Data modeling with the elliptical gamma distribution

Suvrit Sra
LIDS
Massachusetts Institute of Technology
Cambridge, MA

Reshad Hosseini
School of ECE, College of Engineering
University of Tehran
Tehran, Iran

Lucas Theis & Matthias Bethge
Werner Reichardt Centre for
Integrative Neuroscience
Tübingen, Germany

Abstract

We study mixture modeling using the elliptical gamma (EG) distribution, a non-Gaussian distribution that allows heavy and light tail and peak behaviors. We first consider maximum likelihood parameter estimation, a task that turns out to be very challenging: we must handle positive definiteness constraints, and more crucially, we must handle possibly nonconcave log-likelihoods, which makes maximization hard. We overcome these difficulties by developing algorithms based on fixed-point theory; our methods respect the psd constraint, while also efficiently solving the (possibly) nonconcave maximization to global optimality. Subsequently, we focus on mixture modeling using EG distributions: we present a closed-form expression of the KL-divergence between two EG distributions, which we then combine with our ML estimation methods to obtain an efficient split-and-merge expectation maximization algorithm. We illustrate the use of our model and algorithms on a dataset of natural image patches.

1 Introduction

Several applications involve data of a non-Gaussian nature. Sometimes to capture manifold structure in data [3, 26, 7], or to model structure such as sparsity [18, 28]. Other common non-Gaussian situations arise when modeling data with heavy or light tails [25, 18], when studying independence [21, 16], or in a host of other situations. Our focus in this paper is also on non-Gaussian data modeling, in particular with the *Elliptical Gamma* (EG) distribution [19].

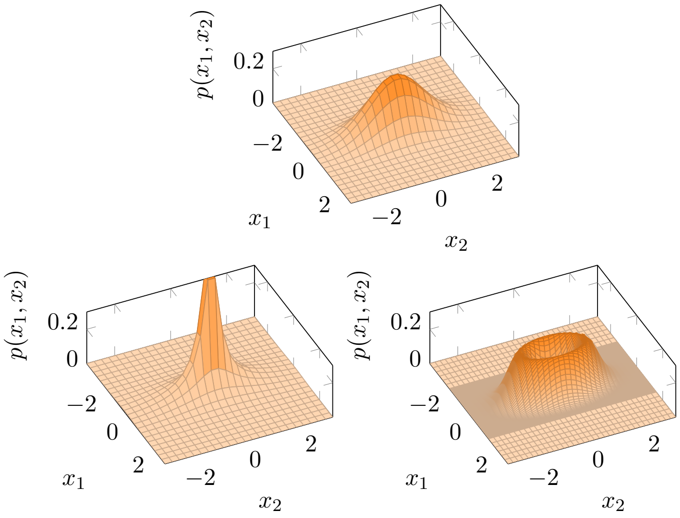

Figure 1: EG density on $\mathbb{R}^2$ with shape parameter $a = 1$, 1/3, and 3 (row-wise). All displayed densities have equal covariances; the density corresponding to $a = q/2 = 1$ (top) is a Gaussian density.

The EG density (mean-zero case) with a $q \times q$ scatter matrix $\boldsymbol{\Sigma} \succ 0$ is given by

$$\begin{aligned} p_{\text{eg}}(\boldsymbol{x}; \boldsymbol{\Sigma}, a, b) &:= \frac{\Gamma(q/2)}{\pi^{q/2}\Gamma(a)b^a|\boldsymbol{\Sigma}|^{1/2}}\varphi(x^\top \boldsymbol{\Sigma}^{-1} x) \\ \varphi(t) &:= t^{a-q/2}e^{-t/b}, \end{aligned} \tag{1.1}$$

where Γ is the usual Gamma function, and $a, b > 0$ are density shaping parameters [8], and φ is the so-called "density generating function". Density (1.1) generalizes the Gaussian, which corresponds to $\varphi(t) = e^{-t/b}$ (obtained for $a = q/2$). The additional elliptical factor $(\boldsymbol{x}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{x})^{a-q/2}$ can be used to encode different tail and peak behaviors; Figure 1 illustrates this point.

Motivation. EGDs are broadly applicable and offer rich modeling power: a mixture of zero-mean EGDs can approximate any symmetric distribution [8]. EGDs actually belong to a wider class of distributions called *Elliptically Contoured Distributions*, which have found use in for multivariate density estimation [25],

Appearing in Proceedings of the 18th International Conference on Artificial Intelligence and Statistics (AISTATS) 2015, San Diego, CA, USA. JMLR: W&CP volume 38.

Bayesian statistical data modeling [2], signal denoising [33], financial data modeling [4], and pattern recognition [34]. Mixtures of ECDs have been used successfully in many applications such as robust statistical modeling [20], denoising [27], signal processing, among others—see also the survey [25].

We consider the following two aspects of EGDs in this paper: (i) algorithms for efficient maximum likelihood (ML) parameter estimation; and (ii) mixture modeling using EGDs, along with a brief application to the modeling of image patches.

A further motivation is robust recovery of multiple subspaces—see e.g., [22]. This topic has for instance various applications in unsupervised learning, computer vision and biomedical engineering—see [30] and references therein.

Surprisingly, even the basic task of obtaining ML estimation for the parameters of an EGD turns out to be numerically very challenging: the log-likelihood may fail to be concave making maximization hard, and the positive-definiteness constraint $\mathbf{\Sigma} \succ 0$ imposes a computational burden. This background motivates the following main contributions of this paper.

Contributions.

- Two new non-Euclidean fixed point algorithms for ML parameter estimation with EGDs. Our algorithms address both the concave ($a \geq q/2$) but still numerically challenging case, as well as the nonconcave ($a < q/2$) case, which, we still manage to efficiently maximize to *global optimality.*
- A computationally efficient "split-and-merge" expectation maximization (EM) algorithm for estimating parameters for a mixture of EGDs. This algorithm uses our ML estimation algorithms for its M-step, and a particular KL-divergence derivation (details in [13] due to space paucity) for its "merge" decisions.
- An illustrative application of our model and algorithms to natural image patches.

2 Background

Let us begin by recalling basics of elliptically contoured distributions (ECDs), of which EGDs are a special case. A q-dimensional random vector $\boldsymbol{X}$ is distributed according to an ECD with a mean parameter $\boldsymbol{\mu} \in \mathbb{R}^q$ and "scatter" matrix $\mathbf{\Sigma} \in \mathbb{R}^{q\times q}$ if its characteristic function is of the form $\Phi_X(\boldsymbol{t}) = \exp(i\,\boldsymbol{t}^\top\boldsymbol{\mu})g(\boldsymbol{t}^\top\mathbf{\Sigma}^{-1}\boldsymbol{t})$, for some function $g : \mathbb{R}_+ \to \mathbb{R}$. If it exists, the density of an ECD has the form (for a suitable function f):

$$p_X(\boldsymbol{x}) \propto |\mathbf{\Sigma}|^{-1/2} f\left((\boldsymbol{x}-\boldsymbol{\mu})^\top\mathbf{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right).$$

For simplicity, we consider mean-zero ECDs, so that

$$p_X(\boldsymbol{x}) \propto |\mathbf{\Sigma}|^{-1/2} f\big(\boldsymbol{x}^\top\mathbf{\Sigma}^{-1}\boldsymbol{x}\big). \tag{2.1}$$

Under this assumption, one can factor $\boldsymbol{X}$ into a uniform hyper-spherical component and a scaled-radial component, so that $\boldsymbol{X} = \mathbf{\Sigma}^{1/2}R\boldsymbol{U}$ with $\boldsymbol{U}$ uniformly distributed over the unit hypersphere $\mathbb{S}^{q-1}$ and R a univariate random variable given by $R = \|\mathbf{\Sigma}^{-1/2}\boldsymbol{X}\|_2$ [9]. The random variable R has the p.d.f.

$$p_R(r) := 2\pi^{q/2} f(r^2) r^{q-1}/\Gamma(\tfrac{q}{2}).$$

For EGDs the squared radial component $\Upsilon = R^2$ is Gamma distributed, according to

$$p_\Upsilon(\upsilon) = \upsilon^{a-1}\Gamma(a)^{-1}b^{-a}\exp\left(-\upsilon/b\right), \tag{2.2}$$

where a is a *shape* parameter and b a *scale* parameter.

Using (2.2) as the radial distribution, we obtain the density shaping function $\varphi = f$ for (2.1); therewith, we obtain the EGD density shown in (1.1). If $\mathbf{\Sigma}$ equals the covariance matrix of the distribution, i.e., $\mathbf{\Sigma} = \mathbb{E}[\boldsymbol{X}\boldsymbol{X}^\top]$, then $b = q/a$ (see [9, Equation 2.16]).

Calculating ML estimates of the parameters of an ECD is generally not analytically possible, though in special cases such as multivariate t-distributions, a recursive algorithm is known [20]. For a review of ML estimation of ECDs see Ollila et al. [25] and references therein, as well as some more recent works [32, 40].

However, for EGDs, we can derive efficient ML estimation procedures; we present an outline in the next section, mentioning only the high-level ideas. The details are quite interesting (in our opinion) but rather technical, and are presented in [13].

3 ML parameter estimation

Let $\{\boldsymbol{x}_i\}$ be an i.i.d. sample from an EGD; then their log-likelihood (up to some constant C) is

$$\begin{aligned}\ell(\boldsymbol{x}|a,b,\mathbf{\Sigma}) &= \left(a-\tfrac{q}{2}\right)\sum\nolimits_{i=1}^{n}\log(\boldsymbol{x}_i^\top\mathbf{\Sigma}^{-1}\boldsymbol{x}_i)\\ &- b^{-1}\sum\nolimits_{i=1}^{n}\boldsymbol{x}_i^\top\mathbf{\Sigma}^{-1}\boldsymbol{x}_i - \tfrac{n}{2}\log|\mathbf{\Sigma}| + C\end{aligned} \tag{3.1}$$

We consider ML estimation of $\mathbf{\Sigma}$ for fixed a and b. From (3.1) we see that if $a \geq q/2$, then the log-likelihood ℓ is strictly concave in $\mathbf{\Sigma}^{-1}$, and therefore any critical point must be unique. But when $a < q/2$, ℓ is *not* concave, though uniqueness still holds (see Appendix B of [13]; another proof of uniqueness follows from [17, Theorem 2.2]).

Kent and Tyler [17] presented an iterative algorithm for computing the unique ML solution (for $a < q/2$)—but in contrast to our methods, their proof depends on existence of the ML-solution, a nontrivial fact that must be first established. We present new iterative ML estimation algorithms and prove their convergence for both the *concave* ($a \geq q/2$) and *nonconcave* ($a < q/2$) versions of the log-likelihood (3.1).

Since we are optimizing over an open set, we have the first-order necessary condition $\nabla_{\mathbf{\Sigma}} \ell = 0$. This yields,

$$-\tfrac{n}{2}\mathbf{\Sigma}^{-1} - (a - \tfrac{q}{2})\mathbf{\Sigma}^{-1}\Big(\sum_{i=1}^{n} \frac{\boldsymbol{x}_i \boldsymbol{x}_i^\top}{\boldsymbol{x}_i^\top \mathbf{\Sigma}^{-1} \boldsymbol{x}_i} + \tfrac{1}{b}\boldsymbol{x}_i\boldsymbol{x}_i^\top\Big)\mathbf{\Sigma}^{-1} = 0.$$

To this equation, add $\frac{n}{2}\mathbf{\Sigma}^{-1}$ and rescale by $\sqrt{\frac{2}{n}}\mathbf{\Sigma}^{\frac{1}{2}}$ to get

$$c\sum_{i=1}^{n} \frac{\mathbf{\Sigma}^{-1/2}\boldsymbol{x}_i\boldsymbol{x}_i^\top\mathbf{\Sigma}^{-1/2}}{\boldsymbol{x}_i^\top\mathbf{\Sigma}^{-1}\boldsymbol{x}_i} + d\sum_{i=1}^{n}\mathbf{\Sigma}^{-1/2}\boldsymbol{x}_i\boldsymbol{x}_i^\top\mathbf{\Sigma}^{-1/2} = \boldsymbol{I}, \tag{3.2}$$

where $c = -\frac{2(a-q/2)}{n}$ and $d = \frac{2}{bn}$. A positive definite solution to (3.2) is a candidate local maximum of the log-likelihood. To ease our presentation, we further modify (3.2) to simplify its second term. Introduce therefore the following matrix,

$$\boldsymbol{B} = d\sum\nolimits_{i=1}^{n}\boldsymbol{x}_i\boldsymbol{x}_i^\top,$$

and apply the transformation $\boldsymbol{y}_i = \boldsymbol{B}^{-1/2}\boldsymbol{x}_i$ to (3.2). Defining $\mathbf{\Gamma} = \boldsymbol{B}^{-1/2}\mathbf{\Sigma}\boldsymbol{B}^{-1/2}$ and using the fact that the square root of $\mathbf{\Gamma}$ has the form $\boldsymbol{B}^{-1/2}\mathbf{\Sigma}^{1/2}\boldsymbol{Q}^\top$ for a suitable orthogonal matrix $\boldsymbol{Q}$, (3.2) turns into

$$c\sum_{i=1}^{n}\frac{\mathbf{\Gamma}^{-1/2}\boldsymbol{y}_i\boldsymbol{y}_i^\top\mathbf{\Gamma}^{-1/2}}{\boldsymbol{y}_i^\top\mathbf{\Gamma}^{-1}\boldsymbol{y}_i} + \mathbf{\Gamma}^{-1} = \boldsymbol{I}. \tag{3.3}$$

From a solution $\mathbf{\Gamma}^*$ to (3.3), we recover $\mathbf{\Sigma}^* = \boldsymbol{B}^{1/2}\mathbf{\Gamma}^*\boldsymbol{B}^{1/2}$ as the corresponding solution to (3.2).

Our solution to equation (3.3) splits into two cases: (i) *concave* ($c < 0$); and (ii) *nonconcave* ($c > 0$).

3.1 The concave case $c \leq 0$

The case $c = 0$ is trivial so we ignore it. Noting that $c < 0$, we rearrange (3.3) to obtain from it the “positivity-preserving” iteration ($p \geq 0$):

$$\mathbf{\Gamma}_{p+1} = \Big(-c\sum_{i=1}^{n}\frac{\mathbf{\Gamma}_p^{-1/2}\boldsymbol{y}_i\boldsymbol{y}_i^\top\mathbf{\Gamma}_p^{-1/2}}{\boldsymbol{y}_i^\top\mathbf{\Gamma}_p^{-1}\boldsymbol{y}_i} + \boldsymbol{I}\Big)^{-1}. \tag{3.4}$$

Clearly, if $\mathbf{\Gamma}_0 \succ 0$, then every $\mathbf{\Gamma}_p \succ 0$ by construction. Any limit point of (3.4) is also positive definite and satisfies (3.3), and therefore yields the ML solution.

We analyze (3.4) via fixed-point theory. Consider therefore the following nonlinear map:

$$\mathcal{G} \equiv \boldsymbol{S} \mapsto \boldsymbol{I} + c'\boldsymbol{S}^{1/2}\boldsymbol{Y}\boldsymbol{D}_{\boldsymbol{S}}\boldsymbol{Y}^\top\boldsymbol{S}^{1/2},$$

where matrix $\boldsymbol{Y}$ has $\boldsymbol{y}_i$ as its ith column, $\boldsymbol{S} = \mathbf{\Gamma}^{-1}$, $\boldsymbol{D}_{\boldsymbol{S}} = \mathrm{Diag}(1/\boldsymbol{y}_i^\top\boldsymbol{S}\boldsymbol{y}_i)$ (diagonal matrix), and $c' = -c$. To show that $\mathcal{G}$ is a fixed-point map, we first need the following technical result.

Proposition 1. *Define $\mathcal{D} := \{\boldsymbol{S} \mid \boldsymbol{I} \preceq \boldsymbol{S} \preceq \mu\boldsymbol{I}\}$, for $\mu > (1 + c'n)$, then $\mathcal{G}(\mathcal{D}) \subset \mathcal{D}$.*

Proof. See Lemma 3 of [13]. □

The main result of this section is Theorem 2.

Theorem 2. *Let $\boldsymbol{S}_0 \in \mathcal{D}$ (where $\mathcal{D}$ is a compact set defined by Prop. 1). Then, the iteration $\boldsymbol{S}_{p+1} = \mathcal{G}(\boldsymbol{S}_p)$ converges to a unique fixed-point $\boldsymbol{S}^*$, and $(\boldsymbol{S}^*)^{-1}$ solves (3.4).*

Proof Sketch. Since $\mathcal{D}$ is compact (Prop. 1), $\mathcal{G}(\mathcal{D}) \subset \mathcal{D}$, and $\mathcal{G}$ is continuous, using Brouwer's fixed-point theorem [11] we know that $\mathcal{G}$ must have a fixed point in $\mathcal{D}$. However, this does not mean that one can just iterate $\boldsymbol{S}_{p+1} = \mathcal{G}(\boldsymbol{S}_p)$ to obtain a fixed point. Fortunately, since $c < 0$, the log-likelihood is strictly concave, so if it exists, its maximum must be unique. Even then, we cannot yet conclude convergence. But using concavity and Brouwer, we can inductively show that the map $\mathcal{G}^m$ has a unique fixed point for all $m \geq 1$. Then, using Proposition 4 of [13], we can conclude that actually iterating $\boldsymbol{S}_{p+1} = \mathcal{G}(\boldsymbol{S}_p)$ takes us to this unique fixed point. This fixed point clearly provides the optimal point, as it satisfies the first order necessary conditions, which are also sufficient due to concavity. □

3.2 The nonconcave case: $c > 0$

The fixed-point iteration (3.4) does not apply to $c > 0$ since positive definiteness can be no longer guaranteed. We therefore rewrite (3.3) differently. Multiplying it on the left and right by $\mathbf{\Gamma}^{1/2}$ and introducing a new parameter $\alpha > 0$, we obtain the iteration ($p \geq 0$)

$$\mathbf{\Gamma}_{p+1} = \alpha_p\mathbf{\Gamma}_p^{1/2}\boldsymbol{N}_p\mathbf{\Gamma}_p^{1/2}, \tag{3.5}$$

where $\alpha_p > 0$ is a free scalar parameter and $\boldsymbol{N}_p$ is given by the following equation:

$$\boldsymbol{N}_p = c\sum_{i=1}^{n}\frac{\mathbf{\Gamma}_p^{-1/2}\boldsymbol{y}_i\boldsymbol{y}_i^\top\mathbf{\Gamma}_p^{-1/2}}{\boldsymbol{y}_i^\top\mathbf{\Gamma}_p^{-1}\boldsymbol{y}_i} + \mathbf{\Gamma}_p^{-1}.$$

We show that under a specific choice of α_p, iteration (3.5) converges, and that in addition $\alpha_p \to \alpha^* = 1$. Thus, $\lim_{p\to\infty}\mathbf{\Gamma}_p = \mathbf{\Gamma}^*$ satisfies 3.3, and $\mathbf{\Gamma}^*$ is therefore the desired ML solution.

Our proof relies on Lemma 3 which shows that one can find α_p values that lead to the increase of the smallest eigenvalue of $\boldsymbol{N}_p$ and decrease of the largest eigenvalue of $\boldsymbol{N}_p$ (Lemma 5 of [13]).

Lemma 3. *Let $\lambda_{1,p} > \alpha_p^{-1}$ and $\lambda_{q,p} < \alpha_p^{-1}$ represent the largest and smallest eigenvalues of $\boldsymbol{N}_p$, respectively. If the data set $\{\boldsymbol{x}_i\}_{i=1}^n$ spans $\mathbb{R}^q$ then $\lambda_{1,p+1} \leq \lambda_{1,p}$ and $\lambda_{q,p+1} \leq \lambda_{q,p+1}$.*

The main result of this section is Theorem 4, which shows that there is a sequence $\{\alpha_p\} \to 1$, for which (3.5) converges (details of the proof may be found in Theorem 6 of [13]).

Theorem 4. *Let $\lambda_{1,p} \geq 1$ and $\lambda_{q,p} \leq 1$ represent the largest and smallest eigenvalues of $\boldsymbol{N}_p$ respectively. If the data set $\{\boldsymbol{x}_i\}_{i=1}^n$ spans $\mathbb{R}^q$ then one can find an α_p such that $1 \leq \lambda_{1,p+1} \leq \lambda_{1,p}$ and $\lambda_{q,p} \leq \lambda_{q,p+1} \leq 1$. This implies that iteration (3.5) converges. If ML solution $\boldsymbol{\Gamma}^*$ exists , the iteration converges to the ML solution. Moreover, $\alpha_p \to 1$, and one possible choice for α_p is given by:*

(i) $\alpha_p = 1$ if the largest eigenvalue λ'_1 and the smallest eigenvalue λ'_q of the matrix $\boldsymbol{N}'$ given below are larger and smaller than one, respectively. The matrix $\boldsymbol{N}'$ is given by

$$\boldsymbol{N}' = c\sum_{i=1}^{n} \frac{\boldsymbol{\Gamma}'^{-1/2}\boldsymbol{y}_i\boldsymbol{y}_i^\top\boldsymbol{\Gamma}'^{-1/2}}{\boldsymbol{y}_i^\top\boldsymbol{\Gamma}'^{-1/2}\boldsymbol{y}_i} + \boldsymbol{\Gamma}'^{-1},$$

where $\boldsymbol{\Gamma}' = \boldsymbol{\Gamma}_p^{1/2}\boldsymbol{N}_p\boldsymbol{\Gamma}_p^{1/2}$.
(ii) $\alpha_p = \lambda_q^{-1}$ if $\lambda'_1 \leq 1$ and $\lambda'_q \leq 1$. Where λ_q is the smallest eigenvalue of the following matrix:

$$\boldsymbol{\Gamma}' - c\sum_{i=1}^{n} \frac{\boldsymbol{y}_i\boldsymbol{y}_i^\top}{\boldsymbol{y}_i^\top\boldsymbol{\Gamma}'^{-1}\boldsymbol{y}_i}. \qquad (3.6)$$

(iii) $\alpha_p = \lambda_1^{-1}$ if $\lambda'_1 \geq 1$ and $\lambda'_q \geq 1$. Where λ_1 is the largest eigenvalue of the matrix in (3.6).

One can invoke a result of [17] or the more general theory of [31] to obtain convergence proofs for a different iteration that computes $\boldsymbol{\Gamma}$. But the convergence results of both [17, 31] depend on the existence of an ML solution.

We note that upon existence of ML solution, the fixed-point iteration works even without α_p (or equivalently with $\alpha_p = 1$). Indeed, the case $\alpha_p = 1$ corresponds to the classic iteration of Kent and Tyler [17]. Another but more *restrictive* way of proving the convergence of fixed-point iteration is by showing that the fixed-point map is contraction [32]. However, including a variable α_p can be seen as improving the contracting factor in the contraction map, which speeds up the empirically observed convergence shown in Section 5.

Theorem 4 proves a **stronger** result because it does not depend on any existence requirement on the ML solution. This generality has some important consequences: (i) if the ML solution exists, then inevitably iteration (3.5) converges to it; but (ii) when the ML solution does not exist (which is well possible), then the iterative algorithm still converges, though now the convergent solution is singular. This singular matrix possesses specific structure that can be then used for robust subspace recovery, which incidentally also generalizes the subspace recovery approach of [39]. This topic is, however, beyond the present paper and will be considered elsewhere.

It worth mentioning that the above theorem suggests α_p values which are not necessarily optimal, though easy to calculate. In practice, we observed that if one chooses the parameter α_p such that the trace of the matrix $\boldsymbol{N}_{p+1}$ becomes q, that is $\alpha_p = \mathrm{tr}(\boldsymbol{\Gamma}'^{-1})/(2a)$, then the convergence is faster. However, for this case our convergence proof does not apply.

4 Mixture modeling with EGDs

After the above theory we are now ready to discuss mixture modeling algorithms. In Section 4.2, we present a “split-and-merge” expectation maximization (EM) algorithm for estimating parameters of an EGD mixture model. This algorithm uses a certain KL-Divergence computation mentioned in Section 4.3 to makes its “merge” decisions (i.e., to decide whether two mixture components should be merged into one).

4.1 EM Algorithm for mixture of EGDs

A K-component mixture of Elliptical Gamma distributions (MEG) has the disitribution

$$p(\boldsymbol{x}) = \sum\nolimits_{k=1}^{K} p_k p_{eg}(\boldsymbol{x}; \boldsymbol{\Sigma}_k, a_k, b_k), \qquad (4.1)$$

where $\sum_k p_k = 1$ ($p_k \geq 0$).

We use a block coordinate ascent algorithm for implementing the maximization step. Specifically, we fix a_k and b_k and apply one step of EM to obtain $\boldsymbol{\Sigma}_k$ ($1 \leq k \leq K$) using the fixed-point algorithms developed above. Next, we fix $\boldsymbol{\Sigma}_k$ and update a_k, b_k. Here, the following variable change $\upsilon_k = \boldsymbol{x}^T\boldsymbol{\Sigma}_k\boldsymbol{x}$ proves helpful, because with it the (4.1) turns into

$$p(\boldsymbol{x}) = \sum\nolimits_{k=1}^{K} p_k p_{ga}(\upsilon_k; a_k, b_k),$$

where p_{ga} is the Gamma density (2.2).

The two main steps of an EM algorithm for the first stage are as follows:

- *E-step*: Compute the weights (for each data-point i and component k):

$$t_{ki} = \frac{p_k p_{eg}(\boldsymbol{x}_i; \boldsymbol{\Sigma}_k, a_k, b_k)}{\sum_{l=1}^{K} p_l p_{eg}(\boldsymbol{x}_i; \boldsymbol{\Sigma}_l, a_l, b_l)} = \frac{p_k p_{ga}(\upsilon_{ki}; a_k, b_k)}{\sum_{l=1}^{K} p_l p_{ga}(\upsilon_{ki}; a_l, b_l)}$$

- *M-step*: Update the parameters of the mixture model by maximizing:

  $$\ell_k(\boldsymbol{\Sigma}_k, a_k, b_k; \{\boldsymbol{x}_i\}_{i=1}^n) = \sum_{i=1}^n t_{ki} \log p_{eg}(\boldsymbol{x}_i; \boldsymbol{\Sigma}_k, a_k, b_k),$$

  where the priors p_k are as usual $p_k = n^{-1}\sum_{i=1}^n t_{ki}$.

The fixed-point methods of Section 3 can be easily modified to accommodate weighted log-likelihoods.

Similar to the first stage, one step of EM for the second stage also consists of two steps that are applied sequentially until convergence. The E-step and updates to p_k are similar to the first stage. But for updating a_k and b_k parameters in the M-step, we maximize the following objective function:

$$\ell_k(a_k, b_k; \{\upsilon_{ki}\}_{i=1}^n) = \sum\nolimits_{i=1}^n t_{ki} \log p_{ga}(\upsilon_{ki}|a_k, b_k)$$

The maximum weighted log-likelihood estimates of these parameters can be calculated efficiently using Generalized Newton method [23]. Modifying the method explained in [23] to account for weights, we will obtain the following fixed-point iteration:

$$\frac{1}{a_{k_{new}}} = \frac{1}{a_k} + \frac{\overline{\log \upsilon_k} - \log \bar{\upsilon}_k + \log a_k - \Psi(a_k)}{a_k^2(\frac{1}{a_k} - \Psi'(a_k))}$$

Where $\bar{z}$ is weighted mean over z $(\sum_i t_{ki}z_{ki}/\sum_i t_{ki})$ and Ψ is the digamma function. The other parameter is calculated simply using the following equation:

$$b_k = \bar{\upsilon}_k / a_k.$$

4.2 Split-and-Merge EM for MEG

To counter the problem of poor local optima that impede EM, following [36] we derive below a more refined "split-and-merge" EM procedure. Ueda et al. [36] identified false division of the number of mixture components in different parts of the data cloud as a major impediment. Consequently, they proposed a remedy for countering local minima that is quite effective in practice. The idea is to iteratively find candidates to merge and candidates to split while maximizing the log-likelihood. This process is continued until further splitting or merging fails to improve the model fit. [5] added new criteria for splitting and merging to the ones given in [36, 37] and also modified the original split-and-merge algorithm. One criterion explained in [5] is to find two components that have the minimum symmetric KL-Divergence difference. We observed that this criterion seems to correctly specify the components, merging which leads to the highest improvement in likelihood. However, all three different criteria for splitting given in [5] have problems in identifying the correct component to split. In practice, these methods often select the component with the highest entropy, though clearly this component is not necessarily the best candidate for splitting.

Algorithm 1: Pseudo-code for split-then-merge algorithm

```
**Input:**
Number of components: K; Observations: $\{\boldsymbol{x}_i\}_{i=1}^n$;
Maximum components: $K_{\max}$
**Initialize:**
$k \leftarrow 1$; $\mathcal{I} \leftarrow \{1\}$; $\Delta\ell_1 \leftarrow \infty$;
$\boldsymbol{\theta}_1^* = \arg\max_{\boldsymbol{\theta}_1} \ell(\boldsymbol{\theta}_1; \{\boldsymbol{x}_i\}_{i=1}^n)$; $\ell_{\text{cur}} \leftarrow \ell(\boldsymbol{\theta}_1^*; \{\boldsymbol{x}_i\}_{i=1}^n)$
**Splitting stage:**
while $\exists i, \Delta\ell_i > -\infty$ and $k < K_{\max}$ do
  $\bar{i} = \arg\max_{i \in \mathcal{I}} \Delta\ell_i$;
  $\{\bar{\boldsymbol{\theta}}_{\bar{i}}, \bar{\boldsymbol{\theta}}_k\} = \arg\max_{\boldsymbol{\theta}_{\bar{i}}, \boldsymbol{\theta}_k} \ell(\{\boldsymbol{\theta}_i^*\}_{i \in \mathcal{I}-\{\bar{i}\}}, \boldsymbol{\theta}_{\bar{i}}, \boldsymbol{\theta}_{k+1}; \{\boldsymbol{x}_i\}_{i=1}^n)$

  $\ell_{\text{new}} \leftarrow \ell_n(\boldsymbol{x}|\{\boldsymbol{\theta}_i^*\}_{i \in \mathcal{I}-\{\bar{i}\}}, \bar{\boldsymbol{\theta}}_{\bar{i}}, \bar{\boldsymbol{\theta}}_{k+1})$ ; $d = \ell_{\text{new}} - \ell_{\text{cur}}$
  if $d > h$ then
    $k \leftarrow k+1$; $\boldsymbol{\theta}_{\bar{i}}^* \leftarrow \bar{\boldsymbol{\theta}}_{\bar{i}}$; $\boldsymbol{\theta}_{k+1}^* \leftarrow \bar{\boldsymbol{\theta}}_{k+1}$; $\Delta\ell_{\bar{i}} \leftarrow d$;
    $\Delta\ell_k \leftarrow d$; $\mathcal{I} = \mathcal{I} \cup \{k\}$; $\ell_{\text{cur}} \leftarrow \ell_{\text{new}}$
  else
    $\Delta\ell_{\bar{i}} \leftarrow -\infty$
  end if
end while
**Merging stage:**
while $k > K$ do
  $\{\bar{i}, \bar{j}\} = \arg\min_{i,j \in \mathcal{I}} \text{KL}\big[\mathcal{EGD}(\boldsymbol{\theta}_i^*)||\mathcal{EGD}(\boldsymbol{\theta}_j^*)\big]$
  $\mathcal{I} = \mathcal{I} - \{\bar{j}\}$;
  $\boldsymbol{\theta}_{\bar{i}}^* = \arg\max_{\boldsymbol{\theta}_{\bar{i}}} \ell(\{\boldsymbol{\theta}_i^*\}_{i \in \mathcal{I}-\{\bar{i}\}}, \boldsymbol{\theta}_{\bar{i}}; \{\boldsymbol{x}_i\}_{i=1}^n)$
  $k \leftarrow k-1$
end while
**Overall Optimization:**
$\{\boldsymbol{\theta}_i^*\}_{i \in \mathcal{I}} = \arg\max_{\{\boldsymbol{\theta}_i\}_{i \in \mathcal{I}}} \ell(\{\boldsymbol{\theta}_i\}_{i \in \mathcal{I}}; \{\boldsymbol{x}_i\}_{i=1}^n)$
return $\{\boldsymbol{\theta}_i^*\}_{i \in \mathcal{I}}$
```

Therefore, we propose a variant of the split-and-merge algorithm that not only solves the problem of finding the component that needs to be split, but does so computationally efficiently—pseudocode is provided as Algorithm 1. Therein, $\boldsymbol{\theta}_k$ represents parameters of component k, i.e., $\{\boldsymbol{\theta}_k\} \equiv \{p_k, a_k, b_k, \boldsymbol{\Sigma}_k\}$. The first stage of the algorithm splits components until no further improvement is possible, or when the number of components reaches an upper limit $K_{\max}$. The algorithm chooses components, which in their previous split led to the highest improvement in log-likelihood $(\Delta\ell_i)$. If after a split the improvement is less than a threshold denoted by t, the split is not accepted and the component is not further split. The second stage of the algorithm finds components with the minimum KL-Divergence and merges them. This process is continued until the number of components reaches K. At the last stage, the algorithm performs one step of optimization over all components. The proposed algorithm is efficient: optimization is performed only over the parameters of the split or merged components; computational complexity of the splitting step is equal to optimizing the mixture model with two components;

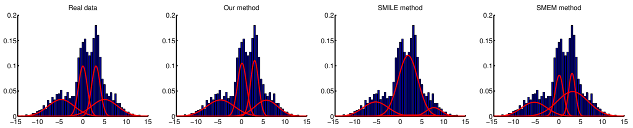

Figure 2: Left to right: ground truth mixture (red line) and empirical density; mixture recovered by our split-and-merge method; mixtures recovered by the SMILE [5] and SMEM [37] algorithms.

complexity of merging step is equal to ML estimation for just one component.

Here we discuss some points regarding split-and-merge algorithm described above.

- For each step of the splitting stage, initialization over split mixture is done simply by randomly perturbing the parameters before splitting.
- For initialization in the merging step, we can simply use one of the mixtures that are merged.
- Parameter $K_{\max}$ is chosen based on the maximum computational time. The typical value of $K_{\max} \approx 2K$ works well in practice.
- If there is no validation set that is checked during optimization to avoid overfitting, then the threshold h can be chosen by

  $$h = d \sum_{k=1}^{K+1} \frac{n_{k_{new}}}{n_{k_{new}} - d - 1} + d \sum_{k=1}^{K} \frac{n_{k_{old}}}{n_{k_{old}} - d - 1},$$

  where $d = q(q+1)/2 + 1$ is the number of parameters for each component and $n_{k_{new}}$, $n_{k_{old}}$ are the number of data in component k after and before split, respectively. The quantity $dn/(n-d-1)$ is *corrected Akaike Information Criterion* (AICc) [15] that approximately measures expected cross-validation bias between training and test sets.
- If early-stopping is used to avoid overfitting, then threshold t can be chosen to be smaller number (a fraction of d like $d/10$ works fine in practice).

The results of applying proposed algorithm to a mixture of one-dimensional Gaussian is shown in Fig. 2, which shows that our algorithm successfully recovers the distribution. The second and third plots in Fig. 2 show the result of an alternative algorithm (SMILE) explained in [5] and the basic SMEM algorithm of [37]. Due to the problem of finding good candidates for splitting, the SMILE and SMEM approaches could not recover the underlying mixture accurately.

4.3 KL Divergence between EGDs

We conclude this section by presenting an expression of the KL-Divergence between two EGDs (for derivation details see Section 4 of [13]). This computation plays a role in the merge step of Algorithm 1, and may be of independent interest too.

Let P and Q be EGDs with parameters $(a_p, b_p, \mathbf{\Sigma}_1)$ and $(a_q, b_q, \mathbf{\Sigma}_2)$ respectively. The KL-Divergence between P and Q is given by:

$$\mathrm{KL}(P\|Q) = \log\left(\frac{\Gamma(a_q) b_q^{a_q}}{\Gamma(a_p) b_p^{a_q}}\right) + (a_p - a_q)\Psi(a_p) - a_p$$
$$- \tfrac{1}{2}\log(|\mathbf{\Sigma}|) + \tfrac{a_p b_p}{q b_q}\mathrm{tr}(\mathbf{\Sigma}) - (a_q - \tfrac{q}{2})\mathcal{A},$$

where Ψ is the digamma function [1, Chapter 6.3], $\mathbf{\Sigma} = \mathbf{\Sigma}_1 \mathbf{\Sigma}_2^{-1}$, and $\mathcal{A} = \mathbb{E}[\log \sum_{i=1}^{q} \lambda_i n_i^2] - \Psi(\frac{q}{2}) - \log 2$; the n_is are independent zero mean and unit variance Gaussian variables; the λ_is are eigenvalues of $\mathbf{\Sigma}$. To compute $\mathcal{A}$ one needs a numerical procedure (practical approaches are described in Appendix C of [13]). The top plot in Fig. 3 shows the KL-Divergence between two EGDs as dimension q increases. The observed distribution is an EGD with arbitrary scale parameter while the model is a Gaussian with the same covariance. This plot reveals that if the scale parameter is very small or if it is very large, the KL-Divergence becomes very large, growing to infinity in the limit. This

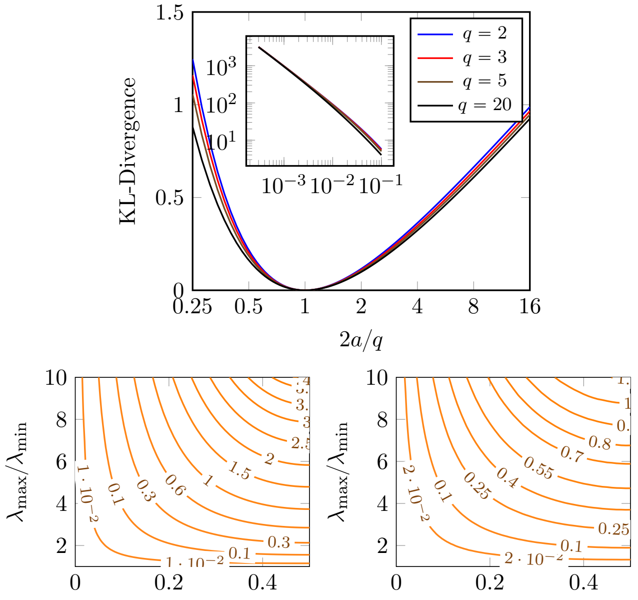

Figure 3: Top plot: KL-divergence between two EGDs as dimension q increases. Observed distribution is a EGD with some arbitrary scale parameter and the model distribution is a Gaussian with the same covariance matrix. Bottom plots: KL-divergence of two EGDs with the same scale parameter in two dimensions (left plot) first part of the expression normalized by the shape parameter ($\mathrm{tr}(\mathbf{\Sigma})/q - 1$) and (right plot) the term $\mathcal{A}$. Observed distribution is rotated version of the model distribution and the X-axis represents the rotation degree. Y-axis is the ratio of the largest eigenvalue to the smallest eigenvalue.

shows that the “goodness-of-fit” can be substantially improved for the EGD model relative to the Gaussian.

If we have two EGDs with the same shape parameter a such that the covariance matrix, say $\mathbf{\Sigma}_1$, of one of the distributions is viewed as a rotation of the other ($\mathbf{\Sigma}_2$). Then, we have $\mathrm{KL}(P\|Q) = a\left(\mathrm{tr}(\mathbf{\Sigma}_1\mathbf{\Sigma}_2^{-1})/q - 1\right) - (a - q/2)\mathcal{A}$. The left plot (2nd row) in Fig. 3 shows contours of the term $\mathrm{tr}(\mathbf{\Sigma}_1\mathbf{\Sigma}_2^{-1})/q - 1$ and the right plot shows contours of the term $\mathcal{A}$. The x-axis is the rotation degree and the y-axis is the ratio of the largest eigenvalue to the smallest one. For small a the two terms get similar signs and since the behavior of two terms look similar, KL-divergence changes more by changing rotation degree and condition number of the covariance matrix. For large a, those terms will get opposite signs and cancel each other.

5 Experiments and application

5.1 ML-estimation using fixed-point iteration

In our first set of experiments we report results on the convergence speed of our fixed-point algorithms, namely (3.4) and (3.3). Fig. 4 compares our algorithms to three state-of-the-art (Riemannian) manifold optimization techniques: limited-memory BFGS (lbfgs), trust-region and conjugate gradient [6]; we also compare the classic iteration (when it applies) of Kent and Tyler [17]. We remark that we also tested other optimization techniques such as interior-point methods [24] but do not include them in the results because they were vastly slower than manifold optimization techniques.

To generate these results, we sampled 10,000 points from an EGD with a random covariance matrix, and initialized the iterations with a random covariance. The left plot in Fig. 4 shows the result for the case $a = 1$ (nonconcave maximization) and right plot is for the case $a = 50$ (concave case). As can be seen from Fig. 4 (which is on a log-scale), our fixed-point theory yields methods that run much faster (5–10 times) than competing techniques, for both the difficult nonconcave case, as well as the concave case.

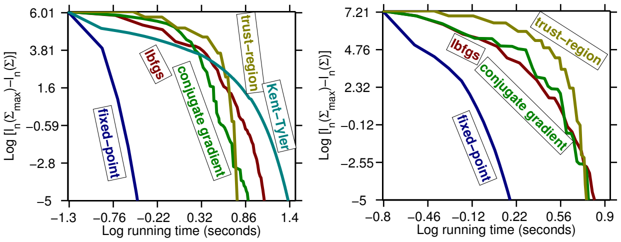

Figure 4: Comparison of the proposed fixed-point algorithms against manifold optimization techniques for EG distributions with dimension equal to 64 (left plot) $a = 1$ (right plot) $a = 50$

5.2 Application: Natural Image Statistics

We use MEG to model statistical distribution of natural image patches. The data used for fitting the model is patches sampled from random locations in a natural image dataset. Fig. 5 provides intuition as to why we model statistics of image patches using MEGs than just a mixture of Gaussians.

We extracted image patches of two different sizes 8×8 and 16×16 from random locations in the van Hateren dataset [38]. This dataset contains 4167 images; we excluded images that had problems, e.g., were noisy, blurred, etc. We extracted 200,000 training image patches, and 10 sets of 100,000 test image patches from remaining 3632 images. We preprocess image patches by log-transforming pixel intensities. Then, we added Gaussian white noise with standard deviation equal to $1/32$ of the standard deviation of images.

We ran our experiments on several different random samplings, but got very small errorbars between 0.01–0.02, so we do not include these in the comparisons to avoid clutter.

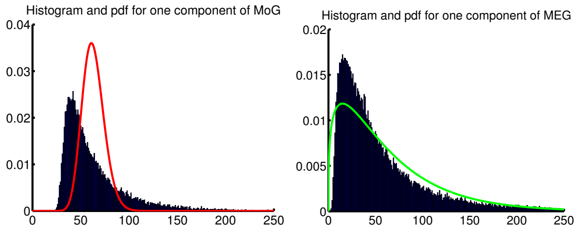

Figure 5: Plots of the radial distribution of one component randomly chosen from 8 components in a mixture of Gaussians (left) and in MEG (right). The MEG component (up to a scaling) seems to describe the data distribution much more accurately.

We evaluate the performance of different models using the *Multi-Information Rate* (MI-Rate) criterion. MI-Rate (in bits/pixel) has the intuitive flavor that it approximately shows the number of bits per pixel that one saves if the patch-level model distribution is used compared to the case that all pixels are modeled independently. Formally, it is defined as

$$\text{MI-Rate} \approx \left(H(X_0) + \tfrac{1}{n-1}\ell(\boldsymbol{\theta}; \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)\right)/\log 2,$$

where $H(X_0)$ is the entropy of one pixel. The relation becomes exact if $n \to \infty$ [14].

Table 1 summarizes the performance of different procedures in terms of MI-Rate. In all models, the DC component is modeled independently using mixture of Gaussians with 10 components. Two different sizes are included in order to observe how the MI-Rate estimate of different models change if the patch size is increased. Among different methods, MEG shows the best performance, *yielding the highest MI-Rate per pixel.*

In the table, Gauss denotes the simple Gaussian model; the MI-Rate captured by this model is called the amount of second-order information present in the data. The number of layers in hierarchical ICA (HICA) is 8 for 8×8 patches and 4 for 16×16 patches [12]. The number of mixtures for MoG and MEG is 8 [41]. Note that both MoG and HICA are universal approximators, therefore theoretically they may reach the performance of MEG *but with more parameters.* In practice, however, parsimonious models are usually preferred. L_p-spherical model is a density model proposed in Sinz et al. [29]. RG+ICA corresponds to radial Gaussianisation followed by one layer ICA [12]. DBN corresponds to Deep Belief Networks and GRBM corresponds to Gaussian Restricted Boltzmann Machine. The MI-Rate of DBN and DRBM were evaluated by the method explained in [35].

We emphasize that *the differences in MI-Rate shown in Table 1 are significant*, because closer to the upper limit of the MI-rate any improvement means capturing a lot of perceptually relevant regularities of the underlying distribution, a claim grounded in the recent psychophysical results in [10].

Finally, Fig. 6 visualizes the effect of number of mixture components on the performance. The baseline Gaussian MI-Rate is plotted as a dotted line.

Model	8×8	16×16
Gauss	2.50	2.60
GRBM	2.69	2.74
DBN	2.73	2.79
ICA	2.73	2.83
EG	2.83	2.90
HICA	2.84	2.91
L_p-spherical	2.85	2.95
RG + ICA	2.87	3.00
MoG	2.89	2.98
MEG	**2.93**	**3.02**

Table 1: MI-Rate (bits/pixel; higher is better) for different models and two different patch sizes. The differences in MI-Rate are significant (please see text for discussion).

6 Discussion and future work

We studied a powerful class of symmetric distributions, namely, Elliptical Gamma distributions. We presented theory outlining existence and uniqueness of maximum likelihood estimators for EGDs and developed simple and computationally effective algorithms computing these. As an application of our theory, we illustrated numerical results against state-of-the-art manifold optimization solvers [6] as well as classical methods [17]: in all cases, the fixed-point algorithms presented in the paper were seen to be much faster than competing approaches. Subsequently, we also studied mixture models based on EGDs and tested them on an application involving natural image statistics, for which our models seem to offer state-of-the-art performance.

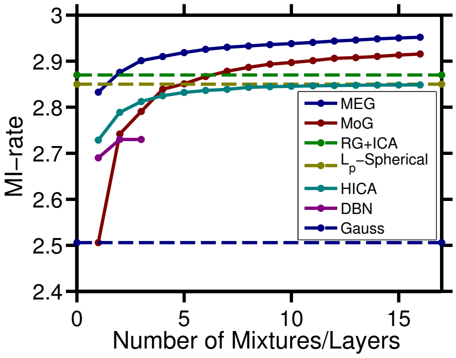

Figure 6: MI-Rate for MEG and other methods with increasing number of parameters. Unsurprisingly, with large enough number of parameters (number of mixture components / layers) the differences between the models become less severe, but MEG still retains an edge.

Several avenues of further research remain open. The most important direction is to study robust subspace recovery and its applications [30]. Other potential directions involve developing mathematical tools to study stochastic processes based on EGDs, as well as to investigate other applications where non-Gaussian data can benefit from EGDs or their mixture models. We hope that the basic theory and practical application outlined in this paper encourage other researchers to also study non-Gaussian modeling with EGDs or families richer than them.

References

[1] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions, With Formulas, Graphs, and Mathematical Tables.* Dover, 1974.

[2] R. B. Arellano-Valle, G. del Pino, and P. Iglesias. Bayesian inference in spherical linear models: robustness and conjugate analysis. *Journal of Multivariate Analysis*, 97(1):179–197, 2006.

[3] A. Banerjee, I. S. Dhillon, J. Ghosh, and S. Sra. Clustering on the unit hypersphere using von Mises-Fisher distributions. *Journal of Machine Learning Research*, 6:1345–1382, Sept. 2005.

[4] N. H. Bingham and R. Kiesel. Semi-parametric modelling in finance: theoretical foundations. *Quantitative Finance*, 2(4):241–250, 2002.

[5] K. Blekas and I. E. Lagaris. Split–merge incremental learning (SMILE) of mixture models. In *Artificial Neural Networks–ICANN 2007*, pages 291–300.

Springer Berlin Heidelberg, 2007.

[6] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre. Manopt, a Matlab toolbox for optimization on manifolds. *Journal of Machine Learning Research*, 15: 1455–1459, 2014. URL http://www.manopt.org.

[7] Y. Chikuse. *Statistics on special manifolds*, volume 174. Springer, 2003.

[8] K.-T. Fang and Y.-T. Zhang. *Generalized multivariate analysis*. Springer, 1990.

[9] K.-T. Fang, S. Kotz, and K. W. Ng. *Symmetric multivariate and related distributions*, volume 1. Chapman and Hall, 1990.

[10] H. E. Gerhard, F. A. Wichmann, and M. Bethge. How sensitive is the human visual system to the local statistics of natural images? *PLoS computational biology*, 9(1):e1002873, 2013.

[11] A. Granas and J. Dugundji. *Fixed Point Theory*. Springer, 2003.

[12] R. Hosseini and M. Bethge. Hierarchical models of natural images. In *Frontiers in Computational Neuroscience*, 2009.

[13] R. Hosseini, S. Sra, L. Theis, and M. Bethge. Statistical inference with the Elliptical Gamma Distribution. *Submitted*.

[14] R. Hosseini, F. Sinz, and M. Bethge. Lower bounds on the redundancy of natural images. *Vision research*, 50(22):2213–2222, 2010.

[15] C. M. Hurvich and C.-L. Tsai. Bias of the corrected aic criterion for underfitted regression and time series models. *Biometrika*, 78(3):499–509, 1991.

[16] A. Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, 1999.

[17] J. T. Kent and D. E. Tyler. Redescending M-estimates of multivariate location and scatter. *The Annals of Statistics*, 19(4):2102–2119, Dec. 1991.

[18] S. Kotz, T. Kozubowski, and K. Podgorski. *The Laplace Distribution and Generalizations: A Revisit With Applications to Communications, Economics, Engineering, and Finance*. Springer, 2001.

[19] M. Koutras. On the generalized noncentral Chi-Squared distribution induced by an elliptical gamma law. *Biometrika*, 73(2):528–532, Aug. 1986.

[20] K. L. Lange, R. J. A. Little, and J. M. G. Taylor. Robust statistical modeling using the t distribution. *Journal of the American Statistical Association*, 84 (408):881–896, Dec. 1989.

[21] T.-W. Lee, M. S. Lewicki, and T. J. Sejnowski. Unsupervised classification with non-gaussian mixture models using ica. In *Advances in neural information processing systems*, pages 508–514, 1999.

[22] G. Lerman, T. Zhang, et al. Robust recovery of multiple subspaces by geometric lp minimization. *The Annals of Statistics*, 39(5):2686–2715, 2011.

[23] T. P. Minka. Estimating a gamma distribution. http://research.microsoft.com/en-us/um/people/minka/papers/minka-gamma.pdf, 2002.

[24] Y. Nesterov and A. Nemirovski. *Interior-point polynomial algorithms in convex programming*. SIAM, 1994.

[25] E. Ollila, D. Tyler, V. Koivunen, and H. V. Poor. Complex elliptically symmetric distributions: Survey, new results and applications. *IEEE Transactions on Signal Processing*, 60(11):5597–5625, 2011.

[26] X. Pennec. Intrinsic statistics on Riemannian manifolds: Basic tools for geometric measurements. *Journal of Mathematical Imaging and Vision*, 25(1):127–154, 2006.

[27] H. Rabbani and M. Vafadust. Image/video denoising based on a mixture of Laplace distributions with local parameters in multidimensional complex wavelet domain. *Signal Processing*, 88(1):158–173, 2008.

[28] M. W. Seeger and H. Nickisch. Large scale bayesian inference and experimental design for sparse linear models. *SIAM Journal on Imaging Sciences*, 4(1):166–199, 2011.

[29] F. Sinz, E. Simoncelli, and M. Bethge. Hierarchical modeling of local image features through l_p-nested symmetric distributions. In *Advances in Neural Information Processing Systems*, pages 1696–1704, 2009.

[30] M. Soltanolkotabi, E. J. Candes, et al. A geometric analysis of subspace clustering with outliers. *The Annals of Statistics*, 40(4):2195–2238, 2012.

[31] S. Sra and R. Hosseini. Geometric optimisation on positive definite matrices for elliptically contoured distributions. In *Advances in Neural Information Processing Systems*, pages 2562–2570, 2013.

[32] S. Sra and R. Hosseini. Conic geometric optimisation on the manifold of positive definite matrices. *SIAM Journal on Optimization*, Accepted.

[33] S. Tan and L. Jiao. Multivariate statistical models for image denoising in the wavelet domain. *International Journal Computer Vision*, 75(2):209–230, 2007.

[34] J. Theiler, C. Scovel, B. Wohlberg, and B. R. Foy. Elliptically contoured distributions for anomalous change detection in hyperspectral imagery. *IEEE Geoscience and Remote Sensing Letters*, 7(2):271–275, 2010.

[35] L. Theis, S. Gerwinn, F. Sinz, and M. Bethge. In ALL likelihood, deep belief is not enough. *Journal of Machine Learning Research*, 12:3071–3096, 2011.

[36] N. Ueda, R. Nakano, Z. Ghahramani, and G. E. Hinton. Split and merge EM algorithm for improving gaussian mixture density estimates. *Journal of VLSI Signal Processing*, 26(1):133–140, 2000.

[37] N. Ueda, R. Nakano, Z. Ghahramani, and G. E. Hinton. Smem algorithm for mixture models. *Neural computation*, 12(9):2109–2128, 2000.

[38] J. H. van Hateren and A. van der Schaaf. Independent component filters of natural images compared with simple cells in primary visual cortex. *Proc. Royal Soc. B*, 265(1394):1724–1726, 1998.

[39] T. Zhang. Robust subspace recovery by geodesically convex optimization. *arXiv preprint arXiv:1206.1386*, 2012.

[40] T. Zhang, A. Wiesel, and S. Greco. Multivariate generalized gaussian distribution: Convexity and graphical models. *IEEE Transaction on Signal Processing*, 60(11):5597–5625, Nov. 2013.

[41] D. Zoran and Y. Weiss. Natural images, gaussian mixtures and dead leaves. In *Advances in Neural Information Processing Systems*, 2012.