigma_{y\hat{x}}^T \left(\Sigma_{\hat{x}\hat{x}} - cI \right)^{-1} \in \mathcal{A} \, , \end{aligned} \tag{34}$$

for $k = 1, \cdots, N$ and scalar $c = \text{trace} \left(\widehat{A}^T \delta \widehat{A} \right)$. When implemented in discrete time, the Bayesian learning algorithm has the form of a *combined iteration* where we loop over the operations,

$$\begin{aligned} \widehat{x}_k &\leftarrow \Pi^{-1}(\widehat{x}_k) \widehat{A}^T \left(\beta(\widehat{x}_k) I + \widehat{A} \Pi^{-1}(\widehat{x}_k) \widehat{A}^T \right)^{-1} y_k \, , \\ k &= 1, \cdots, N \quad \text{and} \\ \widehat{A} &\leftarrow \widehat{A} - \gamma \left(\delta \widehat{A} - \text{trace} \, (\widehat{A}^T \delta \widehat{A}) \widehat{A} \right) \quad \gamma > 0 \, . \end{aligned} \tag{35}$$

This *merged* procedure should be compared to the *separate* iterations involved in the maximum likelihood approach given in (21)-(22) above. The projection in (35) of the dictionary update onto the tangent plane of $\mathcal{A}$ (see the discussion in the Appendix) can be crucial for ensuring the well-behavedness of the MAP algorithm [6]. Comparison with (21) shows that (35)/(30) corresponds to performing gradient descent on the manifold $\mathcal{A}$. The first equation in (35) corresponds to a specific step-size choice in the discrete-time algorithm[11]. Further analysis and discrete-time variants of the algorithms proposed in this subsection will be presented in the near future.

4 Experimental Results

To test the algorithm for learning an unknown, random dictionary (35) simulated data were created following the method in the references [20, 35]. The generating matrix A of size 20 x 30 is created by drawing each element from a normal distribution with $\mu = 0$, $\sigma^2 = 1$ (written as $\mathcal{N}(0,1)$) and normalizing such that $\|A\|_F = 1$. Sparse source vectors x_k, $k = 1 \ldots 1000$ are created with $r = 4$ non-zero elements. The magnitudes of each non-zero element are also drawn from $\mathcal{N}(0,1)$ and limited so that $x_{kl} > 0.1$. The y_k are generated using (1) and no noise is added.

For the first iteration of the algorithm, the columns of A_{init} are the first $n = 30$ training vectors y_k. The initial x vectors are set to the pseudoinverse solution $\widehat{x}_k = A^T (AA^T)^{-1} y_k$. Constant parameters are set as follows: $\gamma =$

[6] Because of the discrete-time approximation in (35), and even more generally because of numerical round-off effects in (29), a renormalization,

$$\widehat{A} \leftarrow \widehat{A} / \|A\|_F \, ,$$

is usually performed at regular intervals.

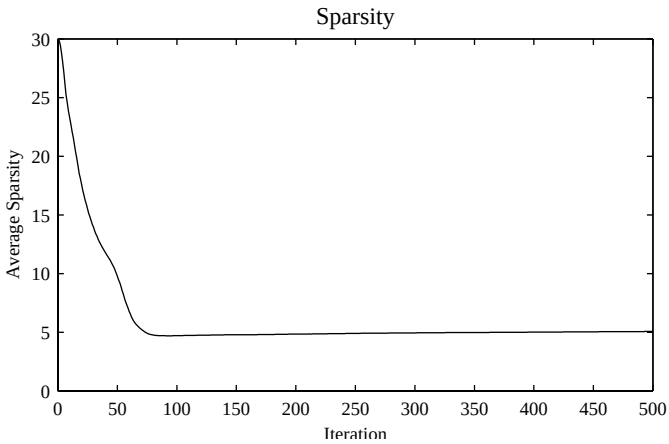

Figure 1: The average sparsity of $\widehat{x}_k$ at each iteration. At iteration 500 the average sparsity is 5.068.

1.0, $\alpha(x) = 0.8$, $p = 1.0$. The regularization parameter λ is a monotonically increasing function of the iteration number, with a maximum of 0.004. While this choice of λ does not have the optimality properties of the modified L-curve method (see Section 2.2), it does not require a one-dimensional optimization for each $\widehat{x}_k$ and so is much less computationally expensive.

The algorithm runs for 500 iterations, each iteration containing updates of $\widehat{A}$ after every 100 updates of $\widehat{x}_k$. Figure 1 shows the average sparsity of the learned vectors, which is found by counting the number of elements greater than 10^{-4} in each $\widehat{x}_k$. By adjusting the parameters λ and p the algorithm can be tuned to converge to different values of average sparsity, although aiming for the generated sparsity (in this case $r = 4$) gives the best results for reconstructing A. Figure 2 shows the sparsity of each learned vector $\widehat{x}_k$ after the final iteration. Most of the vectors have sparsity 4, while a few have higher numbers of high-magnitude elements. Figure 3 shows the number of columns $\widehat{a}_i$ of $\widehat{A}$ that match a column a_j in the original generating matrix A. The A matrix can only be learned to within sign and scale changes and column permutations. Before comparison the columns are normalized so that $||\widehat{a}_i|| = ||a_j|| = 1$. A match occurs if

$$1 - |a_j^T \widehat{a}_i| < 0.01. \tag{36}$$

The results in Figure 3 represent a best-case trial. When the same experiment is run repeatedly with different random data, the average number of learned columns of A is 27.3 (standard deviation, $\sigma = 2.11$, over 20 trials). In four of the trials all 30 columns were learned to within the tolerance. In comparison, the algorithm [20, 35] which allows the FOCUSS iterations to converge for each $\widehat{x}_k$ (and without using the L-curve, see Section 2.2) averages 26.9, $\sigma = 2.38$, essentially equivalent performance.

In the case where A is square (size 30 x 30), the algorithm (35) learns on average 29.6 ($\sigma = 0.99$) of the columns, with 16 of the 20 trials matching all 30 columns.

5 APPENDIX – CONVERGENCE OF THE LEARNING ALGORITHM

Here we provide a formal proof that the algorithm (29)-(30) converges to a local minimum of (25) on the manifold (27).

First we consider the form of admissible derivatives, $\dot{A} = \frac{d}{dt}A$ for matrices belonging to (27). Towards this end, we view (27) as embedded in the finite dimensional Hilbert space of matrices, $\mathsf{R}^{m \times n}$, with inner product

$$< A, B >= \text{trace}\,(A^T B) = \text{trace}\,(B^T A) = \text{trace}\,(AB^T) = \text{trace}\,(BA^T)\,.$$

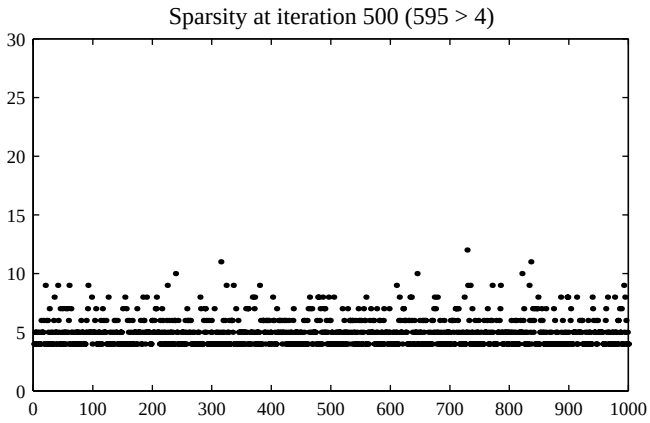

Figure 2: The sparsity of each $\widehat{x}_k$ at iteration 500. The data was generated with sparsity 4.

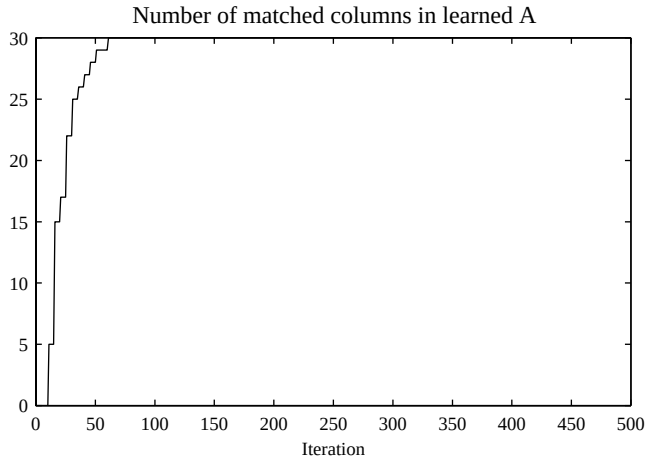

Figure 3: Number of columns in A that match a column in A_{orig}.

The corresponding matrix norm is the *Frobenius norm*,

$$\|A\| = \|A\|_F = \sqrt{\text{trace}\, A^T A} = \sqrt{\text{trace}\, A A^T}\,.$$

We will call this space the *Frobenius Space* and the associated inner product the *Frobenius inner product*. It is useful to note the isometry,

$$A \in \mathsf{R}^{m\times n} \Longleftrightarrow \mathbf{A} = \text{vec}\,(A) \in \mathsf{R}^{mn}\,,$$

where $\mathbf{A}$ is the mn-vector formed by "stacking" the columns of A. Henceforth, bolding represents the stacked version of a matrix (e.g., $\mathbf{B} = \text{vec}(B)$). The stacked vector $\mathbf{A}$ belongs to the standard Hilbert space R^{mn}, which we shall henceforth refer to as the *Stacked Space*. This space has the standard Euclidean inner product and norm,

$$< \mathbf{A}, \mathbf{B} >= \mathbf{A}^T\mathbf{B}\,, \quad \|\mathbf{A}\| = \sqrt{\mathbf{A}^T\mathbf{A}}\,.$$

It is straightforward to show that

$$< A, B >=< \mathbf{A}, \mathbf{B} > \quad \text{and} \quad \|A\| = \|\mathbf{A}\|\,.$$

In particular, we have

$$A \in \mathcal{A} \Longleftrightarrow \|A\| = \|\mathbf{A}\| = 1\,.$$

Thus, the manifold (27) corresponds to the unit sphere in the Stacked Space, R^{mn} (which, with a slight abuse of notation, we will continue to denote by $\mathcal{A}$). Determining the form of admissible derivations on (27) is equivalent to determining the form of admissible derivatives on the unit R^{mn}–sphere.

Note, then, that on the unit sphere, we have the well-known fact that

$$\mathbf{A} \in \mathcal{A} \Longrightarrow \frac{d}{dt}\|\mathbf{A}\|^2 = 2\,\mathbf{A}^T\dot{\mathbf{A}} = 0 \Longrightarrow \dot{\mathbf{A}} \perp \mathbf{A}\,.$$

This shows that the general form of $\dot{\mathbf{A}}$ is $\dot{\mathbf{A}} = \mathbf{\Lambda Q}$, where $\mathbf{Q}$ is arbitrary and

$$\mathbf{\Lambda} = \left(\mathbf{I} - \frac{\mathbf{A}\mathbf{A}^T}{\|\mathbf{A}\|^2}\right) = \left(\mathbf{I} - \mathbf{A}\mathbf{A}^T\right) \tag{37}$$

is the Stacked Space projection operator onto the tangent space of the unit R^{mn}–sphere at the point $\mathbf{A}$. (Note that we used the fact that $\|A\| = 1$.) This projection can be easily rewritten in the Frobenius Space,

$$\mathbf{\Lambda Q} = \mathbf{Q} - < \mathbf{A}, \mathbf{Q} > \mathbf{A} \Longleftrightarrow \Lambda Q = Q - < A, Q > A = Q - \text{trace}\,(A^T Q)A\,. \tag{38}$$

Of course this result can be derived directly in the Frobenius Space using the fact that

$$A \in \mathcal{A} \Longrightarrow \frac{d}{dt}\|A\|^2 = 2 < A, \dot{A} >= 2\,\text{trace}\,(A^T\dot{A}) = 0\,,$$

from which it is directly evident that $\dot{A}$ must be of the form[7]

$$\dot{A} = \Lambda Q = Q - \frac{\text{trace}\,(A^T Q)}{\text{trace}\,(A^T A)}A = Q - \text{trace}\,(A^T Q)A\,, \tag{39}$$

but we have found it useful to work in the Stacked Space where one can more readily motivate the derivations for those readers who have not had an introduction to abstract Hilbert spaces. In the Stacked Space (with some additional abuse of notation) we represent the quadratic form for a positive semidefinite symmetric matrices $\mathbf{W}$ as

$$\|\mathbf{A}\|^2_{\mathbf{W}} = \mathbf{A}^T\mathbf{W}\mathbf{A}\,.$$

[7] I.e., it must be the case that $\Lambda = I - \frac{|A><A|}{\|A\|^2} = I - |A><A|$, using the physicist's "bra-ket" notation.

Note that this is a weighted norm if, and only if, $\mathbf{W}$ is positive definite, which might not be the case by definition. Finally, note from (39) that $\forall A \in \mathcal{A}$,

$$\Lambda Q = 0 \Longleftrightarrow Q = c\,A\,, \;\; \text{with} \;\; c = \text{trace}\,(A^T Q)\,. \tag{40}$$

Consider the Lyapunov function,

$$V_N(X,A) = \langle d_q(y - Ax) + \lambda\, d_p(x) \rangle_N \;\;, \quad A \in \mathcal{A}\,, \tag{41}$$

which is precisely the loss function to be minimized in (25). If we can determine smooth parameter trajectories (i.e., a parameter-vector adaptation rule) $(\dot{X}, \dot{A})$ such that along these trajectories $\dot{V}(X,A) \leq 0$, then as a consequence of the La Salle invariance principle[44] the parameter values will converge to the largest invariant set (of the adaptation rule viewed as a nonlinear dynamical system) contained in the set

$$\Gamma = \left\{ (X,A) \,|\, \dot{V}_N(X,A) \equiv 0 \text{ and } A \in \mathcal{A} \right\}\,. \tag{42}$$

The set Γ contains the local minima of V_N. With some additional technical assumptions (generally dependent upon the choice of adaptation rule), the elements of Γ will contain only local minima of V_N.

Assuming the iid $q = 2$ gaussian measurement noise case of (18),[8] the loss (Lyapunov) function to be minimized is then,

$$V_N(X,A) = \left\langle \frac{1}{2}\|Ax - y\|^2 + \lambda\, d_p(x) \right\rangle_N \;\;, \quad A \in \mathcal{A}\,, \tag{43}$$

which is essentially the loss function to be minimized in (25).[9]

Suppose for the moment, as in (14)–(19), that X is assumed to be known and note that then (ignoring constant terms depending on X and Y) V_N can be rewritten as

$$V_N(A) = \text{trace}\,\left\{ A\Sigma_{xx}A^T - 2A\Sigma_{xy} \right\}\,,$$

for Σ_{xx} and $\Sigma_{xy} = \Sigma_{yx}^T$ defined as in (20). Using standard results from matrix calculus[45], we can show that $V_N(A)$ is minimized by the solution (18). This is done by setting,

$$\frac{\partial}{\partial A} V_N(\widehat{A}) = 0\,,$$

and using the identities (valid for W symmetric),

$$\frac{\partial}{\partial A}\text{trace}\,AWA^T = 2AW \;\; \text{ and } \;\; \frac{\partial}{\partial A}\text{trace}\,AB = B^T \;\;.$$

This yields,

$$\frac{\partial}{\partial A} V_N(\widehat{A}) = 2\widehat{A}\Sigma_{xx} - 2\Sigma_{xy}^T = 2\left(\widehat{A}\Sigma_{xx} - \Sigma_{yx}\right) = 0 \;\;,$$

which (assuming that Σ_{xx} is invertible) results in (19) as claimed. For Σ_{xx} non-singular, the solution is unique and globally optimal. This is, of course, a well-known result.

Now return to the general case (43), where both X and A are unknown. For the data indexed by $k = 1, \cdots, N$, define the quantities

$$d_k = d(x_k)\,, \;\; d(x) = Ax - y\,, \;\; \text{and} \;\; e_k = Ax_k - y_k\,.$$

[8] Note that in the appendix, unlike the notation used in equation (18) *et seq.*, the "hat" notation has been dropped. Nonetheless, it should be understood that the quantities A and X are unknown parameters to be optimized over in (25), while the measured signal-vectors Y are known.

[9] The factor $\frac{1}{2}$ is added for notational convenience and does not materially affect the derivation.

Then, to determine an appropriate adaptation rule, note that

$$\dot{V}_N = T_1 + T_2 \,, \tag{44}$$

where

$$T_1 = \left\langle \left(e^T(x)A + \lambda \nabla^T d(x) \right) \dot{x} \right\rangle_N = \frac{1}{N} \sum_{k=1}^{N} \left(e_k^T A + \lambda \nabla^T d_k \right) \dot{x}_k \tag{45}$$

and

$$T_2 = \left\langle e^T(x) \dot{A} x \right\rangle_N = \frac{1}{N} \sum_{k=1}^{N} e_k^T \dot{A} x_k \,. \tag{46}$$

Enforcing the *separate* conditions

$$T_1 \leq 0 \text{ and } T_2 \leq 0$$

(as well as the additional condition that $A \in \mathcal{A}$) will be sufficient to ensure that $\dot{V}_N \leq 0$. In this case the solution-containing set Γ of (42) is given by

$$\Gamma = \{ (X, A) \,|\, T_1(X, A) \equiv 0 \,,\; T_2(X, A) \equiv 0 \text{ and } A \in \mathcal{A} \} \,. \tag{47}$$

To force the condition $T_1 \leq 0$ and derive the first adaptation rule given in (29), we note that we can factor $\nabla d_k = \nabla d(x_k)$ as

$$\nabla d_k = \alpha_k \Pi_k x_k$$

with $\alpha_k = \alpha_{x_k} > 0$ and $\Pi_{x_k} = \Pi_k$ positive definite and diagonal for all nonzero x_k. Then, defining $\beta_k = \lambda \alpha_k > 0$ and selecting an arbitrary set of (adaptable) symmetric positive-definite matrices Ω_k, we choose the learning rule

$$\dot{x}_k = -\Omega_k \left\{ A^T e_k + \lambda \nabla d_k \right\} = -\Omega_k \left\{ \left(A^T A + \beta_k \Pi_k \right) x_k - A^T y_k \right\} \,, \;\; k = 1, \cdots, N \,, \tag{48}$$

which is the adaptation rule for the state estimates $x_k = \widehat{x}_k$ given in the first line of (29). With this choice we obtain

$$T_1 = -\left\langle \| A^T e(x) + \lambda \nabla d(x) \|_\Omega^2 \right\rangle = -\frac{1}{N} \sum_{k=1}^{N} \| A^T e_k + \lambda \nabla d_k \|_{\Omega_k}^2 = -\frac{1}{N} \| \left(A^T A + \beta_k \Pi_k \right) x_k - A^T y_k \|_{\Omega_k}^2 \leq 0 \,, \tag{49}$$

as desired. Assuming convergence to the set (47) (which will be seen to be the case after we show below how to ensure that we also have $T_2 \leq 0$), we will asymptotically obtain (reintroducing the "hat" notation to now denote converged parameter estimates)

$$\| \left(\widehat{A}^T \widehat{A} + \beta_k \Pi_k \right) \widehat{x}_k - \widehat{A}^T y_k \|_{\Omega_k}^2 \equiv 0 \,,\; k = 1, \cdots, N \,,$$

which is equivalent to

$$\widehat{x}_k = \left(\widehat{A}^T \widehat{A} + \beta_k \Pi_k \right)^{-1} \widehat{A}^T y_k \,, \;\; k = 1, \cdots, N \,, \tag{50}$$

at convergence.

Exploiting the fact that Ω_k in (48) are arbitrary (subject to the symmetry and positive-definiteness constraint), let us make the specific choice shown in (32),

$$\Omega_k = \eta_k \left(A A^T + \beta_k \Pi_k \right)^{-1} \,, \;\; \eta_k > 0 \,, \;\; k = 1, \cdots, N \,. \tag{51}$$

Also note the (trivial) identity,

$$\left(A^T A + \beta \Pi \right) \Pi^{-1} A^T = A^T \left(A \Pi^{-1} A^T + \beta I \right) \,,$$

which can be recast nontrivially as

$$\left(A^T A + \beta\Pi\right)^{-1} A^T = \Pi^{-1} A^T \left(\beta I + A\Pi^{-1} A^T\right)^{-1} . \tag{52}$$

With (51) and (52), the learning rule (48) can be recast as

$$\dot{x}_k = -\eta_k \left\{ x_k - \Pi_k^{-1} A^T \left(\beta_k I + A\Pi_k^{-1} A^T\right)^{-1} y_k \right\} , \;\; k = 1, \cdots, N , \tag{53}$$

which is the alternative learning algorithm (33). At convergence (when $T_1 \equiv 0$) we have the condition shown in the first line of (34),

$$\widehat{x}_k = \Pi^{-1} \widehat{A}^T \left(\beta_k I + \widehat{A}\Pi^{-1} \widehat{A}^T\right)^{-1} y_k . \tag{54}$$

This also follows from the convergence condition (50) and the identity (52), showing that the result (54) is independent of the specific choice of $\Omega_k > 0$. Note from (48) and (49) that $T_1 \equiv 0$ also results in $\dot{x}_k \equiv 0$ for $k = 1, \cdots, N$, so that we will have converged to constant values, $\widehat{x}_k$, which satisfy (54).

We now turn to the enforcement of the second convergence condition, $T_2 \leq 0$ and the development of the dictionary adaptation rule shown in (29). First, as in (30) we define the error term δA as

$$\delta A = \left\langle e(x) x^T \right\rangle_N = \frac{1}{N} \sum_{k=1}^{N} e(x_k) x_k^T = A\, \Sigma_{xx} - \Sigma_{yx} , \tag{55}$$

using the fact that $e(x) = Ax - y$. Then, from (46), we have

$$T_2 = \left\langle e^T(x) \dot{A} x \right\rangle_N = \left\langle \text{trace} \left(x e^T(x) \dot{A} \right) \right\rangle_N = \text{trace} \left(\left\langle x e^T(x) \right\rangle_N \dot{A} \right) = \text{trace} \left(\delta A^T \dot{A} \right) = \delta \mathbf{A}^T \dot{\mathbf{A}} . \tag{56}$$

Thus, to ensure that $T_2 \leq 0$ and that $\dot{A}$ is in the tangent space of the unit sphere in R^{mn}, we take

$$\dot{\mathbf{A}} = -\mu \mathbf{\Lambda} \delta \mathbf{A} \Longleftrightarrow \dot{A} = -\mu \Lambda \delta A = -\mu \left(\delta A - \text{trace}\,(A^T \delta A) A \right) , \;\; \mu > 0 , \tag{57}$$

which is the adaptation rule given in (29). With this choice, we have

$$T_2 = -\mu \|\delta \mathbf{A}\|_{\mathbf{\Lambda}}^2 \leq 0 ,$$

as required. Note that at convergence, the condition $T_2 \equiv 0$, yields $\dot{A} \equiv 0$, so that we will have converged to constant values for the dictionary elements, and

$$0 = \Lambda \delta \widehat{A} = \Lambda \left(\widehat{A} \Sigma_{\hat{x}\hat{x}} - \Sigma_{y\hat{x}} \right) \Longrightarrow \delta \widehat{A} = \left(\widehat{A} \Sigma_{\hat{x}\hat{x}} - \Sigma_{y\hat{x}} \right) = c\, \widehat{A} , \tag{58}$$

from (40), where $c = \text{trace} \left(\widehat{A}^T \delta \widehat{A} \right)$. Thus, the steady-state solution is

$$\widehat{A} = \Sigma_{y,\hat{x}} \left(\Sigma_{\hat{x}\hat{x}} - cI \right)^{-1} \in \mathcal{A} . \tag{59}$$

Note that (54) and (59) are the steady state values given earlier in (34).

References

[1] I. Gorodnitsky, J. George, and B. Rao, "Neuromagnetic Source Imaging with FOCUSS: a Recursive Weighted Minimum Norm Algorithm," *Journal of Electroencephalography and Clinical Neurophysiology* **95**, pp. 231–251, October 1995.

[2] B. Rao, "Analysis and Extensions of the FOCUSS Algorithm," in *Proceedings of the 1996 Asilomar Conference on Circuits, Systems, and Computers*, vol. 2, pp. 1218–23, IEEE, (New York), 1997.

[3] B. Rao and I. Gorodnitsky, "Affine Scaling Transformation Based Methods for Computing Low Complexity Sparse Solutions," in *Proc. 1996 ICASSP*, vol. III, pp. 1783–86, IEEE, (New York), 1997.

[4] I. Gorodnitsky and B. Rao, "Sparse Signal Reconstruction from Limited Data Using FOCUSS: A Re-weighted Minimum Norm Algorithm," *IEEE Trans. Signal Processing* **45**, pp. 600–616, March 1997.

[5] J. Adler, B. Rao, and K. Kreutz-Delgado, "Comparison of Basis Selection Methods," in *Proceedings of the* 30^{th} *Asilomar Conference on Signals, Systems, and Computers*, November 1996.

[6] B. Rao and K. Kreutz-Delgado, "Deriving Algorithms for Computing Sparse Solutions to Linear Inverse Problems," in *Proceedings of the 1997 Asilomar Conference on Circuits, Systems, and Computers*, J. Neyman, ed., IEEE, (New York), 1997.

[7] B. Rao, "Signal Processing with the Sparseness Constraint," in *Proc. 1998 ICASSP*, IEEE, (New York), 1998.

[8] D. Donoho, "On Minimum Entropy Segmentation," in *Wavelets: Theory, Algorthms, and Applications*, C. Chui, L. Montefusco, and L. Puccio, eds., pp. 233–269, Academic Press, 1994.

[9] B. Rao and K. Kreutz-Delgado, "Basis Selection in the Presence of Noise," in *Proceedings of the 1998 Asilomar Conference*, IEEE, (New York), 1998.

[10] B. Rao and K. Kreutz-Delgado, "Sparse Solutions to Linear Inverse Problems with Multiple Measurement Vectors," in *Proceedings of the 8th IEEE Digital Signal Processing Workshop*, IEEE, (New York), 1998.

[11] B. Rao and K. Kreutz-Delgado, "An Affine Scaling Methodology for Best Basis Selection," *IEEE Trans. Signal Processing* **1**, pp. 187–202, January 1999.

[12] K. Kreutz-Delgado and B. Rao, "*A General Approach to Sparse Basis Selection: Majorization, Concavity, and Affine Scaling* (Full Report with Proofs)." Center For Information Engineering Report No. UCSD-CIE-97-7-1, ECE Department, UC San Diego. URL: http://raman.ucsd.edu, July 1997.

[13] K. Kreutz-Delgado and B. Rao, "Measures and Algorithms for Best Basis Selection," in *Proceedings of the 1998 ICASSP*, IEEE, (New York), 1998.

[14] K. Kreutz-Delgado and B. Rao, "Gradient Factorization-Based Algorithm for Best-Basis Selection," in *Proceedings of the 8th IEEE Digital Signal Processing Workshop*, IEEE, (New York), 1998.

[15] K. Kreutz-Delgado and B. Rao, "A New Algorithm and Entropy-like Measures for Sparse Coding," in *Proc. 5th Joint Symp. Neural Computation*, UC San Diego, (La Jolla, CA), 1998.

[16] K. Kreutz-Delgado and B. Rao, "Sparse Basis Selection, ICA, and Majorization: Towards a Unified Perspective," in *Proc. 1999 ICASSP*, (Phoenix, AZ), 1999.

[17] K. Kreutz-Delgado, B. Rao, K. Engan, T.-W. Lee, and T. Sejnowski, "Convex/Schur–Convex (CSC) Log-Priors and Sparse Coding," in *Proc. 6th Joint Symposium on Neural Computation*, (Caltech, Pasadena, California), May 1999.

[18] K. Kreutz-Delgado, B. Rao, K. Engan, T.-W. Lee, and T. Sejnowski, "Learning Overcomplete Dictionaries: Convex/Schur-Convex (CSC) Log-Priors, Factorial Codes, and Independent/Dependent Component Analysis (I/DCA)," in *Proc. 6th Joint Symposium on Neural Computation*, (Caltech, Pasadena, California), May 1999.

[19] K. Kreutz-Delgado, B. Rao, and K. Engan, "Novel Algorithms for Learning Overcomplete Dictionaries," in *Proc. 1999 Asilomar Conference*, (Pacific Grove, California), November 1999.

[20] K. Engan, B. Rao, and K. Kreutz-Delgado, "Frame Design Using FOCUSS with Method of Optimal Directions (MOD)," *Proc. NORSIG-99* , September 1999.

[21] D. Field, "What is the Goal of Sensory Coding," *Neural Computation* **6**, pp. 559–99, 1994.

[22] D. Ruderman, "The Statistics of Natural Images," *Network: Computation in Neural Systems* **5**, pp. 517–48, 1994.

[23] P. Comon, "Independent Component Analysis: A New Concept?," *Signal Processing* **36**, pp. 287–314, 1994.

[24] A. Bell and T. Sejnowski, "An Information-Maximization Approach to Blind Separation and Blind Deconvolution," *Neural Computation* **7**, pp. 1129–59, 1995.

[25] G. Deco and D. Obradovic, *An Information-Theoretic Approach to Neural Computing*, Springer, 1996.

[26] B. Olshausen and D. Field, "Emergence of Simple-Cell Receptive Field Properties by Learning a Sparse Code for Natural Images," *Nature* **381**, pp. 607–9, June 1996.

[27] S. Zhu, Y. Wu, and D. Mumford, "Minimax Entropy Principle and its Application to Texture Modeling," *Neural Computation* **9**, pp. 1627–60, 1997.

[28] K. Wang, C. Lee, and B. Juang, "Selective Feature Extraction via Signal Decomposition," *IEEE Signal Processing Letters* **4**(1), pp. 8–11, 1997.

[29] M. Lewicki and B. Olshausen, "Inferring Sparse, Overcomplete Image Codes Using an Efficient Coding Framework," April 1998. Submitted to *J. Opt. Soc. America A*.

[30] R. Coifman and M. Wickerhauser, "Entropy-Based Algorithms for Best Basis Selection," *IEEE Transactions on Information Theory* **IT-38**, pp. 713–18, March 1992.

[31] S. Mallat and Z. Zhang, "Matching Pursuits with Time-Frequency Dictionaries," *IEEE Trans. ASSP* **41**(12), pp. 3397–415, 1993.

[32] D. Donoho, "De-noising by Soft-thresholding," *IEEE Trans. on Information Theory* **41**, pp. 613–27, 1995.

[33] S. Watanabe, "Pattern Recognition as a Quest for Minimum Entropy," *Pattern Recognition* **13**(5), pp. 381–87, 1981.

[34] A. Basilevsky, *Statistical factor analysis and related methods: theory and applications*, Wiley, 1994.

[35] K. Engan, *Frame based signal representation and compression*. PhD thesis, Stavanger Universty College, Norway, 2000.

[36] P. C. Hansen and D. P. O'Leary, "The use of the L-curve in the regularization of discrete ill-posed problems," *SIAM J. Sct. Comput.* **14**, pp. 1487–1503, November 1993.

[37] S. Kassam, *Signal Detection in Non-Gaussian Noise*, Springer-Verlag, New York, 1982.

[38] D. Pham, "Blind Separation of Instantaneous Mixture of Sources via an Independent Component Analysis," *IEEE Trans. Signal Processing* **44**, pp. 2768–79, November 1996.

[39] S. Roberts, "Independent Component Analysis: Source Assessment and Sepration, a Bayesian Approach," *IEE (Brit.) Proc. Vis. Image Signal Processing* **145**, pp. 149–54, June 1998.

[40] A. Marshall and I. Olkin, *Inequalities: Theory of Majorization and Its Applications*, Academic Press, 1979.

[41] M. Wickerhauser, *Adapted Wavelet Analysis from Theory to Software*, A.K. Peters, Wellesley, MA, 1994.

[42] R. Rockafellar, *Convex Analysis*, Princeton University Press, 1970.

[43] J.-P. Nadal and N. Parga, "Nonlinear Neurons in the Low Noise Limit: A Factorial Code Maximizes Information Transfer," *Network* **5**, pp. 565–81, 1994.

[44] H. Khalil, *Nonlinear Systems*, Prentice Hall, Upper Saddle River, NJ, second ed., 1996.

[45] P. Dhrymes, *Mathematics for Econometrics*, Springer-Verlag, New York, 2nd ed., 1984.