Theory and algorithms for linear and nonlinear component analysis

J. A. Palmer

Abstract

In this report we describe a general framework for component estimation and basis vector learning. This problem is important in signal processing areas like Independent Component Analysis (ICA), Blind Source Separation (BSS), wavelet analysis and signal denoising [1], image reconstruction [2, 3], Factor Analysis [4], and sparse coding [5, 6]. It is also useful for unsupervised feature extraction for general classification problems in statistical learning.

The theory is based on the idea of generalized convexity, in particular an independently derived theory called relative convexity. We use this theory to characterize super-gaussian random variables, which are common in real-world signals, and often assumed ICA models, and we show how this theory can be used to derive convergent algorithms for component estimation and learning basis vectors. We attempt to unify the theory underlying variational algorithms based on convex duality and Automatic Relevance algorithms based on the evidence framework of Mackay [7].

1 Introduction

This paper is concerned with unsupervised learning of data representations, specifically linear and nonlinear component analysis representations, for signal processing applications and for general statistical learning tasks.

Given data vectors $\mathcal{D} = \{y_k\}_{k=1}^{N} \subset \mathcal{H}$, where $\mathcal{H}$ is a finite or infinite dimensional Hilbert space, linear component analysis finds a finite set of basis vectors $\{a_i\}_{i=1}^{n} \subset \mathcal{H}$, such that,

$$y_k = \sum_{i=1}^{n} x_{k,i} a_i + \epsilon_k \quad k = 1, \ldots, N$$

where $\epsilon$ is the error in the representation. The vector $x_k \in \mathbb{R}^n$ is the component vector constituting the analysis of the data vector $y_k$ in terms of the basis vectors $a_i, i = 1, \ldots, n$.

Nonlinear component analysis first maps each data point $y_k$ to the discrete, possibly infinite vector $\Phi(y_k) \in \mathcal{F}$, where $\mathcal{F}$ is a higher dimension Hilbert space referred to as “feature space”. Linear component analysis is then performed.
in feature space. The “features” in the feature vector are derived from the expansion of a symmetric kernel function $K(x, y)$ in terms of its eigenfunctions,

$$K(x, y) = \sum_{j=1}^{\infty} \lambda_j \phi_j(x) \phi_j(y)$$

The $j$th component of the feature vector, $\Phi_j(y)$ is $\sqrt{\lambda_j} \phi_j(y)$. The data is then represented by the model,

$$\Phi(y_k) = \sum_{i=1}^{n} x_{k,i} \Phi(v_i) + \epsilon_k \quad k = 1, \ldots, N$$

where the $v_i \in \mathcal{H}$ are to be determined.

We consider a framework for component analysis based on statistical estimation theory, regularization theory, and convex analysis.

1.1 Statistical framework

Component analysis can be set in a statistical context, where the assumed noise distribution and prior distribution on the components operate in a manner similar to the functions $d$ and $f$ in the optimization context. Consider the Maximum Likelihood estimate of the $m \times n$ matrix $A$ in the linear model,

$$y = Ax + \nu$$

where $\nu$ is a noise vector. Given $Y = [y_1 \ldots y_N]$, the ML estimate of $A$ is,

$$\hat{A}_{ML} = \arg\max_A p(Y; A) = \arg\max_A \int p(Y|X; A) p(X) dX$$

If the assumed noise and prior distributions are Gaussian and the data and components are assumed to be independent, then we have,

$$\int p(Y|X; A) p(X) dX = \prod_{k=1}^{N} \int \mathcal{N}(y_k; Ax_k, \Sigma) \mathcal{N}(x_k; 0, \frac{1}{\gamma}I) dx_k$$

Taking the negative logarithm, the problem is equivalent to,

$$\hat{A}_{ML} = \arg\min_A \sum_{k=1}^{N} \gamma y_k^T \left(I + \frac{1}{\gamma}AA^T\right)^{-1} y_k + N \log \left|I + \frac{1}{\gamma}AA^T\right|$$

At a stationary point, $\hat{A}_{ML}$ satisfies,

$$\left(I + \frac{1}{\gamma}AA^T\right)^{-1} \left(\sum_{k=1}^{N} y_k y_k^T\right) \left(I + \frac{1}{\gamma}AA^T\right)^{-1} = N \left(I + \frac{1}{\gamma}AA^T\right)^{-1} A$$
or equivalently,
\[
\left( \frac{1}{N} \sum_{k=1}^{N} y_k y_k^T \right) \left( I + \frac{1}{\gamma} AA^T \right)^{-1} A = A
\]

Using the matrix inversion lemma, this can be simplified to,
\[
\left( \frac{1}{N} \sum_{k=1}^{N} y_k y_k^T \right) A = \left( I + \frac{1}{\gamma} AA^T \right) A
\]

Using the reduced SVD of \( A = U_1 \Sigma_1 V_1^T \), where \( U_1 \) is an orthonormal matrix spanning the range of \( A \), we see that,
\[
\frac{1}{N} \left( \sum_{k=1}^{N} y_k y_k^T \right) U = UD
\]

where \( D \) is diagonal. The minimization in (4) implies that \( U \) contains the eigenvectors corresponding to the \( n \) largest eigenvalues. See [8] for further details. Thus any \( A \) whose range is the subspace corresponding to these eigenvectors is a stationary point, and is in fact a local maximum of the likelihood [8]. There is thus some indeterminacy in the Maximum Likelihood estimate with Gaussian priors, and further restriction on \( A \) is needed to recover \( \hat{A}_{PCA} \).

If the noise or the assumed prior on the components is non-gaussian, then the integral in (3) is generally intractable. We can proceed however by using a variational EM algorithm, treating the components as hidden data, or by using convex duality to bound the likelihood and optimize this instead. We consider these approaches in section 5.

The determinant term may

In section 4 we derive an adaptive algorithm for normalized component analysis which may be seen as an approximate Newton algorithm for finding a stationary point of the Lagrangian.

1.2 Independent Component Analysis

Bases can also be determined by the requirement that the estimated components be independent. The Bayesian linear model,
\[
y = Ax + \nu
\]
determines an estimate \( \hat{x}_k \) for each observation \( y_k \), which may be thought of as a deterministic, generally nonlinear mapping defining the random variable \( \hat{x}_k = \mathcal{W}(y_k) \). If we make the prior assumption or requirement that the components are independent, then we can seek a mapping \( \mathcal{W} \) that makes the components of \( \hat{x}_k \) “as independent as possible”. One approach to determining independent components is to minimize the mutual information in \( \hat{x} \). The mutual information is the Kullback-Leibler divergence between the joint distribution and the product of the marginals, and is thus non-negative and vanishes if and
only if the random variables are independent. Minimizing an estimate of the mutual information thus tends to make the components independent. The mutual information can also be written as the difference between the sum of the marginal entropies and the joint entropy \[9\],

\[
I(\hat{x}_1; \hat{x}_2; \cdots; \hat{x}_n) = \left( \sum_{i=1}^{n} H(\hat{x}_i) \right) - H(\hat{x})
\]

The second term can be seen as characteristic of size or complexity of the class of mappings from which we choose \(W\). \(H(\hat{x})\) is increased if the mapping \(W\) increases uncertainty beyond that present in \(y\) \[9\]. If we restrict the class of mappings such that \(H(\hat{x}) \approx H(y)\), for example in the linear case restricting \(W = W\) to be a volume preserving mapping, then this term is not essential for the purpose of enforcing independence. This term however is in fact the focus of the “maximum entropy” approach to ICA used in \[10\], where the mapping \(W\) is constrained to be a neural network and the entropy is maximized by choosing the nonlinearity to match the distribution of the sources.

The marginal entropies in the first term are defined by,

\[
H(\hat{x}_i) = -E \log p(\hat{x}_i)
\]

which suggests the approximation,

\[
\sum_{i=1}^{n} H(\hat{x}_i) \approx \sum_{i=1}^{n} \left( \frac{1}{N} \sum_{k=1}^{N} -\log p_i(x_{k,i}) \right) = \frac{1}{N} \sum_{k=1}^{N} \sum_{i=1}^{n} f_i(\hat{x}_{k,i})
\]

Define \(f(\hat{x}_k) \equiv \sum_{i} f_i(\hat{x}_{k,i})\). If we assume that the marginal distributions belong to a certain class of \(P\) of probability distributions, then \(f\) belongs to a certain class \(F\) of separable functions. The marginal entropy is minimized when the assumed marginal distribution matches the true marginal distribution, assuming the data was in fact generated by the model, so that minimizing the entropy becomes a problem of optimizing over \(A\) and \(f\),

\[
\hat{A}_{ICA} = \arg \min_{A} \min_{f \in F} \left( \min_{\{x_k\}} \sum_{k=1}^{N} f(x_k) \text{ s.t. } H(\hat{x}) \approx H(y) \right)
\]

If we assume independent Gaussian components with equal variance, then \(\hat{A}_{ICA}\) is equivalent to \(\hat{A}_{PCA}\) discussed in section \??.

Optimization of the function \(f\), particularly using kernel methods, is an area of current research but is not discussed in the present report. We concentrate on the case of assumed functional forms both in the optimization and statistical frameworks. The assumed forms will generally be non-quadratic and non-gaussian, but we use certain sequential quadratic or Gaussian approximations to optimize. The theory developed in section 2 gives insight into the effect of the choice of \(f\) on the learned basis. We give a theoretical characterization of certain ordered classes of functions based on generalized convexity, and use
this define the concept of strong sub- and super-gaussianity for densities. Convergence of the algorithms is seen directly according the whether the functions or densities satisfy a certain differential inequality.

The organization of the report is as follows. In section 2, we describe the theory of generalized convexity and strong super-gaussianity, which is important in considering the convergence of algorithms for component estimation. We also examine the relationship between the generalized convexity criterion and the more traditional kurtosis criterion for super-gaussianity. In section 3 we show how this theory can be used to derive globally convergent algorithms for estimation of independent super-gaussian components with a given basis, both in the optimization framework and in the statistical framework. In section 4 we derive a Lagrangian fixed point algorithm for learning normalized components and give examples of applying this algorithm to toy problems and to learning basis vectors from images. In section 5 we describe variational algorithms for component estimation based on variational bounding technique of [11] and the evidence framework [7, 12]. We have recently extended these algorithms to the nonlinear component case described earlier, but due to time and space constraints this extension only mentioned briefly in this report.

2 Generalized convexity and super-gaussianity

According to [13], the foundations of the theory of convex functions were laid by Jensen in 1905-6, though it is noted that Hölder had earlier considered inequalities essentially similar to those defined by Jensen as convex. The basic inequality defining convex functions is the following,

$$\varphi(\mathbb{E}X) \leq \mathbb{E} \varphi(X)$$

which, for invertible \(\varphi\) can also be expressed as,

$$\mathbb{E}X \leq \varphi^{-1}(\mathbb{E} \varphi(X))$$

The term \(\varphi^{-1}(\mathbb{E} \varphi(X))\) can be seen as a generalized mean value [14],[13, ch.3]. The \(L_p\) norms \((\mathbb{E}|X|^p)^{1/p}\) for example fit this framework. The notion of convexity was generalized by B. Jessen in [14] to compare two functions in terms of the means defined by them. An increasing function \(\varphi\) is defined to be convex with respect to another increasing function \(\psi\) if,

$$\psi^{-1}(\mathbb{E} \psi(X)) \leq \varphi^{-1}(\mathbb{E} \varphi(X))$$

This can be written,

$$\varphi \circ \psi^{-1}(\mathbb{E} \psi(X)) \leq \mathbb{E} \varphi \circ \psi^{-1}(\psi(X))$$

which shows that \(\varphi\) is convex with respect to \(\psi\) if and only if \(\varphi \circ \psi^{-1}\) is convex. The latter formulation, used in [14, 13, 15, 16], does not require \(\varphi\) to be invertible. This formulation is not antisymmetric and thus does not define a
partial ordering. Here we shall use an independently derived formulation that is similar, but is anti-symmetric and defines a partial ordering, while not requiring invertibility of either function in the relation.

We now give an informal derivation of generalized convexity from a different perspective. Let $f : \mathbb{R} \to \mathbb{R}$ be increasing on the interval $(a, b)$. The basic criterion for convexity of $f$ is $f(\alpha x + \bar{\alpha} y) \leq \alpha x + \bar{\alpha} y$ for all $x, y \in (a, b), 0 \leq \alpha \leq 1$. Intuitively, this criterion asserts that for any two points $x$ and $y$ in $(a, b)$, the function value at all intervening points is less than the value of the linear function defined to match the value of $f$ at the points $x$ and $y$. In the intervals $(a, x)$ and $(y, b)$, the value of the convex function $f$ will be greater than that of the linear function [17]. Analytically, we have $f$ convex on $(a, b)$ if,

$$f(y) \leq f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0} (y - x_0) \quad \forall x_0 \leq y \leq x_1$$

The inequality is reversed for $y$ in $(a, x_0)$, or $(x_1, b)$. Letting $x_1 \to x_0 \equiv x$, we have the ordinary definition of convexity for differentiable $f$,

$$f(y) \geq f(x) + f'(x)(y - x) \quad \forall x, y \in (a, b)$$

Thus convexity of a function on an interval can be seen as a relationship between the function and a linear model of the function based on the function value and first derivative.

Let $g : \mathbb{R} \to \mathbb{R}$ be strictly increasing on $(a, b)$. From considerations similar to those given in the linear case, we can define $f$ to be convex with respect to $g$ on $(a, b)$ if a model of $g$ using an affine transform of $f$, given by $\alpha f + \beta$, defined so that $f$ and $g$ are equal at two given points, behaves in a manner similar to the line in the convex case (see Figure 1.)

For any three points $x_0 < y < x_1$ in the interval $(a, b)$, we then have,

$$f(y) \leq f(x_0) + \frac{f(x_1) - f(x_0)}{g(x_1) - g(x_0)} (g(y) - g(x_0)) \quad \forall y \in (x_0, x_1)$$

with the inequality reversed in $(a, b)$ outside of $(x_0, x_1)$. Again letting $x_1 \to x_0 \equiv x$, we have $f$ convex relative to $g$ on $(a, b)$ if,

$$f(y) \geq f(x) + \frac{f'(x)}{g'(x)} (g(y) - g(x)) \quad \forall x, y \in (a, b)$$

(7)

It can be verified that (7) is equivalent to the differential definition of the convexity of the composite function $f \circ g^{-1}$ on the interval $(g(a), g(b))$. Thus this formulation agrees with the generalized mean value formulation.

If $f$ and $g$ are twice differentiable on $(a, b)$, we can use the second derivative criterion for convexity to derive a simple criterion for $f$ convex with respect to $g$. It can be verified that the condition,

$$D^2[f \circ g^{-1}](x) \geq 0 \quad \forall x \in (g(a), g(b))$$
Figure 1: (a) A convex function “bends upward” with respect to a linear function when defined to have the same value at two distinct points. (b) When defined to have the same value and the same slope at a point, a convex function lies completely above a linear function. (c) $x^4$ (red) is convex with respect to $x^2$ (blue), and thus $\alpha x^4 + \beta$ “bends more” than $\gamma x^2 + \delta$ when the functions are equated at two distinct points. (c) When the two functions are set to have the same value and the same slope at a point, the quartic lies entirely above the quadratic.
for $f$ and $g$ increasing, is equivalent to,
\[
\frac{f''(x)}{f'(x)} \geq \frac{g''(x)}{g'(x)} \quad \forall x \in (a, b)
\]

These facts were rediscovered in the course of our research, but have been known for some time. The second differential criterion can be found in [18, 19].

Various other related generalizations of convexity have been proposed. Another line of development starts from the theory of convexity of higher order considered in Nörlund [20] and E. Hopf [21] and generalized by Popoviciu [22]. This line was continued by Karlin [23, 24]. The emphasis in these developments is on systems of functions especially for use in approximation theory. The generalization of convexity given in [14] can be considered as a special case of this theory. Beckenbach [25] considers a further generalization of convexity as a relationship between functions.

### 2.1 Relative convexity

For invertible $g$, $f$ is defined to be convex with respect to $g$ if $f \circ g^{-1}$ is convex on $(g(a), g(b))$ [14, 13, 15, 16]. As mentioned, this does not in general imply that $g$ is concave with respect to $f$. A slightly modified definition allows us to define a reciprocal relation that induces a partial ordering, and is not limited to invertible $g$. We refer to the modified definition as relative convexity.

**Definition 1.** $f$ is convex relative to $g$ if there exists a function $h$ that is convex and strictly increasing on the range of $g$ and such that $f = h(g)$. $f$ is concave relative to $g$ if $g$ is convex relative to $f$.

It is immediate that $f$ is concave relative to $g$ iff $h$ is strictly increasing and concave. Since $f = h(g)$ with $h$ strictly increasing, $f$ and $g$ must be increasing and decreasing on the same intervals. We shall say that such functions are isotonic, and are members of the same isotonic class. It is thus necessary for $f$ and $g$ to be isotonic for $f$ to be convex or concave relative to $g$. If $f$ and $g$ are affinely related with positive multiplier, i.e. $\exists \alpha > 0, \beta \in \mathbb{R}$ such that $g = \alpha f + \beta$, then we shall say that $f$ and $g$ are of the same type. A function type defines an equivalence class. Functions with the same type are both convex and concave with respect to each other.

We now show that this relation induces a partial ordering on isotonic classes. Let $f$ convex with respect to $g$ be denoted $f \succ g$, with $f$ concave relative to $g$ denoted $f \prec g$.

**Theorem 1 (Partial Ordering).** The relative convexity relation induces a partial ordering of function types on isotonic classes.

**Proof.** We show that the relation is reflexive, antisymmetric, and transitive.

Reflexivity: $\forall f$ we have $f = h(f)$ where $h$ is the identity function, which is strictly increasing and convex. Thus $f \prec f$.

Antisymmetry: Suppose $f \succ g$ and $f \prec g$. Then $f = h_1(g)$ with $h_1$ strictly
increasing increasing and convex, and \( f = h_2(g) \) with \( h_2 \) strictly increasing and concave. Thus \( h_1 = h_2 \) must be linear (with strictly positive slope), so that \( f \) and \( g \) are of the same type.

Transitivity: Suppose \( f \succ g \) and \( g \succ w \). Then \( f = h_1(g) \) and \( g = h_2(w) \) with \( h_1 \) and \( h_2 \) strictly increasing and convex on their respective domains. Thus \( f = h_1(h_2(w)) \), and since composition of strictly increasing convex functions produces another strictly increasing convex function, \( h_1 \circ h_2 \) is increasing and convex and \( f \succ w \).

Relative convexity is transitive, but comparability under the relation is not. Thus even though we may have \( f \prec h \) and \( g \prec h \), this does not imply that \( f \) can be ordered with respect to \( g \). If, however, \( f \prec h \) and \( h \prec g \), then \( f \prec g \) by transitivity.

The following theorem gives an equivalent criterion that is useful in proving properties of the relation without reference to the function \( h \).

**Theorem 2.** \( f \) is convex relative to \( g \) on \((a, b)\) iff \( \forall x \in (a, b) \exists \lambda \in [0, \infty) \) such that,

\[
f(y) - f(x) \geq \lambda(g(y) - g(x)) \quad \forall y \in (a, b)
\]

(9)

**Proof.** If \( f = h(g) \) and \( h \) is strictly increasing and convex on the range of \( g \), then from the ordinary definition of convexity, we have \( \forall x \in (a, b) \exists \lambda > 0 \) such that,

\[
f(y) - f(x) = h(g(y)) - h(g(x)) \geq \lambda(g(y) - g(x)) \quad \forall y \in (a, b)
\]

(10)

Conversely, suppose (9) holds and define the function \( h \) that maps the range of \( g \) to the range of \( f \) by mapping \( g(x) \) to \( f(x) \) for each \( x \in (a, b) \). To show that this defines a single valued mapping, i.e. a function, we show that (9) implies that if \( g(x) = g(y) \), then \( f(x) = f(y) \). If \( g(x) = g(y) \), then we have \( f(y) - f(x) \geq \lambda(x) \cdot 0 = 0 \), and \( f(x) - f(y) \geq \lambda(y) \cdot 0 = 0 \), so that \( f(x) = f(y) \). Thus all \( x \) values that map to the same \( g \) also map to the same \( f \), so that there is a unique value in range of \( f \) associated with each value in the range of \( g \). Then (9) shows that \( h \) is convex on the range of \( g \). And since (9) also implies that if \( g(y) > g(x) \), then \( f(y) > f(x) \) or \( h(g(y)) > h(g(x)) \), we also have that \( h \) is strictly increasing.

If \( f \) and \( g \) are differentiable and \( f \succ g \), then \( \lambda = h'(g(x)) \). Since \( f(x) = h(g(x)) \), we have \( f'(x) = h'(g(x))g'(x) \), or \( h'(g(x)) = f'(x)/g'(x) \). Thus for \( f \succ g \), we have,

\[
f(y) - f(x) \geq \frac{f'(x)}{g'(x)}(g(y) - g(x)) \quad \forall y \in (a, b)
\]

(11)

For \( f \prec g \),

\[
f(y) - f(x) \leq \frac{f'(x)}{g'(x)}(g(y) - g(x)) \quad \forall y \in (a, b)
\]

(12)
A definition similar to (9) is given in [14], with $\lambda$ allowed to be negative. We require $\lambda$ to be positive to ensure that $f$ is convex relative to $g$ if and only if $g$ is concave relative to $f$. This also has the effect of ensuring that the subdifferential of the interior of range of $g$ contains only strictly positive $\lambda$, which allows us to define the partial ordering. It is immediate from Theorem 2 that $f \prec g$ if and only if $-f \succ -g$.

As a simple example of the use of Theorem 2, we prove the following lemma on addition.

Lemma 1. (a) If $f, g \prec h$, then $f + g \prec h$. Likewise if $f, g \succ h$, then $f + g \succ h$.

(b) If $f \prec g$, then $f \prec f + g \prec g$.

Proof. (a) From the definition, we have $f(y) - f(x) \leq \lambda(h(y) - h(x))$ and $g(y) - g(x) \leq \xi(h(y) - h(x))$ for all $y$, with $\lambda, \xi > 0$. Adding these inequalities, we get $f(y) + g(y) - (f(x) + g(x)) \leq (\lambda + \xi)(h(y) - h(x))$, with $\lambda + \xi > 0$. Thus, $f + g \prec h$. Similarly, $f, g \succ h$ implies $f + g \succ h$. (b) We have $f(y) - f(x) \leq \lambda(g(y) - g(x))$ and $f(y) - f(x) \leq f(y) - f(x)$. Thus $f(y) + g(y) - (f(x) + g(x)) \leq (1 + \lambda)(g(y) - g(x))$, and $f + g \prec g$. Similarly, with $f(y) - f(x) \geq f(y) - f(x)$, we have $f(y) + g(y) - (f(x) + g(x)) \geq (1 + \lambda^{-1})(f(y) - f(x))$, or $f(y) - f(x) \leq \frac{1}{1 + \lambda}(f(y) + g(y) - (f(x) + g(x)))$, and $f \prec f + g$.

Lemma 1a shows that the set of functions convex (or concave) relative to $h$ is a convex cone. The $f$ and $g$ in the definition need not be comparable. Lemma 1b shows that if $f$ and $g$ are comparable, then adding $f$ and $g$ in a sense mixes
the convexity of the functions. We will make this more explicit later. Lemma 1 can easily be extended to integrals over a parameterized set of functions.

The following theorem is important in the application of the theory to estimation.

**Theorem 3.** If \( f, g \succ h \), then \( \log(e^f + e^g) \succ h \).

*Proof.* We have \( f(y) \geq f(x) + \lambda(h(y) - h(x)) \) and \( g(y) \geq g(x) + \xi(h(y) - h(x)) \), and thus,

\[
e^f(y) \geq e^f(x) e^{\lambda(h(y) - h(x))} \quad \text{and} \quad e^g(y) \geq e^g(x) e^{\xi(h(y) - h(x))}
\]

Adding these inequalities and dividing by \( e^f(x) + e^g(x) \), we get,

\[
\frac{e^f(y) + e^g(y)}{e^f(x) + e^g(x)} \geq \left( \frac{e^f(x)}{e^f(x) + e^g(x)} \right) e^{\lambda(h(y) - h(x))} + \left( \frac{e^g(x)}{e^f(x) + e^g(x)} \right) e^{\xi(h(y) - h(x))}
\]

\[
\geq e^{\gamma(h(y) - h(x))}
\]

where \( \gamma = (\lambda \exp f(x) + \xi \exp g(x))/[\exp f(x) + \exp g(x)] \geq 0 \), and the second inequality follows from the convexity of the exponential function. Taking the logarithm of both sides gives the desired result. \( \square \)

If \( f \) and \( g \) are twice differentiable, then a simple criterion can be derived for \( f \succ g \).

**Theorem 4.** If \( f \) and \( g \) are twice differentiable on \( (a, b) \), then,

\[
f \succ g \quad \text{iff} \quad \frac{f''}{|f'|} \geq \frac{g''}{|g'|}
\]

*Proof.* (a) Let the function \( h \) be defined by \( f = h(g) \). We have \( f \succ g \) if and only if \( h'' \) is non-negative on the range of \( g \). Since,

\[
h''(g(x)) = \frac{|f''(x)|}{g'(x)^2} \left( \frac{f''(x)}{|f'(x)|} - \frac{g''(x)}{|g'(x)|} \right)
\]

we have \( h'' \) non-negative on the range of \( g \) if and only if \( \frac{f''(x)}{|f'(x)|} \geq \frac{g''(x)}{|g'(x)|} \quad \forall x \in (a, b) \).

Theorem 4 provides a measure that can be used to determine the relative convexity ordering of functions. We have \( f \succ g \) on \( (a, b) \) if \( f''/|f'| \geq g''/|g'| \) uniformly on \( (a, b) \). Thus we can compare several functions simultaneously rather than pairwise. The measure can be seen as a sort of curvature measure that is invariant to affine scaling of the function, or invariant over types. We shall refer to \( f''/|f'| \) as the function curvature of \( f \). The function curvature can be related to the geometric curvature of the level curves of \( f \) for multivariate functions.

The following theorem gives a relationship between the convexity of convex and concave functions to their Fenchel-Legendre conjugates.
Theorem 5. $f \succ x^2$ if and only if $f^* \prec \phi^2$. Also, $f \prec \log$ if and only if $f^* \succ \log$.

Proof. For simplicity suppose $f$ is twice differentiable, though the result holds in the non-differentiable case as well. We have $f \succ x^2$ if and only if $f''(x)/f'(x) \geq 1/x$ for $x \in (0, \infty)$. Since under our assumptions $f^*$ is twice differentiable, we have $f \succ x^2$ if and only if $f''(x)/f'(x) \geq 1/x$ for $x \in (0, \infty)$. But this holds if and only if,

$$\frac{f''(\phi)}{f'(\phi)} = \frac{1}{x(\phi)f''(x(\phi))} \leq \frac{1}{f'(x(\phi))} = \frac{1}{\phi}$$

that is, if and only if $f^* \prec \phi^2$.

Similarly, $f \prec \log$ if and only if $f''(x)/f'(x) \leq -1/x$ for $x \in (0, \infty)$. But this holds if and only if,

$$\frac{f''(\phi)}{f'(\phi)} = \frac{1}{x(\phi)f''(x(\phi))} \geq -\frac{1}{f'(x(\phi))} = \frac{1}{\phi}$$

that is, if and only if $f^* \succ \log$.

Thus the two self dual functions $\frac{1}{2}x^2$ and $e \log x$ are central in a sense among convex and concave functions respectively. This is similar to the result that an increasing function $f$ is convex if and only if $f^{-1}$ is concave. These topics are further developed in [26].

The symmetry about $\frac{1}{2}x^2$ implies that if $p(x)$ is strongly sub-gaussian (defined below), then the dual problem will be equivalent to estimation with a strongly super-gaussian density, and thus amenable to the results given here.

3 Super-gaussianity and estimation in the linear model

We now give an application of the theory to estimation of the weights in the linear model $y = Ax + \nu$, when $x$ is a vector of independent super-gaussian random variables. The estimation of components with a known basis $A$ is important because algorithms for estimation basis vectors are built around algorithms for estimating components.

A common assumption on or requirement of component densities is that they be super-gaussian. The most widely used criterion for super-gaussianity is positive kurtosis. In applying the relative convexity theory to estimation, we use a stronger criterion given in the following.

Definition 2. Let $p(x)$ be a continuous, symmetric, unimodal density. We say that $p$ is strongly super-gaussian if $-\log p \prec x^2$, and strongly sub-gaussian if $-\log p \succ x^2$.

This is equivalent to the definition of super-gaussianity given in [27, p. 60-61] and cited in [28] as the proper definition of super-gaussianity, though we
have not found it used outside of [27]. The definition used there is given in terms of $f(x) = -\log p(x)$ requiring the $f'(x)/x$ be increasing or decreasing. Differentiating $f'(x)/x$ gives the criterion in Theorem 4. The definition we give is more general than that in [27] as it does not require differentiability.

Strong super-gaussianity is shown to be sufficient for positive kurtosis in the following theorem.

**Theorem 6.** If $p(x)$ is strongly super-gaussian, then $p(x)$ has positive kurtosis.

**Proof.** See [26].

Examples of strongly super-gaussian densities include,

1. Generalized Gaussian, $\exp(-\gamma |x|^p)$, for $p \leq 2$
2. Logistic, $\frac{d}{dx} (1 + \exp(-x))^{-1}$, hyperbolic tangent, $\frac{d}{dx} \tanh(x)$
3. Student’s $t$
4. Cauchy, $(1 + x^2)^{-1}$
5. Symmetric $\alpha$-stable densities, $F(\exp(-|\omega|^{\alpha}))$ for all $0 < \alpha \leq 2$

The first 4 can be shown to be strongly super-gaussian using Theorem 4 with $f(x) = -\log p(x)$. Strong super-gaussianity of the $\alpha$-stable densities follows from Theorem 3 and the fact that symmetric $\alpha$-stable densities can be represented as integrals over the variance parameter of a Gaussian kernel (they are Normal scale mixtures) [29].

### 3.1 MAP estimation of components: Generalized FOCUSS and Half-quadratic algorithms

Consider the MAP estimate of the sources in the non-gaussian linear model $y = Ax + \nu$ when $x$ and $\nu$ are zero-mean, componentwise independent, and strongly super-gaussian, and $A$ is a known $m \times n$ matrix.

$$\hat{x}_{MAP} = \arg \max_x p(x|y) = \arg \min_x -\log p(x) - \log p(y|x)$$

$$= \arg \min_x f(x) + d(y - Ax)$$

We derive a globally convergent descent algorithm for solving this problem using relative convexity and the related concept of generalized convex duality.

#### 3.1.1 Generalized FOCUSS algorithm

We can reformulate the MAP estimation problem as follows,

$$\min_{x,e} f(x) + d(e) \ s.t. \ Ax + e = y$$
Assuming independent components and noise (a similar algorithm can be given for Gaussian noise with known covariance), the problem can be written,

\[ \hat{x}_{\text{MAP}} = \arg \min_x \sum_{i=1}^n f_i(x_i) + \sum_{j=1}^m d_j(e_j) \text{ s.t. } Ax + e = y \]  

(13)

If we define \( \tilde{A} \equiv [A \ I] \) and \( \tilde{x} \equiv [x^T \ e^T]^T \), we have

\[ \hat{x}_{\text{MAP}} = [x_{\text{MAP}}^T \ e^T]^T = \arg \min_{\tilde{x}} \sum_{i=1}^{n+m} \tilde{f}_i(\tilde{x}_i) \text{ s.t. } \tilde{A} \tilde{x} = \tilde{y} \]

where we define \( \tilde{f}_i \) to range over the \( f_i \) and \( d_j \) functions, each of which is assumed square-concave. \( \tilde{A} \) is overcomplete and full rank regardless of the dimension and rank of \( A \). Thus in the following, we concentrate on the overcomplete case \((m < n)\), \( \hat{x} = \arg \min_x \sum_{i=1}^n f_i(x_i) \text{ s.t. } Ax = y \).

By assumption, we have each \( f_i \prec x^2 \). Thus for arbitrary \( z \),

\[ f_i(z) - f_i(x) \leq \frac{1}{2} \frac{f_i'(x)}{x} (z^2 - x^2) \]

We can use this inequality to define a descent algorithm as follows. Given a current estimate \( x \), for arbitrary \( z \) we have,

\[ f(z) - f(x) = \sum_{i=1}^n f_i(z_i) - f_i(x_i) \]
\[ \leq \frac{1}{2} \sum_{i=1}^n \frac{f_i'(x_i)}{x_i} (z_i^2 - x_i^2) \]
\[ = \frac{1}{2} z^T \Pi(x) z - \frac{1}{2} x^T \Pi(x) x \]  

(14)

where \( \Pi(x) \) is a diagonal matrix whose \( i \)th diagonal component is,

\[ [\Pi(x)]_{i,i} = \frac{f_i'(x_i)}{x_i} \]  

(15)

Thus if we update \( x \) according to,

\[ x_{\text{new}} = \arg \min_x x^T \Pi(x_{\text{old}}) x \text{ s.t. } Ax = y \]  

(16)

we can guarantee that right side of (14) is negative, and thus \( f(x_{\text{new}}) \leq f(x_{\text{old}}) \). The algorithm is formulated explicitly in Appendix B.

3.1.2 Generalized convex duality: Half-quadratic algorithm

In this section we derive the algorithm defined by (16) from a different perspective. This approach is essentially a “half-quadratic” algorithm [2] as used in
the image restoration literature. The derivation is based on a notion of generalized convex duality arising from the relative convexity relation. The theory described here was derived independently in the course of this work, but the idea was used earlier in the image restoration literature [2, 3] and in the machine learning literature [30, 31].

The conjugate of a function is defined by
\[
\phi^*(\phi) = \sup_x \phi(x) - h(x)
\]
for convex \( h \), and
\[
\phi^*(\phi) = \inf_x \phi(x) - h(x)
\]
for concave \( h \). If the optima are attained at \( \bar{x} \) and \( \bar{\phi} \), we have
\[
\bar{x} = h^*'(\bar{\phi}) \quad \text{and} \quad \bar{\phi} = h'(\bar{x}) \quad (17)
\]

There is thus a “duality” between the argument \( x \) and the slope \( \phi \), where \( x \) is both the argument of \( h \) and also the slope of \( h^* \) evaluated at \( h'(x) \). Similarly \( \phi \) is both the argument of \( h^* \) and the slope of \( h \) evaluated at \( h^*(\phi) \).

We can define a generalized duality as follows. If \( f \prec g \) then there is a concave increasing function \( h \) such that
\[
f(x) = h(g(x)) = \inf_{\phi} g(x) - h^*(\phi)
\]
where \( h^* \) is given by,
\[
h^*(\phi) = \inf_x \phi(x) - h(g(x)) = \inf_x \phi(x) - f(x)
\]

This can be used in optimization as follows. Suppose we are given the problem \( \min_{x \in C} f(x) \) which we cannot solve directly, but we are able to solve \( \min_{x \in C} g(x) \), where \( f \prec g \). Then we can write,
\[
\min_x f(x) = \min_x \min_{\phi} \phi g(x) - h^*(\phi)
\]
We then solve this problem by coordinate descent in \( x \) and \( \phi \). Given \( \bar{x} \), we find,
\[
\bar{\phi} = \arg \min_{\phi} \phi g(\bar{x}) - h^*(\phi) \quad (18)
\]
h is increasing and concave, so \( h^* \) is increasing and concave. If \( h \) is differentiable, then \( \bar{\phi} \) satisfies,
\[
h^*(\bar{\phi}) = g(\bar{x})
\]
and thus from (17), we have,
\[
\bar{\phi} = h'(g(\bar{x})) = \frac{f'(\bar{x})}{g'(\bar{x})}
\]
Thus to solve to solve \( \min_{x \in C} f(x) \), we use the following algorithm. Given \( x_{\text{old}} \), we set,
\[
x_{\text{new}} = \arg \min_{x \in C} \left( \frac{f'(x_{\text{old}})}{g'(x_{\text{old}})} \right) g(x)
\]
Letting \( g(x) = x^2 \) and letting \( C \) be the linear variety defined by \( Ax = y \), we have the algorithm defined by (16).
For strongly super-gaussian priors that are also negative log-convex, the MAP estimate is unique and this iteration defines a descent algorithm that will converge to the global optimum. If we are interested in finding maximally sparse components, however, negative log-concave priors are more appropriate. Unfortunately, these priors introduce the problem of local optima. This problem can be mitigated somewhat by using a variational approximation to the posterior that is valid for strongly super-gaussian functions, as described in the next section.

3.2 Variational bounding approach

As mentioned, when the prior on \( x \) is negative log concave, then the problem,

\[
\arg \max_x p(x|y) = \arg \min_x d(y - Ax) + f(x)
\]

will have multiple local optima since \( f \) is a concave function. One approach to dealing with this problem is to use a variational approximation to \( p(x|y) \) that is Gaussian. Even though \( p(x) \) is assumed to be very non-gaussian, \( p(x|y) \) may still be approximated well by a Gaussian. That is, the observation \( y \) may determine the subspace in which \( x \) lies, and within that subspace, \( x \) may be Gaussian distributed. For an observation from a random subspace, however, if there are many components, a given component is not likely to be active, so that marginally the components will have very super-gaussian densities.

First, consider the ordinary Maximum Likelihood estimate of the prior component variances \( \xi = [\xi_1 \ldots \xi_n]^T \) in the Gaussian linear model,

\[
\hat{\xi} = \arg \max_\xi p(y; \xi) = \arg \max_\xi \int p(y|x) p(x; \xi) \, dx
\]

where \( A = \text{diag}(\xi) \). We can define an EM algorithm for estimating \( \xi \) by treating the \( x \) as hidden or latent variables. The complete log likelihood is then,

\[
- \log p(y, x; \xi) = \frac{1}{2} x^T A^T \Sigma^{-1} A x - y^T \Sigma^{-1} A x + \frac{1}{2} \log |\Lambda| + \frac{1}{2} x^T \Lambda^{-1} x + \text{const.}
\]

In the EM algorithm we take the expectation of the complete log likelihood with respect to the posterior distribution using the current parameters \( \xi^{(k)} \), where \( k \) denotes the iteration. For the posterior, we have,

\[
p(x|y; \xi) = \frac{p(y|x) p(x; \xi)}{p(y)} = \mathcal{N}(\mu_x, \Sigma_x)
\]

where,

\[
\mu_x = \Lambda A^T (\Lambda A A^T + \Sigma)^{-1} y, \quad \Sigma_x = (A^T \Sigma^{-1} A + \Lambda^{-1})^{-1}
\]
Taking the expectation of the terms in the complete log likelihood that involve the parameter $\xi$, we have,

$$E_{\mathbf{x}|\mathbf{y};\xi^{(k)}} \left[ \sum_{i=1}^{n} \log \xi_i + \frac{x_i^2}{\xi_i} \right] = \sum_{i=1}^{n} \log \xi_i + \frac{E_{\mathbf{x}|\mathbf{y};\xi^{(k)}}[x_i^2]}{\xi_i}$$

Then, minimizing with respect to $\xi_i$, we have,

$$\xi_i^{(k+1)} = E_{\mathbf{x}|\mathbf{y};\xi^{(k)}}[x_i^2] = [\mu_{\mathbf{x}}]^2_i + [\Sigma_{\mathbf{x}}]_{i,i}$$  \hspace{1cm} (22)

Thus the EM algorithm consists of updating the mean and covariance of the posterior according to (21) with $\Lambda = \text{diag}(\xi^{(k)})$, and updating the variance parameters according to (22).

We can generalize this algorithm for non-gaussian component priors by considering a variational form of the prior, and using an EM-type algorithm to estimate the variational parameters.

For a strongly super-gaussian prior $p(x) = \exp(-f(x))$, we have $f < x^2$, so that there exists $h$ concave and increasing on $(0, \infty)$ such that $f(x) = h(x^2)$. By definition of the concave conjugate of $h$, we have,

$$f(x) = h(x^2) = \inf_{\phi} \phi x^2 - h^*(\phi) = \inf_{\xi} \frac{1}{2} x^2 - h^*(\frac{1}{2\xi})$$

and thus,

$$p(x) = \exp(-f(x)) = \sup_{\xi} \exp\left(\frac{1}{2} x^2 \xi^{-1}\right) \exp\left(h^*(\frac{1}{2\xi})\right)$$

$$= \sup_{\xi} \mathcal{N}(x; 0, \xi) \varphi(\xi)$$  \hspace{1cm} (23)

where,

$$\varphi(\xi) = \sqrt{2\pi\xi} \exp\left(h^*(\frac{1}{2\xi})\right)$$

Thus we have,

$$\max_{\mathbf{x}} p(\mathbf{x}|\mathbf{y}) = \max_{\xi} \frac{p(\mathbf{y}|\mathbf{x}) p(\mathbf{x})}{p(\mathbf{y})} p(\mathbf{y}) \prod_{i=1}^{n} \varphi(\xi_i)$$

where $\Lambda = \text{diag}(\xi)$. The variational bounding EM-type algorithm treats this last expression as a complete likelihood of $(\mathbf{y}, \mathbf{x})$ with $\xi$ a parameter vector of the distribution to be estimated,

$$L(\mathbf{y}, \mathbf{x}; \xi) = \frac{p(\mathbf{y}|\mathbf{x}) \mathcal{N}(\mathbf{x}; 0, \Lambda)}{p(\mathbf{y})} \prod_{i=1}^{n} \varphi(\xi_i)$$
The algorithm performs an EM-type iteration to minimize the negative log of this “likelihood” with respect to the “parameter” vector $\xi$.

\[
-\log L(y, x; \xi) = \frac{1}{2} \|y - Ax\|_{\Sigma^{-1}}^2 + \frac{1}{2} x^T \Lambda^{-1} x + \frac{1}{2} \log |\Lambda| - \sum_{i=1}^{n} \log \varphi(\xi_i) + \text{const.}
\]

Taking the expected value with respect to the approximate posterior $N(\mu_x, \Sigma_x)$ and minimizing with respect to $\xi$, we have,

\[
\frac{1}{2} \xi = h'(E_{x|y, \xi(\xi)}[x_i^2])
\]

Let $\sigma^2_{x_i} = E_{x|y, \xi(\xi)}[x_i^2]$. Then,

\[
\frac{1}{2} \xi = h'(\sigma^2_{x_i}) = \frac{f'(\sigma_{x_i})}{2\sigma_{x_i}}
\]

and,

\[
[\Lambda^{-1}]_{i,i} = \frac{f'(\sigma_{x_i})}{\sigma_{x_i}}
\]

This exhibits more clearly the similarity between $\Lambda^{-1}$ and the matrix $\Pi(x)$ from the Generalized FOCUSS algorithm (15), which is noted in [11]. According to (22), we have,

\[
\sigma^2_{x_i} = \mu_{x_i}^2 + [\Sigma_x]_{i,i}
\]

The Generalized FOCUSS algorithm, on the other hand, can be seen as using $\sigma^2_{x_i} = [\mu_{x_i}]^2$, neglecting the variance term. It should be noted, however, that this derivation is rather heuristic as the EM analysis does not apply for the non-normalized approximate density. In the next section, we look at super-gaussian priors that can be written as variance mixtures of Gaussian, allowing a more rigorous formulation of the algorithm.

### 3.3 Variational evidence approach: automatic relevance and hyperprior algorithms

Suppose the prior $p(x)$ can be written

\[
p(x) = \int p(x|\xi)p(\xi) \, d\xi = \int_0^\infty N(x; 0, \xi)p(\xi) \, d\xi
\]

so that we can imagine an auxiliary random variable $\xi$ such that given $\xi$, $x$ is conditionally Gaussian with variance $\xi$. Now consider MAP estimation of the vector $\xi$,

\[
\hat{\xi} = \arg \max_{\xi} \log p(\xi|y)
\]
We can define an EM-type algorithm to find a locally optimal \( \xi \). Using the fact that,

\[
\log p(\xi | y) = E_{x|y} \log \frac{p(\xi, x | y)}{p(x | \xi, y)}
\]

we have,

\[
\log p(\xi^{(k+1)} | y) - \log p(\xi^{(k)} | y) = E_{x|y} \log p(\xi^{(k+1)}, x | y) - E_{x|y} \log p(\xi^{(k)}, x | y)
\]

\[
+ D \left( p(\xi^{(k)}, x | y) \ || \ p(\xi^{(k+1)}, x | y) \right)
\]

Thus we can treat \( \log p(\xi, x | y) \) as a complete log-likelihood, analogous to that in the standard EM algorithm, and minimize \( E_{x|y} \log p(\xi, x | y) \) with respect to \( \xi \), which will usually be possible in closed form.

The posterior \( p(x | \xi, y) \) is given by,

\[
p(x | \xi, y) = \frac{p(y | x, \xi) p(x | \xi)}{p(y | \xi)} = \mathcal{N}(x; \mu_x, \Sigma_x)
\]

where again, \( \mu_x = \Lambda A^T (\Lambda A A^T + \Sigma)^{-1} y \), \( \Sigma_x = (\Lambda^T \Sigma^{-1} \Lambda + \Lambda^{-1})^{-1} \), and \( \Lambda = \text{diag}(\xi) \). For the complete log-likelihood term, we have,

\[
- \log p(\xi, x | y) = - \log p(y | x) - \log p(x | \xi) - \log p(\xi) + \text{const.}
\]

\[
= \frac{1}{2} \| y - Ax \|_\Sigma^{-1}^2 + \frac{1}{2} x^T \Lambda^{-1} x + \frac{1}{2} \log |\Lambda| - \log p(\xi) + \text{const.}
\]

\[
= \frac{1}{2} \| y - Ax \|_\Sigma^{-1}^2 + \sum_i \left[ \frac{1}{2} x_i^2 + \frac{1}{2} \log \xi_i - \log p(\xi_i) \right] + \text{const.}
\]

The EM-type algorithm proceeds by alternately updating the mean and covariance of \( p(x | \xi, y) \), and setting \( \xi_i^{(k+1)} \) to satisfy,

\[
\xi_i + 2 \xi_i \frac{p(\xi_i)}{p(\xi)} = E_{x|\xi_i,y} [x_i^2]
\]

As the hyperprior becomes non-informative (flat), the second term tends to zero, and the algorithm becomes a standard EM algorithm for estimating the (non-random) variance of the hyperprior. In the case of random \( \xi \), the EM-type iteration still guarantees an increase the posterior likelihood \( p(\xi | y) \). The key to this is algorithm is writing the prior \( p(x) \) as a scale mixture of Normals. According to the integral form of Lemma 2, if \( f_t \prec h \) for all \( t \), then

\[
- \log \int_0^\infty \mathcal{N}(x; 0, \xi) p(\xi) \, d\mu(t) \prec h.
\]

In particular,

\[
- \log \int_0^\infty \mathcal{N}(x; 0, \xi) p(\xi) \, d\xi \prec x^2
\]

for any \( p(\xi) \) such that the integral exists. Thus a density can be written in the form (26) only if it is strongly super-gaussian.
A question naturally arises as to what kind of densities can be written as scale mixtures of Gaussian. One useful tool in this regard is the Mellin transform [32], which owing to a change of variable relationship to the Fourier transform, satisfies a type of convolution theorem. These topics are an area of current research.

As an example, the Laplacian density can be written as conditionally Normal with standard deviation distributed Rayleigh.

\[
\frac{1}{2} \exp(-|x|) = \int_0^\infty N(x;0,\sigma^2)p(\sigma) \, d\sigma
\]

\[
= \int_0^\infty \left[ \frac{1}{\sqrt{2\pi}\sigma} \exp\left( -\frac{1}{2} x^2/\sigma^2 \right) \right] \cdot \left[ \sigma \exp\left( -\frac{1}{2} \sigma^2 \right) \right] \, d\sigma
\]

Deriving formula (27) in terms of \( \sigma = \sqrt{\xi} \), we have,

\[
\sigma_i^2 - \sigma_i^3 \frac{p'(\sigma_i)}{p(\sigma_i)} = E_{x|\sigma(\cdot),y}[x_i^2]
\]

(28)

For the Rayleigh distribution,

\[
\frac{p'(\sigma_i)}{p(\sigma_i)} = \frac{1}{\sigma_i} - \sigma_i
\]

Substituting this in (28), we get,

\[
\sigma_i^4 = \xi_i^2 = E_{x|\sigma(\cdot),y}[x_i^2]
\]

This differs from the case of assuming Gaussian components and estimating the variance parameter described at the beginning of section 3.2 in that the \( \xi^2 \), the square of the variance, is set to the posterior variance, whereas in (22) the \( \xi \) (not \( \xi^2 \)) is set to the posterior variance.

The Laplacian can also be written as conditionally Gaussian with random variance distributed exponentially, which gives a different optimal \( \xi \) given \( x \), so the algorithms are dependent on the form of the conditional Gaussianization.

4 Learning normalized bases: Lagrangian algorithm

In this section we derive an algorithm for normalized basis learning that can be viewed as joint estimation of \( A \) and \( X = [x_1 \ldots x_N] \) where the component and noise distributions are assumed to be independent and super-gaussian. The duality relationship between sub- and super-gaussianity allows the algorithm to be applied in the case of independent sub-gaussian component and noise distributions as well.

The algorithm was motivated by the problem of learning data representations based on a linear generative model [33, 5, 34, 6, 35, 36, 37, 11]. Given observations \( y = [y_1 \ldots y_N] \), the problem is to estimate the parameters \( A \in \mathbb{R}^{m \times n} \).
and \( x = [x_1 \ldots x_N] \) in the Bayesian linear model,

\[ y_k = Ax_k + \nu_k, \quad k = 1, \ldots, N \]  

(29)

assuming that the sources \( x_k \) are independent. The low noise limit is equivalent to the case in which the noise random variables \( \nu_k \) are not present. Since Field [33], much consideration has been given to representations that assume sparse and distributed sources, i.e. many source components with relatively few of the components having significant magnitude, or “active”, at any given time. One way to ensure a sparse representation is to take \( A \) to be “overcomplete”, or have more columns than rows. However, sparse coding can also be carried out when the matrix \( A \) is not overcomplete, for example when the data is high dimensional but occupies a relatively low dimensional manifold [38].

Two main statistical approaches have been used in the estimation of overcomplete \( A \): an ML approach estimating \( A \) by (approximately) marginalizing over \( x \), and a joint maximum à posteriori (MAP) estimation approach estimating both \( A \) and \( x \). Lewicki and Olshausen [34] and Lewicki and Sejnowski [6] propose an approximate ML framework. The likelihood is marginalized over the possible generating sources,

\[ p(Y|A) = \int p(Y, X|A) \, dX \]

and the resulting integral is approximated by a Gaussian density. Various other approximations are made to obtain a generalization of the Infomax algorithm of Bell and Sejnowski [10]. Girolami [11] also uses the ML approach, but employs a variational approximation similar to those used by Jaakkola [30]. An algorithm is derived using a variational bound on the Laplacian (sparse) prior that makes the marginalization tractable. The EM algorithm is used to optimize the variational likelihood over the augmented set of parameters. As the variational parameters converge, the variational approximation approaches Laplacian. This algorithm apparently converges to a local minimum of \( p(Y|A) \) itself rather than an approximation as in [6]. It is also globally convergent via the EM algorithm. As usual, however, the algorithm is subject to convergence to local optima.

Another approach is to find the joint estimate of \( A \) and \( X \). Here we maximize,

\[ p(A, X|Y) = p(Y|A, X) p(A, X) \]

For the optimization to be well defined, a constraint must be put on either \( X \) or \( A \), which amounts to determining \( p(A, x) \) [39, 35, 37]. Olshausen and Field [5] used a cost function equivalent to that implied by the MAP framework. The MAP framework is used explicitly in Hyvärinen [35] for complete and undercomplete \( A \), and in Kreutz-Delgado and Rao [37] for overcomplete \( A \). Algorithms are also found in [36]. The usual idea is to view the log likelihood as consisting of an error term and a sparsity term, and alternately adapt \( X \) and \( A \), with \( X \) adapting to increase the sparsity of the representation, and \( A \) adapting to maintain fidelity of the representation.
Here we employ the MAP framework to estimate $\mathbf{A}$ and $\mathbf{X}$ given $\mathbf{Y}$ in a novel way. Rather than adapting $\mathbf{A}$ to reduce an error term only, thus reducing the sparsity of the representation only indirectly, we derive a general procedure for adapting $\mathbf{A}$ to reduce the source prior term directly in the noiseless case, and show that the noiseless analysis can be applied to the noisy case as well by a change of variables. The noiseless algorithms can be seen as new generalizations of the Infomax algorithm [10] to the overcomplete case, which differ from that given in Lewicki and Sejnowski [6]. The noisy algorithms are similar to the approximate algorithms found in [35] for the complete and undercomplete cases. Our analysis however does not require $\mathbf{A}$ to be invertible, and we derive a descent algorithm for the original cost function, not an approximation. The generality of the approach allows application to the undercomplete dictionary case as well, which may be useful when the intrinsic dimensionality of the data is less than the dimensionality of the observed vectors [38]. For example, an image compression scheme might code 12 pixel $\times$ 12 pixel blocks, but the image blocks that are coded may have a simple structure that is representable in fewer than the 144 basis vectors that result from using even a complete representation, much less an overcomplete representation.

We first derive a descent algorithm, showing that the iterates decrease a negative log likelihood at each iteration. Let $\mathbf{A}$ be uniformly distributed over a compact set $\mathcal{S}_\mathbf{A} \subset \mathbb{R}^{m \times n}$. Let $\mathbf{Y} = [\mathbf{y}_1 \ldots \mathbf{y}_N]$ be observations from the linear model, $\mathbf{y}_k = \mathbf{A}\mathbf{x}_k + \nu_k$, with all random variables in the model independent, zero mean, symmetric, and strongly super-gaussian. We consider the zero noise limit problem (13), with the general problem handled by first transforming the problem as in section 3. We wish to find $\mathbf{A}$ and $\mathbf{X} = [\mathbf{x}_1 \ldots \mathbf{x}_N]$ to minimize $\sum_k f(\mathbf{x}_k) \equiv f(\mathbf{X})$ subject to $\mathbf{A}\mathbf{X} = \mathbf{Y}$, $\mathbf{A} \in \mathcal{S}_\mathbf{A}$.

We develop a descent algorithm for $f(\mathbf{X})$ such that at each iteration we have $\mathbf{A}\mathbf{X} = \mathbf{Y}$, and $\mathbf{A} \in \mathcal{S}_\mathbf{A}$. The iterates of the algorithm shall be denoted by $l = 1, 2, \ldots$, and we denote $\mathbf{A}_l$, $\mathbf{X}_l$, and $\mathbf{X}_{k,l}$ at the $l$th iteration by $\mathbf{A}_l$, $\mathbf{X}_l$, and $\mathbf{X}_{k,l}$ respectively.

Considering the development in section 3, we shall assume that given $\mathbf{A}_{l+1}$, we update $\mathbf{X}$ according to (39), so that,

$$
\mathbf{x}_{k,l+1} = \Pi_{k,l}^+ \mathbf{A}_T ^{l+1} (\mathbf{A}_l \Pi_{k,l}^+ \mathbf{A}_T ^{l+1})^{-1} \mathbf{y}_k
$$

(30)

where $\Pi_{k,l} \equiv W_{k,l}^+ \nabla f(\mathbf{x}_{k,l})$, and $W_{k,l} \equiv \text{diag}(\mathbf{x}_{k,l})$. This update guarantees feasibility of $\mathbf{X}_{l+1}$ at the end of an iteration consisting of updating $\mathbf{A}$ and then updating $\mathbf{X}$, and guarantees that $\mathbf{X}$ moves in a reasonable direction. Although updating the source estimates according to (39) provides descent in the case of known constant $\mathbf{A}$, (30) does not in general reduce $f(\mathbf{x}_k)$, since (39) only guarantees that $f(\mathbf{x}_{k,l+1}) < f(\mathbf{x}_{k,l})$ for $\mathbf{x}_{k,l}$ feasible. After $\mathbf{A}_l$ is updated to $\mathbf{A}_{l+1}$, $\mathbf{X}_l$ is no longer feasible. Thus updating it according to (30) does not alone guarantee descent of $f(\mathbf{X}; \mathbf{A})$. We can, however, guarantee that $f(\mathbf{X}_{l+1}; \mathbf{A}_{l+1}) < f(\mathbf{X}_l; \mathbf{A}_l)$ by exploiting our knowledge that $\mathbf{X}_l$ will change to $\mathbf{X}_{l+1}$ according to (30) after we update $\mathbf{A}_l$ to $\mathbf{A}_{l+1}$. Using the inequality (14)
for square-concave functions, we have for each \( x_k \),
\[
f(x_{k,l+1}) - f(x_{k,l}) \leq \frac{1}{2} x_{k,l+1}^T \Pi_{k,l} x_{k,l+1} - \frac{1}{2} x_{k,l}^T \Pi_{k,l} x_{k,l}
\]
Using (30), for each \( x_k \) we have,
\[
x_{k,l+1}^T \Pi_{k,l} x_{k,l+1} = y_k^T (A_{l+1}^T \Pi_{k,l}^+ A_{l+1}^T)^{-1} y_k
\]
Define the functions,
\[
h_k(A) \equiv y_k^T (A \Pi_{k,l}^+ A^T)^{-1} y_k - x_{k,l}^T \Pi_{k,l} x_{k,l}
\]
and \( h(A) = \sum_k h_k(A) \). Then we have \( f(X_{l+1}) - f(X_l) \leq h(A_{l+1}) \). Now consider the value of \( h \) at \( A_l \). We have, \( h_k(A_l) = y_k^T (A_l \Pi_{k,l}^+ A_l^T)^{-1} y_k - x_{k,l}^T \Pi_{k,l} x_{k,l} = x_{k,l}^T \Pi_{k,l} x_{k,l} - x_{k,l}^T \Pi_{k,l} x_{k,l} \), where \( x_k = \arg \min_x x^T \Pi_{k,l} x \) s.t. \( A_l x = y_k \). Thus,
\[
h(A_l) = \sum_k h_k(A_l) = \sum_k x_{k,l}^T \Pi_{k,l} x_{k,l} - x_{k,l}^T \Pi_{k,l} x_{k,l} \leq 0
\]
since \( A_l x_{k,l} = y_k \) for all \( k \), and \( x_k \) achieves the minimum of \( x^T \Pi_{k,l} x \) for \( A_l x = y \). Thus if we can find \( A_{l+1} \in S_A \) such that \( h(A_{l+1}) < h(A_l) \), then we will have,
\[
f(X_{l+1}) - f(X_l) \leq h(A_{l+1}) < h(A_l) \leq 0
\]
For simplicity, we use the method of gradient descent to decrease \( h \). First suppose \( S_A \) is the sphere of unit Frobenius norm matrices. We could define geodesic gradient descent algorithms to maintain \( A \) in \( S_A \), but a more convenient method is to simply project the gradient of \( A \) evaluated at \( A_l \) onto the subspace orthogonal to the “vector” \( A_l \), i.e. onto the hyperplane tangent to the sphere of constant Frobenius norm at \( A_l \) [37]. Since the projection operator is positive semidefinite, the projected gradient is still a descent direction. For the projected gradient, we have,
\[
A_{l+1} = A_l - \alpha \text{Proj} \left( \frac{\partial h(A_l)}{\partial A} \right)
\]
\[
= A_l - \alpha \left( \frac{\partial h(A_l)}{\partial A} - \frac{\langle \partial h(A_l), A_l \rangle}{\langle A_l, A_l \rangle} A_l \right)
\]
(31)

The partial derivative of \( h_k(A) = y_k^T (A \Pi_{k,l}^+ A^T)^{-1} y_k \) with respect to \( A \) can be found using standard matrix calculus to be,
\[
\frac{\partial h(A)}{\partial A} = -2 \sum_k (A \Pi_{k,l}^+ A^T)^{-1} y_k y_k^T (A \Pi_{k,l}^+ A^T)^{-1} A \Pi_{k,l}^+
\]
(32)
With this we have for the inner product in (31),
\[
\left\langle \frac{\partial h(A_l)}{\partial A}, A_l \right\rangle = \text{tr} \left( \frac{\partial h(A_l)}{\partial A} A_l^T \right) = -2 \sum_k \text{tr} \left( (A_l \Pi_k^T A_l^T)^{-1} y_k y_k^T \right)
\]
\[
= -2 \sum_k y_k^T (A_l \Pi_k^T A_l^T)^{-1} y_k
\]

Define \( \bar{\lambda}_{k,l} = (A_l \Pi_k^T A_l^T)^{-1} y_k \). Then, absorbing the factor of 2 into the parameter \( \alpha \), and using the definition of \( \bar{x}_k = \Pi_k^+ A_l^T \bar{\lambda}_{k,l} \), the update (31) becomes,
\[
A_{l+1} = (1 - \alpha \sum_k \bar{\lambda}_{k,l} y_k) A_l + \alpha \sum_k \bar{\lambda}_{k,l} \bar{x}_k
\]

Or, redefining \( \alpha \), we can write,
\[
A_{l+1} = (1 - \alpha) A_l + \alpha \|A\|^2 F^2 \sum_k \bar{\lambda}_{k,l} \bar{x}_k
\] (33)

which makes setting \( \alpha \) much easier by reducing or eliminating dependence of the step size on the problem size. Note that in this setup we must finish the iteration by updating \( X_l \) according to (30), which amounts to a sort of double iteration on \( \hat{X} \). The iterations, however, must be done as prescribed for the algorithm to perform as stated. For example, we cannot simply iterate (30) twice since the derived descent depends on the two updates being done with different \( \Pi \) parameters (see (35) below).

It may be seen that in the case of complete \( A \), this algorithm is equivalent to the Infomax algorithm of [10], and is thus a generalization of Infomax different from the algorithm given in [6]. The latter algorithm and the algorithm given in [37] may be seen as using alternative estimates of the Lagrange multiplier type vectors \( \lambda_k \). Writing the Lagrangian for the optimization problem under consideration, we have,
\[
L(A, X) = \sum_k \left( f(x_k) + \lambda_k^T (y_k - Ax_k) \right) + \mu(\|A\|_F^2 - 1)
\]

Setting the partial gradients of \( L \) equal to zero suggests (using star to denote optimality),
\[
\lambda_k^* = (A^* \Pi^+ (x_k^*) A^{*T})^{-1} y_k
\]
\[
A^* = \frac{\sum_k \lambda_k^* x_k^{*T}}{\sum_k \lambda_k^* y_k} \quad x_k^* = \Pi^+(x_k^*) A^{*T} \lambda_k^*
\]

Thus, except for the norm multiplier, (33) can be seen as a fixed point Lagrangian algorithm. Indeed, eliminating \( \|A\|_F^2 \) from the expression yields an
algorithm that is shown experimentally to converge to a solution with unit Frobenius norm, though it may temporarily increase the objective function on its way. The iteration given in (33), if initialized with a unit norm matrix, will, as expected, steadily but very slightly increase the norm of $A$, usually coming to a fixed point before increasing. Also as expected it monotonically decreases the objective function for problem-independent step sizes of approximately $10^{-2}$ or $10^{-1}$.

One problem with algorithms of this form for component estimation however, as noted in [6] and elsewhere, is that there is nothing to prevent individual columns from going to zero, reaching a sort of degenerate solution. We can eliminate this problem by constraining $A$ to have unit column norm. In this case the Lagrangian is,

$$L(A, x) = \sum_k \left[ f(x_k) + \lambda_k^T (y_k - Ax_k) \right] + \sum_{i=1}^n \mu_i (a_i^T a_i - 1) \quad (34)$$

where $A = [a_1 \ldots a_n]$. Following the example of the constrained Frobenius norm case, we can derive a fixed point Lagrangian algorithm that converges to an $A^*$ with unit column norms. We follow the same protocol suggested by the globally convergent algorithm. Let $W_{k,l} = \text{diag}(x_{k,l})$. We first calculate the Lagrange multiplier and $\bar{x}$ estimates,

$$\Pi_{k,l} = \text{diag}(W_{k,l} + \lambda_{k,l} \nabla f(x_{k,l})) \quad \bar{\lambda}_{k,l} = (A_l \Pi_{k,l} A_l^T)^{-1} y_k$$

Then update $A$,

$$A_{l+1} = (1 - \alpha) A_l + \alpha \left( \sum_k \bar{\lambda}_{k,l} x_{k,l}^T \right) \text{diag}(\bar{\mu}_l)^{-1}$$

Finally, update the $x_k$ using the new $A$, and (old) $\Pi_{k,l}$,

$$\lambda_{k,l} = (A_{l+1} \Pi_{k,l} A_{l+1}^T)^{-1} y_k \quad x_{k,l+1} = \Pi_{k,l} A_{l+1}^T \lambda_{k,l} \quad (35)$$

The algorithm can also be seen as a Newton method for finding a stationary point of the Lagrangian using sequentially updated estimates of the Lagrange multipliers. The gradient of (34) with respect to $A$ is,

$$- \sum_{k=1}^N \lambda_k x_k^T + A \text{ diag}(\mu) \quad (36)$$

The Hessian operator is $\text{diag}(\mu)$ multiplied on the right, and the inverse Hessian operator is $\text{diag}(\mu)^{-1}$ multiplied on the right. Thus for the Newton direction, given the estimates $\hat{\lambda}_k, \hat{x}_k, k = 1, \ldots, N$ and $\hat{\mu}$, we have,

$$\Delta A = - \left( \sum_{k=1}^N \hat{\lambda}_k \hat{x}_k^T \right) \text{diag}(\hat{\mu})^{-1} + A \quad (37)$$
Thus a general approximate Newton algorithm for learning $A$ is given by,

$$A_{t+1} = (1 - \alpha)A_t + \alpha \left( \sum_{k=1}^{N} \hat{\lambda}_k \hat{x}_k^T \right) \text{diag}(\hat{\mu})^{-1}$$

A current area of research is the use of this algorithm in adaptive environments where $N$ is small.

5 Learning Maximum Likelihood bases: Variational methods

In this section we consider a variational approach to the EM algorithm for estimation in the linear model with independent weights. We generalize and simplify the analysis in [11], which was based on [31].

Consider the problem of Maximum Likelihood estimation of $A$ in the linear model $y = Ax + \nu$ given $N$ independent observations of $y$. We can extend the variational algorithms for estimating components given in from sections 3.2 and 3.3 by simply updating $A$ according to the usual EM update for the Gaussian linear model.

The variational bounding algorithm for estimation of overcomplete bases is considered in [11], based on the variational methods in [30], but the analysis is given only for the Laplacian and is derived in a rather tortuous way, made to seem particular to the Laplacian density rather than employing the abstract properties of strongly super-gaussian densities to derive a general algorithm.

Suppose $p(x)$ is strongly super-gaussian, and let $f = -\log p$. Then using (23), we have,

$$p(y; A) = \int p(y|x; A)p(x) dx \geq \left( \prod_i \varphi(\xi_i) \right) \int y(y(Ax, \Sigma)) N(x; 0, \Lambda) dx = \left( \prod_i \varphi(\xi_i) \right) N(y; 0, \Sigma + AA^T)$$

We can use a variational EM algorithm to maximize this lower bound in a manner similar to the evidence EM algorithm of section 3.2. The only addition is that we update $A$ according to the standard EM update (which maximizes the expected value of the complete log likelihood),

$$A = \left( \sum_{k=1}^{N} y_k \mu_{x_k}^T \right) \left( \sum_{k=1}^{N} (E_{x_k}^T x_k^T) \right)^{-1} = \left( \sum_{k=1}^{N} y_k \mu_{x_k}^T \right) \left( \sum_{k=1}^{N} \Sigma_{x_k} + \mu_{x_k} \mu_{x_k}^T \right)^{-1}$$

where $\mu_{x_k}$ and $\Sigma_{x_k}$ are given by (21) and $A$ is given by (24).
Appendix A: Explicit form of MAP algorithm for component estimation

To formulate explicitly the algorithm for non-gaussian component estimation, let $x_{\text{old}}$ be denoted $\bar{x}$, and denote $x_{\text{new}}$ simply by $x$. The minimization in (16) can be carried out by solving,

$$
\begin{bmatrix}
\Pi(\bar{x}) & A^T \\
A & 0
\end{bmatrix}
\begin{bmatrix}
x \\
\lambda
\end{bmatrix}
= \begin{bmatrix}
0 \\
y
\end{bmatrix}
$$

(38)

We can write $x_{\text{new}}$ in closed form as,

$$
x = \Pi^+(\bar{x})A^T(A\Pi^+(\bar{x})A^T)^{-1}y
$$

(39)

In the overcomplete case with noise, optimizing $[x^T e^T]^T$, we have,

$$
x = \Pi^+(\bar{x})A^T(A\Pi^+(\bar{x})A^T+\Pi^+(\bar{e}))^{-1}y
$$

(40)

where $\bar{e} = y - Ax$ and $\Pi^+(\bar{e})$ is defined in the same way as $\Pi(\bar{x})$ except in terms of the functions $d_i$ rather than $f_i$. It is unnecessary to solve for the new $e$ as it is constrained to be $y - Ax$.

In the undercomplete case (when $m > n$), we can use the matrix inversion lemma to rewrite (40) as,

$$
x = (\Pi(\bar{x}) + A^T\Pi(\bar{e})A)^{-1}A^T\Pi(\bar{e})y
$$

(41)

The algorithm is not limited to the case in which the constraint set is a linear variety defined by $Ax = y$ and can be generalized to the case where $C$ is an arbitrary convex set. For super-gaussian priors (square-concave $f$), $f'(x)/x$ tends to infinity as $x$ goes to zero. Using $\Pi^+$ in the overcomplete case makes this unproblematic as the corresponding element of $\Pi^+$ tends to zero. In the undercomplete case, if one of the optimal components is very small, it may be necessary numerically to handle this explicitly.

References


