When does Alternating Descent Conditional Gradient work better than non-convex optimization with random restarts?

Charles Zheng and ?

November 23, 2015

Abstract

Alternating Descent Conditional Gradient (ADCG) is a method for solving sparse inverse problems which combines nonconvex and convex optimization techniques. Although ADCG is observed to achieve superior performance to alternative approaches, little is known about its performance due to its non-convex subroutines. In this work, we consider a sparse inverse problem with i.i.d. noise on a d-dimensional lattice, and given a general nonlinear optimization subroutine, compare the performance of ADCG equipped with the given subroutine versus the approach of simply applying the subroutine with random intializations. We establish a regime where the size of the lattice scales with the number of sources as well as the amplitude of each source, in which ADCG converges to the global minimum with fixed number of calls while the probability of reaching the global minimum under a random intitialization goes to zero.

1 Introduction

In numerous applications, one is interested in reconstructing the locations and parameters of multiple signal sources from noisy observations. For instance, in super-resolution imaging, one uses a microscope to obtain a 2D image of a number of fluorescing point sources, and the goal is to recover the locations of the point sources on the slide.

Such a sparse inverse problem is described by a known, or unknown number of sources K, a vector of parameters $\theta_i \in \mathbb{R}^p$ describing the ith source (e.g. location in space), and a positive real weight w_i giving amplitude of the signal from the ith source. The signal from a single source θ is a function in $\mathbb{R}^d$, denoted by ψ_θ, and the combined signal from all K sources is given by the function

$$\mu(x) = \sum_{i=1}^{K} w_i \psi_{\theta_i}(x).$$

In the case of super-resolution imaging, for example, the parameter $\theta \in \mathbb{R}^2$ gives the location of the point source on the slide, w_i gives the intensity of the fluorescence, and $\psi_\theta(x)$ is a symmetric kernel function centered at θ, given by the point-spread function of the lens.

Given measurement locations $x_1, \ldots, x_N$, we observe signals

$$y_i \sim F(\mu(x_i)),$$

where $\{F(\mu)\}_{\mu \in \mathbb{R}}$ is a family of parametric distributions. For instance, a simple case is $F(\mu) = N(\mu, \sigma^2)$, corresponding to the familiar Gaussian error model

$$y_i = \mu(x_i) + \epsilon_i,\ \epsilon_i \sim N(0, \sigma^2).$$

Very shortly, we will assume that $F(\mu)$ has a density $f_\mu(y)$ on $\mathbb{R}$, and hence a negative log-likelihood function

$$\ell(y; \mu) = -\log(f_\mu(y))$$

Having observed $y_1, \ldots, y_N$, our goal is to estimate the number of sources K (if K is unknown), and to recover the locations $\theta_1, \ldots, \theta_K$. A natural approach is to minimize the objective function

$$\text{minimize} \sum_{i=1}^{n} \ell\left(y_i; \sum_{j=1}^{K} w_j \psi_{\theta_j}(x_i)\right)$$

subject to some sparsity constraint, where ℓ is the log-likelihood function for $\{F(\mu)\}$. The objective function is a function of a set of weight-parameter pairs $\{w_i, \theta_i\}$, which as Boyd et al. noted describe a positive measure on the parameter space

$$\sum_{i=1}^{K} w_i \delta_{\theta_i}.$$

Since we are concerned with computational aspects in this work, we will generally work with the weight-parameter pairs represented as a set of tuples

(w_i, θ_i) rather than as a measure, though we will borrow from the terminology and refer to a weight-parameter pair (w_i, θ_i) as an *atom*, w_i as the weight or amplitude of the ith atom, and θ_i as the location of the atom.

However, throughout the literature, a variety of methods have been proposed for choosing the sparsity constraint and for minimizing the objective function. We describe three main categories of methods:

- Using nonlinear optimization (e.g. gradient descent or Newton's method) to minimize the objective function subject to a constraint on the putative number of sources K.

- Discretizing the parameter space by choosing candidate parameters $\theta_1, \ldots, \theta_m \in \mathbb{R}^p$, then finding the optimal weights

  $$\text{minimize}_w \sum_{i=1}^{n} \ell \left(y_i; \sum_{j=1}^{m} w_j \psi_{\theta_j}(x_i) \right),$$

  possibly subject to an L_1-norm constraint or penalty on the weights w. Note that the discretized problem is *convex* if ℓ is convex.

- Alternating Descent Conditional Gradient. (To be described below.)

Nonlinear optimization is not guaranteed to achieve the global minimum of the objective function, hence a usual approach is to run nonlinear optimization multiple times with random starting conditions. As a result, it is often quite costly to get a good solution of the optimization problem using nonlinear optimization.

In contrast, when ℓ is convex, it is possible to deterministically approximate the global minimum with the discretization approach, by choosing a suitably fine discretization and then applying convex optimization to solve the discretized objective function. However, one is limited to using convex constraints, which excludes the possibility of solving the optimization problem subject to a constraint or penalty on the number of sources K, which is equal to the L_0 norm of the weights. Instead, here one typically places a constraint or penalty on the L_1 norm of the weights, since the L_1-norm is the tightest convex relaxation of the L_0 norm. Intuitively, one expects the L_1 convex relaxation to yield a worse solution than the L_0 constrained problem; while plenty of theoretical results (Morgenshtern, Candes, etc.) establish the statistical properties of the estimators resulting from L_1 minimization, little is known about the comparative performance of L_0-constrained minimization, even supposing that the global minima are achieved. In particular, it is not possible to apply the sparse recovery results of Donoho et al., since the design matrix in our problem is typically highly collinear and hence violates the usual L_1 support recovery conditions. It is worth mentioning the work by Slawski (2012), which introduces a framework for studying the sparse recovery problem given such highly correlated design matrices, and which also provides results on denoising.

Alternating Descent Conditional Gradient (Boyd et al.) combines the convex and nonconvex approaches. It is shown to have guaranteed performance to the global minimizer, subject to *convex* sparsity constraints. The algorithm is defined with reference to a gradient subroutine τ and a nonlinear descent subroutine ν. The gradient subroutine is given residuals $r_1, \ldots, r_N$ as input, and and outputs the parameter θ whose signal maximizes the inner product with the gradient of the loss with respect to the residuals:

$$\tau(r_1, \ldots, r_n) = \text{argmax}_\theta \sum_{i=1}^{n} \psi_\theta(x_i)\dot{\ell}(r_i).$$

The descent subroutine ν is given a set of atoms $\{(w_i, \theta_i)\}_{i=1}^K$ as input, and applies nonlinear optimization to the objective function, starting from the input parameters. The output will be a set of K atoms which constitute a local minimum of the objective function, and with the property that the output set has an equal or lower value of the objective function than the input set.

The ADCG algorithm iteratively updates a set of atoms $\{(w_i, \theta_i)\}$, whose size may change from iteration to iteration. The parameter set is intitialized as the empty set. In each iteration, the gradient subroutine is applied to the current residuals to yield a new parameter θ, which defines the location of a new atom $(0, \theta)$ that is added to the set of atoms. The convex reweighting step minimizes the objective function fixing the parameters θ_i. Having updated the weights w_i, one optionally prunes all atoms with zero weight, Finally, one applies the descent subroutine ν to jointly update the weights and locations of the atoms.

In each iteration, the atoms grows by at most one, due to newly atom from the gradient subroutine. Unless the algorithm has already converged, the newly added atom is sure to acquire a positive weight after the convex reweighting step; hence it will not be pruned. However, one or more of the atoms from previous iterates may be pruned; hence the set of atoms may experience a net decrease in size.

While ADCG is guaranteed to optimize the global minimimum under a convex constraint, one can also consider using the algorithm to optimize the L_0 constraint. Given a constraint on the number of atoms K, one runs ADCG until the number of atoms reaches K. Due to the descent subroutine, the resulting set of atoms is guaranteed to be a local optimum of the objective function. Utilizes this way, ADCG can be thought of as an intelligent way to iteratively build up a good intitialization for the nonlinear descent step in the final iteration. Additionally, if the number of sources K is unknown, one can select for each possible $K = 1, 2, \ldots,$ the ADCG iterate with the best objective function. This allows one to efficiently obtain solutions for the L_0-constrained problem for each candidate value of K.

In this paper, we will focus on the problem of solving the optimization problem subject to an L_0 constraint. We will leave the analysis of the discretization approach to future work[1], and concentrate on comparison of ADCG and the "naive" approach of applying nonlinear descent with random starting conditions. Since the same nonlinear descent algorithm can be used in both ADCG and the random restart approach, this allows the two approaches to be compared "on equal footing": namely, we measure the complexity of each approach by the number of calls to the nonlinear descent algorithm ν. Since the nonlinear descent algorithm is usually the bottleneck in practice, this provides a reasonable estimate of the true computational cost.

In the following section, we present our model and the particular variants of ADCG and nonlinear descent to be studied.

2 Setup

Consider a sparse recovery problem with parameters $\theta \in \mathbb{R}^d$ corresponding to locations of point sources, where the signal a source at θ is given by

$$\psi_\theta(x) = \psi(x - \theta),$$

where $\psi : \mathbb{R}^p \to \mathbb{R}^+$ is a nonnegative kernel function. We assume that ψ is twice-differentiable and has a bounded support in the sense that there exists a bandwidth $h > 0$ such that $\psi(x) = 0$ for all $x \in \mathbb{R}^p$ with $||x|| > h$.

Let the true atoms be given by the set $\{(w_i^0, \theta_i^0)\}_{i=1}^{K^0}$, such that $w_i^0 > 0$. Furthermore, let us assume that θ_i are known *a priori* to lie in a hypercube $[-T, T]^d$. We observe data y_z for each measurement point z in the lattice

[1] While methods for converting an L_1-sparse solution to an L_0-sparse solution exist in the literature (e.g. "peak-finding" in diffusion-weighted imaging), such methods are usually application-specific.

$Z = \{-n, ..., n\}^d \in \mathbb{Z}^d$, where $n >> T$, given by

$$y_z = \epsilon_z + \sum_{i=1}^{K^0} w_i^0 \psi(z - \theta_i^0)$$

where ϵ_z are identically and independently distributed according to a distribution F, with zero mean and unit variance.

The condition that $n >> T$ ensures that the algorithms we study will not be affected by boundary issues: later, we will give exact conditions on the relationship between n and T. In practice, our results should still be approximately correct even if $n = T$.

Consider the problem of minimizing the objective under L_2 loss, given by

$$\mathcal{L}(\{w_i, \theta_i\}) = \frac{1}{2} \sum_{z \in Z} \left\| y_z - \sum_{j=1}^{K} w_j \psi(z - \theta) \right\|^2$$

where K, the number of atoms, is fixed.

We compare the following two approaches:

- *Random restarts.* For a fixed number of iterations $k = 1, \ldots, M$, choose a random starting condition $\{(0, \theta_i^{(k,0)})\}_{i=1}^K$ by drawing θ_i i.i.d. from the uniform distribution on $[-T, T]^d$. Apply nonlinear subroutine ν with the starting condition to obtain local minimum $\{(w_i^{(k,1)}, \theta_i^{(k,1)})\}_{i=1}^K$, and let $\mathcal{L}^{(k)}$ denote the value of the objective function. After M such iterations, return $\{(w_i^{(k)}, \theta_i^{(k)})\}$ with the smallest $o^{(k)}$.
- *ADCG without pruning.* Run the ADCG algorithm without pruning for K iterations, then return the final set of atoms. In the gradient step of ADCG, limit the search to $\theta \in [-T, T]^d$.

For both approaches, we consider *gradient descent* with a fixed step size ϵ_{grad} and a fixed number of steps L_{grad}, which we describe explicitly in the following section.

3 Gradient descent under pure noise

Under the setup described in 2, the gradient of the objective at a particular a set of atoms $\{(w_i, \theta_i)\}$ is given by

$$\frac{\partial \mathcal{L}}{\partial w_i} = -\sum_{z \in Z} r_z(\{(w_i, \theta_i)\}) \psi(z - \theta_i)$$

$$\frac{\partial \mathcal{L}}{\partial \theta_{ij}} = -w_i \sum_{z \in Z} r_z(\{(w_i, \theta_i)\}) \psi_j(z - \theta_i)$$

where ψ_j denotes the jth partial derivative of $\psi(\theta)$, and r_z denotes the residual at z:

$$r_z(\{(w_i, \theta_i)\}) = y_z - \sum_{i=1}^{K} w_i \psi(z - \theta).$$

Given a set of atoms $\{(w_i, \theta_i)\}$, gradient descent produces iterates $\{(w_i^{(k)}, \theta_i^{(k)})\}$ for $k = 1, \ldots, L_{grad}$, defined recusively by

$$w_i^{(k+1)} = \left[w_i^{(k)} - \epsilon_{grad} \frac{\partial \mathcal{L}}{\partial w_i} (\{(w_i^{(k)}, \theta_i^{(k)})\}) \right]_+$$

$$\theta_{ij}^{(k+1)} = \theta_{ij}^{(k)} - \epsilon_{grad} \frac{\partial \mathcal{L}}{\partial \theta_{ij}} (\{(w_i^{(k)}, \theta_i^{(k)})\})$$

for $i = 1, \ldots, K$ and $j = 1, \ldots, d$, and where the initial iterate $\{(w_i^{(0)}, \theta_i^{(0)})\}$ is given by the the input set.

We will need to assume the following property of gradient descent for $K = 1$.

Definition. We say that gradient descent under noise F, kernel ψ, and tuning parameters ϵ_{grad}, L_{grad} satisfies the *bounded path-length condition* if, under the single source model $K = 1$ and under an infinite lattice $Z = \mathbb{Z}^d$, we have the path length $S(\theta)$ finite with probability 1, where $S(\theta)$ is defined as

$$S(\theta) = \sum_{k=1}^{L_{grad}} ||\theta_1^{(k)} - \theta_1^{(k-1)}||$$

given the initial condition $w_i^{(0)} = 0, \theta_i^{(0)} = \theta$, for all $\theta \in \mathbb{R}^p$.

While it is difficult to verify the bounded path-length condition, it usually holds in practice.