Projection-Based Methods in Optimization

Charles Byrne
(Charles_Byrne@uml.edu)
http://faculty.uml.edu/cbyrne/cbyrne.html
Department of Mathematical Sciences
University of Massachusetts Lowell
Lowell, MA 01854, USA

June 16, 2013

Talk Available on Web Site

This slide presentation and accompanying article are available on my web site, http://faculty.uml.edu/cbyrne/cbyrne.html ; click on “Talks”.

The Basic Problem

We want to reconstruct (estimate, approximate) a function $F : \mathbb{R}^J \to \mathbb{C}$, given limited noisy measurements pertaining to F, a (perhaps) simplified model of the measuring process, and some vague prior knowledge of what F should be. There are several issues to address:

- 1. continuous vs discrete F;
- 2. physically realistic model vs easily computed estimate;
- 3. linear vs nonlinear data;
- 4. deterministic vs stochastic approach;
- 5. small vs large problem;
- 6. slow vs fast reconstruction.

Many such problems come from “remote sensing”: what we want is not the same as what we can measure.

An Urns Model for Remote Sensing

We have J urns, each containing marbles of various colors. I know the distribution of colors for each urn. There is a box with many slips of paper, each one marked with one of the urn numbers $j = 1, ..., J$. At each of many trials, my assistant removes one slip of paper from the box, and without revealing the urn number to me, takes one marble from the indicated urn and announces the color. My data is a long list of colors, from which I must estimate the distribution of the numbers j in the box. Urns with nearly the same content are harder to distinguish in a small number of trials: this is a resolution problem.

The Urn Model and Positron Emission Tomography

Instead of urns we have pixels numbered $j = 1, ..., J$. Without telling me, nature selects a pixel and has it release a positron. I learn only the line of response (LOR), which is like learning only the color of the drawn marble. I know the probability that an emission at the jth pixel will lead to any particular LOR, just like I know the probability that the jth urn will yield any particular color of marble. From the list of LORs I must estimate the number of times each pixel was chosen by nature, thereby estimating the radionuclide concentration in each pixel. Nearby pixels tend to have the same LOR probabilities, making them hard to distinguish with relatively few emissions.

Fourier Transform as Data

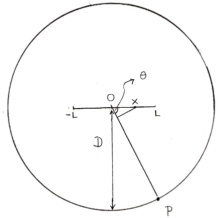

Figure : Far-field Measurements. The distance from x to P is approximately $D - x\cos\theta$.

Far-field Measurements

Each point x in $[-L, L]$ sends out the signal

$$F(x)\exp(i\omega t).$$

The known frequency ω is the same for each x. We must estimate the function $F(x)$, for $|x| \leq L$. What a point P in the far-field receives from each x is approximately

$$F(x)\exp\left(ix\frac{\omega\cos(\theta)}{c}\right),$$

for c the propagation speed, so our measurement at P provides an approximation of

$$\int_{-L}^{L} F(x)\exp\left(ix\frac{\omega\cos(\theta)}{c}\right)dx.$$

If we have

$$\frac{\omega\cos(\theta)}{c} = \frac{n\pi}{L},$$

then we have the nth Fourier coefficient of the function $F(x)$.

The Limited-Data Problem

We can have

$$\cos(\theta) = \frac{n\pi c}{\omega L} = n\frac{\lambda}{2L},$$

where λ is the wavelength, if and only if

$$|n| \leq \frac{2L}{\lambda},$$

so we can measure only a limited number of the Fourier coefficients of $F(x)$. Note that the upper bound on $|n|$ is the length of the interval $[-L, L]$ in units of wavelength.

Can We Get More Data?

Clearly, we can take our measurements at any point P on the circle, not just at those satisfying the equation

$$\cos(\theta) = \frac{n\pi c}{\omega L} = n\frac{\lambda}{2L}.$$

These measurements provide additional information about $F(x)$, but won't be additional Fourier coefficients for the Fourier series of $F(x)$ on $[-L, L]$. How can we use these additional measurements to improve our estimate of $F(x)$?

Over-Sampled Data

Suppose that we **over-sample**; let us take measurements at points P such that the associated angle θ satisfies

$$\cos(\theta) = \frac{n\pi c}{\omega KL},$$

where $K > 1$ is a positive integer, instead of

$$\cos(\theta) = \frac{n\pi c}{\omega L}.$$

Now we have Fourier coefficients for the function $G(x)$ that is $F(x)$ for $|x| \leq L$, and is zero on the remainder of the interval $[-KL, KL]$.

Using Support Information

Given our original limited data, we can calculate the orthogonal projection of the zero function onto the subset of all functions on $[-L, L]$ consistent with this data; this is the minimum-norm estimate of $F(x)$, also called the discrete Fourier transform (DFT) estimate, shown in the second graph below.

If we use the over-sampled data as Fourier coefficients for $G(x)$ on the interval $[-KL, KL]$, we find that we haven't improved our estimate of $F(x)$ for $|x| \leq L$.

Instead, we calculate the orthogonal projection of the zero function onto the subset of all functions on $[-L, L]$ that are consistent with the over-sampled data. This is sometimes called the *modified* DFT (MDFT) estimate. The top graph below shows the MDFT for a case of $K = 30$.

Two Minimum-Norm Estimates

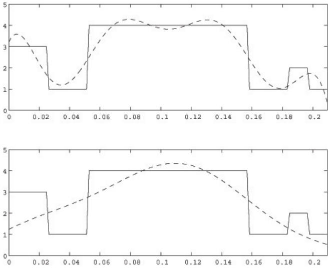

Figure : The non-iterative band-limited extrapolation method (MDFT) (top) and the DFT (bottom); 30 times over-sampled.

Over-Sampled Fourier-Transform Data

For the simulation in the figure above, $f(x) = 0$ for $|x| > L$. The top graph is the minimum-norm estimator, with respect to the Hilbert space $L^2(-L, L)$, called the *modified* DFT (MDFT); the bottom graph is the DFT, the minimum-norm estimator with respect to the Hilbert space $L^2(-30L, 30L)$, shown only for $[-L, L]$. The MDFT is a non-iterative variant of Gerchberg-Papoulis band-limited extrapolation.

Using Orthogonal Projection

In both of the examples above we see minimum two-norm solutions consistent with the data. These reconstructions involve the orthogonal projection of the zero vector onto the set of solutions consistent with the data. The improvements illustrate the advantage gained by the selection of an appropriate ambient space within which to perform the projection. The constraints of data consistency define a subset onto which we project, and the distance to zero is the function being minimized, subject to the constraints. This leads to the more general problem of optimizing a function, subject to constraints.

The Discrete Problem

From now on, we treat the discrete case, in which the object to be estimated is a vector x in $\mathbb{R}^J$. Our data is insufficient to determine a unique x. One approach is to select a prior estimate p of x and to project p onto the set of vectors consistent with the data. This approach is equivalent to minimizing a distance function over the constraint set of all data-consistent vectors. We can incorporate other prior knowledge about x in the constraint set, or in the choice of the distance function to be minimized.

Optimization

The basic problem we consider here is to minimize a real-valued function

$$f : X \to \mathbb{R},$$

over a subset $C \subseteq X$, where X is an arbitrary set. With

$$\iota_C(x) = 0, \text{for } x \in C, \text{and } + \infty, \text{otherwise},$$

we can rewrite the problem as minimizing $f(x) + \iota_C(x)$, over all $x \in X$.

Constraints

We want our reconstruction to be at least approximately consistent with the measured data. We may know that the entries of x are non-negative. When x is a vectorized image, we may have prior knowledge of its general appearance (a head slice, for example). If x is an extrapolated band-limited time series, we may have prior knowledge of the extent of the band.

Representing Constraints

Prior knowledge of general properties of x can be incorporated through the choice of the ambient space. Other constraints, such as the measured data, tell us that x lies in a subset C of the ambient space.

An Example: Linear Functional Data

Suppose that the measured data vector b is linear in x, with $Ax = b$ under-determined. The minimum two-norm solution minimizes

$$\sum_{j=1}^{J} |x_j|^2,$$

subject to $Ax = b$. Let the vector p with entries $p_j > 0$ be a prior estimate of the magnitudes $|x_j|$. The minimum weighted two-norm solution minimizes

$$\sum_{j=1}^{J} |x_j|^2 / p_j,$$

subject to $Ax = b$.

Using Prior Knowledge

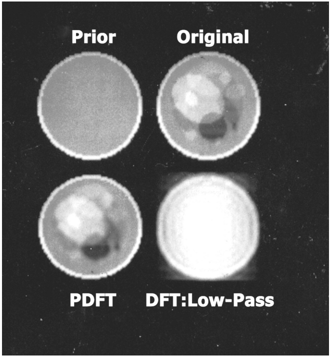

Figure : Minimum Two-Norm and Minimum Weighted Two-Norm Reconstruction.

Using Optimization

When the constraint set C is relatively small, any member of C may provide an adequate solution to the problem. More likely, C is relatively large, and we may choose to determine x by minimizing a cost function over the set C.

An Example- Systems of Linear Equations

Let the data b pertaining to x be linear, so that $Ax = b$ for some matrix A.

- **1.** When there are infinitely many solutions, we may select x to minimize a norm or other suitable distance measure;
- **2.** When there are no solutions, we may select a least-squares or other approximate solution;
- **3.** When the data b is noisy, we may select a regularized solution.

Barrier Functions: An Example

The problem is to minimize the function

$$f(x) = f(x_1, x_2) = x_1^2 + x_2^2, \text{subject to } x_1 + x_2 \geq 1.$$

For each positive integer k, the vector x^k with entries

$$x_1^k = x_2^k = \frac{1}{4} + \frac{1}{4}\sqrt{1 + \frac{4}{k}}$$

minimizes the function

$$B_k(x) = x_1^2 + x_2^2 - \frac{1}{k}\log(x_1 + x_2 - 1) = f(x) + \frac{1}{k}b(x).$$

Notice that $x_1^k + x_2^k > 1$, so each x^k satisfies the constraint. As $k \to +\infty$, x^k converges to $(\frac{1}{2}, \frac{1}{2})$, which solves the original problem.

Penalty Functions: An Example

The problem is to minimize the function

$$f(x) = (x+1)^2, \text{subject to } x \geq 0.$$

Our penalty function is

$$p(x) = x^2, \text{for } x \leq 0,\ p(x) = 0, \text{for } x > 0.$$

At the kth step we minimize

$$f(x) + kp(x)$$

to get $x^k = \frac{-1}{k+1}$, which converges to the right answer, $x^* = 0$, as $k \to \infty$. The limit x^* satisfies the constraint, but the x^k do not; this is an *exterior-point* method.

Auxiliary-Function Methods

The Problem: to minimize a function $f : X \to (-\infty, \infty]$, over a non-empty subset C of X, where X is an arbitrary set.

Auxiliary-Function Methods: At the kth step of an auxiliary-function (AF) algorithm we minimize a function

$$G_k(x) = f(x) + g_k(x)$$

over $x \in C$ to get x^k. Auxiliary functions g_k have the properties

- 1. $g_k(x) \geq 0, \text{for all } x \in C$;
- 2. $g_k(x^{k-1}) = 0$.

Main Theorem for Auxiliary-Function Algorithms

We have the following theorem.

Theorem

Let $\{x^k\}$ be generated by an AF algorithm. Then the sequence $\{f(x^k)\}$ is non-increasing.

Proof: We have

$$f(x^{k-1}) = G_k(x^{k-1}) \geq G_k(x^k) \geq f(x^k).$$

∎

Auxiliary-function algorithms are closely related to **sequential unconstrained minimization** (SUM) methods. Several SUM methods, such as barrier-function and penalty-function methods, can be reformulated as AF methods.

GOALS:

The vector x^k minimizes

$$G_k(x) = f(x) + g_k(x), \text{over } x \in C.$$

- **1.** We would like for the sequence $\{x^k\}$ to converge to some $x^* \in C$ that solves the problem. This requires a topology on X.
- **2.** Failing that, we would like the sequence $\{f(x^k)\}$ to converge to
$$d = \inf\{f(x) | x \in C\}.$$
- **3.** At the very least, we want the sequence $\{f(x^k)\}$ to be non-increasing.

SUMMA

An AF algorithm in said to be in the SUMMA class if

$$G_k(x) - G_k(x^k) \geq g_{k+1}(x) \geq 0,$$

for all $x \in C$.

Theorem

For algorithms in the SUMMA class, the sequence $\{f(x^k)\}$ is non-increasing, and we have

$$\{f(x^k)\} \downarrow d = \inf\{f(x) | x \in C\}.$$

SUMMA Proof:

If $f(x^k) \geq d^* > f(z) \geq d$ for all k, then

$$g_k(z) - g_{k+1}(z) \geq g_k(z) - (G_k(z) - G_k(x^k))$$

$$= f(x^k) - f(z) + g_k(x^k) \geq d^* - f(z) > 0.$$

But the decreasing non-negative sequence $\{g_k(z)\}$ cannot have its successive differences bounded away from zero.

Examples of SUMMA

A number of well known algorithms either are in the SUMMA class, or can be reformulated to be in the SUMMA class, including

- **1.** Barrier-function methods;
- **2.** Penalty-function methods;
- **3.** Forward-backward splitting (the CQ algorithm, projected Landweber, projected gradient descent);
- **4.** Alternating minimization with the 5-point property (simultaneous MART);
- **5.** Certain cases of the EM algorithm;
- **6.** Proximal minimization with Bregman distances;
- **7.** Statistical majorization minimization.

Barrier-Function Methods

A function $b : C \to [0, +\infty]$ is a barrier function for C, that is, b is finite on C, $b(x) = +\infty$, for x not in C, and, for topologized X, has the property that $b(x) \to +\infty$ as x approaches the boundary of C. At the kth step of the iteration we minimize

$$f(x) + \frac{1}{k} b(x)$$

to get x^k. Equivalently, we minimize

$$G_k(x) = f(x) + g_k(x),$$

with

$$g_k(x) = [(k-1)f(x) + b(x)] - [(k-1)f(x^{k-1}) + b(x^{k-1})].$$

Then

$$G_k(x) - G_k(x^k) = g_{k+1}(x).$$

Penalty-Function Methods

We select a non-negative function $p : X \to \mathbb{R}$ with the property that $p(x) = 0$ if and only if x is in C and then, for each positive integer k, we minimize

$$f(x) + kp(x),$$

or, equivalently,

$$p(x) + \frac{1}{k}f(x),$$

to get x^k. Most, but not all, of what we need concerning penalty-function methods can then be obtained from the discussion of barrier-function methods.

Bregman Distances

Let $f : Z \subseteq \mathbb{R}^J \to \mathbb{R}$ be convex on its domain Z, and differentiable on the interior U of Z. For $x \in Z$ and $y \in U$, the Bregman distance from y to x is

$$D_f(x, y) = f(x) - f(y) - \langle \nabla f(y), x - y \rangle.$$

Then, because of the convexity of f, $D_f(x, y) \geq 0$ for all x and y.

The Euclidean distance is the Bregman distance for

$$f(x) = \frac{1}{2}\|x\|_2^2.$$

Proximal Minimization Algorithms (PMA)

Let $f : C \subseteq \mathbb{R}^J \to \mathbb{R}$ be convex on C and differentiable on the interior U of C. At the kth step of a proximal minimization (prox min) algorithm (PMA), we minimize the function

$$G_k(x) = f(x) + D_h(x, x^{k-1}),$$

to get x^k. The function $g_k(x) = D_h(x, x^{k-1})$ is the Bregman distance associated with the Bregman function h. We assume that each x^k lies in U, whose closure is the set C. We have

$$G_k(x) - G_k(x^k) = D_f(x, x^k) + D_h(x, x^k) \geq D_h(x, x^k) = g_{k+1}(x).$$

Another Job for the PMA

Suppose that

$$f(x) = \frac{1}{2}\|Ax - b\|_2^2,$$

with AA^T invertible. If the PMA sequence $\{x^k\}$ converges to some x^* and $\nabla h(x^0)$ is in the range of A^T then x^* minimizes $h(x)$ over all x with $Ax = b$.

A Question About the PMA

Suppose that x^k minimizes

$$f(x) + D_h(x, x^{k-1}),$$

and $\{x^k\}$ converges to x^*. We know that x^* minimizes $f(x)$ over all x in the closure of the essential domain of h. Let M be the set of all such minimizers. Does x^* also minimize $h(z)$ over all z in M? In general, the answer is no; D_h does not determine h uniquely. What if $h(x) = D_h(x, x^0)$? There are several examples, using both Euclidean and Kullback-Leibler distances, in which the answer is yes.

Obstacles in the PMA

To obtain x^k in the PMA we must solve the equation

$$\nabla f(x^k) + \nabla h(x^k) = \nabla h(x^{k-1}).$$

This is usually not easy. However, we can modify the PMA to overcome this obstacle. This modified PMA is an interior-point algorithm that we have called the IPA.

The IPA

We still get x^k by minimizing

$$G_k(x) = f(x) + D_h(x, x^{k-1}).$$

With

$$a(x) = h(x) + f(x),$$

we find that x^k solves

$$\nabla a(x^k) = \nabla a(x^{k-1}) - \nabla f(x^{k-1}).$$

Therefore, we look for $a(x)$ so that

- **1.** $h(x) = a(x) - f(x)$ is convex;
- **2.** obtaining x^k from $\nabla a(x^k)$ is easy now.

The IPA and Constraints

In the PMA, h can be chosen to force x^k to be in U, the interior of the domain of h. In the IPA it is $a(x)$ that we select, and incorporating the constraints within $a(x)$ may not be easy.

An Example of the IPA: Projected Gradient Descent

Let f be convex and differentiable on $\mathbb{R}^J$, with ∇f L-Lipschitz, and $0 < \gamma \leq \frac{1}{L}$. Let $C \subseteq \mathbb{R}^J$ be closed, nonempty, and convex. Let

$$a(x) = \frac{1}{2\gamma}\|x\|_2^2.$$

At the kth step we minimize

$$G_k(x) = f(x) + \frac{1}{2\gamma}\|x - x^{k-1}\|_2^2 - D_f(x, x^{k-1}),$$

over $x \in C$, obtaining

$$x^k = P_C(x^{k-1} - \gamma\nabla f(x^{k-1}));$$

P_C is the orthogonal projection onto C.

Projected Gradient Descent as SUMMA

The auxiliary function $g_k(x)$ can be written as

$$g_k(x) = \frac{1}{2\gamma}\|x - x^{k-1}\|_2^2 - D_f(x, x^{k-1}) = D_h(x, x^{k-1}),$$

where $h(x)$ is the convex differentiable function

$$h(x) = a(x) - f(x) = \frac{1}{2\gamma}\|x\|_2^2 - f(x).$$

Then

$$G_k(x) - G_k(x^k) = D_a(x, x^k) \geq D_h(x, x^k) = g_{k+1}(x).$$

The Projected Landweber Algorithm as IPA

We want to

$$\text{minimize } f(x) = \frac{1}{2}\|Ax - b\|_2^2, \text{ over } x \in C.$$

We select γ so that $0 < \gamma < \frac{1}{\rho(A^TA)}$ and

$$a(x) = \frac{1}{2\gamma}\|x\|_2^2.$$

We have

$$D_f(x, y) = \frac{1}{2}\|Ax - Ay\|_2^2,$$

and we minimize

$$G_k(x) = f(x) + \frac{1}{2\gamma}\|x - x^{k-1}\|_2^2 - \frac{1}{2}\|Ax - Ax^{k-1}\|_2^2$$

over x in C to get

$$x^k = P_C(x^{k-1} - \gamma A^T(Ax^{k-1} - b)).$$

Majorization Minimization or Optimization Transfer

Majorization minimization or **optimization transfer** is used in statistical optimization.

For each fixed y, let $g(x|y) \geq f(x)$, for all x, and $g(y|y) = f(y)$. At the kth step we minimize $g(x|x^{k-1})$ to get x^k. This statistical method is equivalent to minimizing

$$f(x) + D(x, x^{k-1}) = f(x) + g_k(x),$$

where

$$g_k(x) = D(x, x^{k-1}),$$

for some distance measure $D(x, z) \geq 0$, with $D(z, z) = 0$.

The Method of Auslander and Teboulle

The method of Auslander and Teboulle is a particular example of an MM method that is not in SUMMA. At the *k*th step of their method

$$\text{minimize } G_k(x) = f(x) + D(x, x^{k-1}) \text{ to get } x^k.$$

They assume that D has an associated *induced proximal distance* $H(a, b) \geq 0$, finite for a and b in C, with $H(a, a) = 0$ and

$$\langle \nabla_1 d(b, a), c - b \rangle \leq H(c, a) - H(c, b),$$

for all c in C. Here $\nabla_1 D(x, y)$ denotes the x gradient. We do have $\{f(x^k)\} \to d$. If $D = D_h$, then $H = D_h$ also; Bregman distances are self-proximal.

Projected Gradient Descent (PGD) Revisited

At the kth step of the PGD we minimize

$$G_k(x) = f(x) + \frac{1}{2\gamma}\|x - x^{k-1}\|_2^2 - D_f(x, x^{k-1}),$$

over $x \in C$. Equivalently, we minimize the function

$$\iota_C(x) + f(x) + \frac{1}{2\gamma}\|x - x^{k-1}\|_2^2 - D_f(x, x^{k-1}),$$

over all x in $\mathbb{R}^J$, with $\iota_C(x) = 0$ for $x \in C$ and $\iota_C(x) = +\infty$ otherwise. Now the objective function is a sum of two functions, one non-differentiable. The forward-backward splitting method then applies.

Moreau's Proximity Operators

Let $f : \mathbb{R}^J \to \mathbb{R}$ be convex. For each $z \in \mathbb{R}^J$ the function

$$m_f(z) := \min_x \{ f(x) + \frac{1}{2} \|x - z\|_2^2 \}$$

is minimized by a unique $x = \mathrm{prox}_f(z)$. Moreau's proximity operator prox_f extends the notion of orthogonal projection onto a closed convex set: if $f(x) = \iota_C(x)$ then $\mathrm{prox}_f(x) = P_C(x)$. Also

$$x = \mathrm{prox}_f(z) \,\mathrm{iff}\, z - x \in \partial f(x) \,\mathrm{iff}\, x = J_{\partial f}(z),$$

where $J_{\partial f}(z)$ is the resolvent of the set-valued operator ∂f.

Forward-Backward Splitting

Let $f : \mathbb{R}^J \to \mathbb{R}$ be convex, with $f = f_1 + f_2$, both convex, f_2 differentiable, and ∇f_2 L-Lipschitz continuous. The iterative step of the FBS algorithm is

$$x^k = \operatorname{prox}_{\gamma f_1}\left(x^{k-1} - \gamma \nabla f_2(x^{k-1})\right),$$

which can be obtained by minimizing

$$G_k(x) = f(x) + \frac{1}{2\gamma}\|x - x^{k-1}\|_2^2 - D_{f_2}(x, x^{k-1}).$$

Convergence of the sequence $\{x^k\}$ to a solution can be established, if γ is chosen to lie within the interval $(0, 1/L]$.

Projected Gradient Descent as FBS

To put the projected gradient descent method into the framework of the forward-backward splitting we let

$$f_1(x) = \iota_C(x),$$

the **indicator function** of the set C, which is zero for $x \in C$ and $+\infty$ otherwise. Then we minimize the function

$$f_1(x) + f_2(x) = \iota_C(x) + f(x)$$

over all $x \in \mathbb{R}^J$.

The *CQ* Algorithm

Let A be a real I by J matrix, $C \subseteq \mathbb{R}^J$, and $Q \subseteq \mathbb{R}^I$, both closed convex sets. The **split feasibility problem** (SFP) is to find x in C such that Ax is in Q. The function

$$f_2(x) = \frac{1}{2}\|P_Q Ax - Ax\|_2^2$$

is convex and differentiable, ∇f_2 is $\rho(A^T A)$-Lipschitz, and

$$\nabla f_2(x) = A^T(I - P_Q)Ax.$$

We want to minimize the function $f(x) = \iota_C(x) + f_2(x)$, over all x. The FBS algorithm gives the iterative step for the *CQ* algorithm; with $0 < \gamma \leq 1/L$,

$$x^k = P_C\Big(x^{k-1} - \gamma A^T(I - P_Q)Ax^{k-1}\Big).$$

Alternating Minimization

Suppose that P and Q are arbitrary non-empty sets and the function $\Theta(p, q)$ satisfies $-\infty < \Theta(p, q) \leq +\infty$, for each $p \in P$ and $q \in Q$.
The general AM method proceeds in two steps: we begin with some q^0, and, having found q^{k-1}, we

- **1.** minimize $\Theta(p, q^{k-1})$ over $p \in P$ to get $p = p^k$, and then
- **2.** minimize $\Theta(p^k, q)$ over $q \in Q$ to get $q = q^k$.

The 5-**point property** of Csiszár and Tusnády is

$$\Theta(p, q) + \Theta(p, q^{k-1}) \geq \Theta(p, q^k) + \Theta(p^k, q^{k-1}).$$

AM as SUMMA

When the 5-point property holds for AM, the sequence $\{\Theta(p^k, q^k)\}$ converges to

$$d = \inf_{p,q} \Theta(p, q).$$

For each $p \in P$, define $q(p)$ to be some member of Q for which $\Theta(p, q(p)) \leq \Theta(p, q)$, for all $q \in Q$; then $q(p^k) = q^k$. Define

$$f(p) = \Theta(p, q(p)).$$

At the kth step of AM we minimize

$$G_k(p) = \Theta(p, q^{k-1}) = f(p) + g_k(p),$$

where

$$g_k(p) = \Theta(p, q^{k-1}) - \Theta(p, q(p)).$$

The 5-point property is then the SUMMA condition

$$G_k(p) - G_k(p^k) \geq g_{k+1}(p) \geq 0.$$

Kullback-Leibler Distances

For $a > 0$ and $b > 0$, the **Kullback-Leibler distance** from a to b is

$$KL(a, b) = a \log a - a \log b + b - a,$$

with $KL(0, b) = b$, and $KL(a, 0) = +\infty$. The KL distance is extended to non-negative vectors entry-wise.

Let $y \in \mathbb{R}^I$ be positive, and P an I by J matrix with non-negative entries. The **simultaneous MART** (SMART) algorithm minimizes $KL(Px, y)$, and the **EMML algorithm** minimizes $KL(y, Px)$, both over all non-negative $x \in \mathbb{R}^J$. Both algorithms can be viewed as particular cases of alternating minimization.

The General EM Algorithm

The general EM algorithm is essentially non-stochastic. Let Z be an arbitrary set. We want to maximize

$$L(z) = \int b(x,z)dx, \text{over } z \in Z,$$

where $b(x,z) : \mathbb{R}^J \times Z \to [0,+\infty]$. Given z^{k-1}, we take $f(z) = -L(z)$ and

$$\text{minimize } G_k(z) = f(z) + \int KL(b(x,z^{k-1}), b(x,z))dx \text{ to get } z^k.$$

Since

$$g_k(z) = \int KL(b(x,z^{k-1}), b(x,z))dx \geq 0,$$

for all $z \in Z$ and $g_k(z^{k-1}) = 0$, we have an AF method.

KL Projections

For a fixed x, minimizing the distance $KL(z, x)$ over z in the hyperplane

$$H_i = \{z|(Pz)_i = y_i\}$$

generally cannot be done in closed form. However, assuming that $\sum_{i=1}^{I} P_{ij} = 1$, **the weighted KL projections** $z^i = T_i(x)$ onto hyperplanes H_i obtained by minimizing

$$\sum_{j=1}^{J} P_{ij} KL(z_j, x_j)$$

over z in H_i, **are given in closed form** by

$$z_j^i = T_i(x)_j = x_j \frac{y_i}{(Px)_i}, \text{ for } j = 1, ..., J.$$

SMART and EMML

Having found x^k, **the next vector in the SMART sequence** is

$$x_j^{k+1} = \prod_{i=1}^{I} (T_i(x^k)_j)^{P_{ij}},$$

so x^{k+1} is a **weighted geometric mean** of the $T_i(x^k)$.

For the EMML algorithm **the next vector in the EMML sequence** is

$$x_j^{k+1} = \sum_{i=1}^{I} P_{ij} T_i(x^k)_j,$$

so x^{k+1} is a **weighted arithmetic mean** of the $T_i(x^k)$.

The MART

The MART algorithm has the iterative step

$$x_j^{k+1} = x_j^k (y_i/(Px^k)_i)^{P_{ij} m_i^{-1}},$$

where $i = k (\mathrm{mod}\, I) + 1$ and

$$m_i = \max\{P_{ij} | j = 1, 2, ..., J\}.$$

We can express the MART in terms of the weighted KL projections $T_i(x^k)$;

$$x_j^{k+1} = (x_j^k)^{1 - P_{ij} m_i^{-1}} (T_i(x^k)_j)^{P_{ij} m_i^{-1}}.$$

We see then that the iterative step of the MART is a relaxed weighted KL projection onto H_i, and a weighted **geometric** mean of the current x_j^k and $T_i(x^k)_j$.

The EMART

The iterative step of the EMART algorithm is

$$x_j^{k+1} = (1 - P_{ij} m_i^{-1}) x_j^k + P_{ij} m_i^{-1} T_i(x^k)_j.$$

We can express the EMART in terms of the weighted KL projections $T_i(x^k)$;

$$x_j^{k+1} = (1 - P_{ij} m_i^{-1}) x_j^k + P_{ij} m_i^{-1} T_i(x^k)_j.$$

We see then that the iterative step of the EMART is a relaxed weighted KL projection onto H_i, and a weighted **arithmetic** mean of the current x_j^k and $T_i(x^k)_j$.

MART and EMART Compared

When there are non-negative solutions of the system $y = Px$, the MART sequence $\{x^k\}$ converges to the solution x that minimizes $KL(x, x^0)$. The EMART sequence $\{x^k\}$ also converges to a non-negative solution, but nothing further is known about this solution. One advantage that the EMART has over the MART is the substitution of multiplication for exponentiation.

SMART as SUMMA

At the kth step of the SMART we minimize the function

$$G_k(x) = KL(Px, y) + KL(x, x^{k-1}) - KL(Px, Px^{k-1})$$

to get x^k with entries

$$x_j^k = x_j^{k-1} \exp\Big(\sum_{i=1}^{I} P_{ij} \log(y_i/(Px^{k-1})_i)\Big).$$

We assume that P and x have been rescaled so that $\sum_{i=1}^{I} P_{ij} = 1$, for all j. Then

$$g_k(x) = KL(x, x^{k-1}) - KL(Px, Px^{k-1})) \geq 0.$$

and

$$G_k(x) - G_k(x^k) = KL(x, x^k) \geq g_{k+1}(x) \geq 0.$$

SMART as IPA

With

$$f(x) = KL(Px, y),$$

the associated Bregman distance is

$$D_f(x, z) = KL(Px, Pz).$$

With

$$a(x) = \sum_{j=1}^{J} x_j \log(x_j) - x_j,$$

we have

$$D_a(x, z) = KL(x, z) \geq KL(Px, Pz) = D_f(x, z).$$

Therefore, $h(x) = a(x) - f(x)$ is convex.

Convergence of the FBS

For each $k = 1, 2, ...$ let

$$G_k(x) = f(x) + \frac{1}{2\gamma}\|x - x^{k-1}\|_2^2 - D_{f_2}(x, x^{k-1}),$$

where

$$D_{f_2}(x, x^{k-1}) = f_2(x) - f_2(x^{k-1}) - \langle \nabla f_2(x^{k-1}), x - x^{k-1} \rangle.$$

Here $D_{f_2}(x, y)$ is the Bregman distance formed from the function f_2. The auxiliary function

$$g_k(x) = \frac{1}{2\gamma}\|x - x^{k-1}\|_2^2 - D_{f_2}(x, x^{k-1})$$

can be rewritten as

$$g_k(x) = D_h(x, x^{k-1}),$$

where

$$h(x) = \frac{1}{2\gamma}\|x\|_2^2 - f_2(x).$$

Proof (p.2):

Therefore, $g_k(x) \geq 0$ whenever $h(x)$ is a convex function. We know that $h(x)$ is convex if and only if

$$\langle \nabla h(x) - \nabla h(y), x - y \rangle \geq 0,$$

for all x and y. This is equivalent to

$$\frac{1}{\gamma}\|x - y\|_2^2 - \langle \nabla f_2(x) - \nabla f_2(y), x - y \rangle \geq 0.$$

Since ∇f_2 is L-Lipschitz, the inequality holds for $0 < \gamma \leq 1/L$.

Proof (p.3):

Lemma

The x^k that minimizes $G_k(x)$ over x is given by

$$x^k = \operatorname{prox}_{\gamma f_1}\left(x^{k-1} - \gamma \nabla f_2(x^{k-1})\right).$$

Proof: We know that x^k minimizes $G_k(x)$ if and only if

$$0 \in \nabla f_2(x^k) + \frac{1}{\gamma}(x^k - x^{k-1}) - \nabla f_2(x^k) + \nabla f_2(x^{k-1}) + \partial f_1(x^k),$$

or, equivalently,

$$\left(x^{k-1} - \gamma \nabla f_2(x^{k-1})\right) - x^k \in \partial(\gamma f_1)(x^k).$$

Consequently,

$$x^k = \operatorname{prox}_{\gamma f_1}(x^{k-1} - \gamma \nabla f_2(x^{k-1})).$$

Proof (p.4):

Theorem

The sequence $\{x^k\}$ converges to a minimizer of the function $f(x)$, whenever minimizers exist.

Proof: $G_k(x) - G_k(x^k) = \frac{1}{2\gamma}\|x - x^k\|_2^2 +$

$$\left(f_1(x) - f_1(x^k) - \frac{1}{\gamma}\langle (x^{k-1} - \gamma \nabla f_2(x^{k-1})) - x^k, x - x^k \rangle\right)$$

$$\geq \frac{1}{2\gamma}\|x - x^k\|_2^2 \geq g_{k+1}(x),$$

because

$$(x^{k-1} - \gamma \nabla f_2(x^{k-1})) - x^k \in \partial(\gamma f_1)(x^k).$$

Proof (p.5):

Therefore,

$$G_k(x) - G_k(x^k) \geq \frac{1}{2\gamma}\|x - x^k\|_2^2 \geq g_{k+1}(x),$$

and the iteration fits into the SUMMA class. Now let $\hat{x}$ minimize $f(x)$ over all x. Then

$$G_k(\hat{x}) - G_k(x^k) = f(\hat{x}) + g_k(\hat{x}) - f(x^k) - g_k(x^k)$$

$$\leq f(\hat{x}) + G_{k-1}(\hat{x}) - G_{k-1}(x^{k-1}) - f(x^k) - g_k(x^k),$$

so that

$$\Big(G_{k-1}(\hat{x}) - G_{k-1}(x^{k-1})\Big) - \Big(G_k(\hat{x}) - G_k(x^k)\Big)$$

$$\geq f(x^k) - f(\hat{x}) + g_k(x^k) \geq 0.$$

Proof (p.6):

Therefore, the sequence $\{G_k(\hat{x}) - G_k(x^k)\}$ is decreasing and the sequences $\{g_k(x^k)\}$ and $\{f(x^k) - f(\hat{x})\}$ converge to zero. From

$$G_k(\hat{x}) - G_k(x^k) \geq \frac{1}{2\gamma}\|\hat{x} - x^k\|_2^2,$$

it follows that the sequence $\{x^k\}$ is bounded. Therefore, we may select a subsequence $\{x^{k_n}\}$ converging to some x^{**}, with $\{x^{k_n-1}\}$ converging to some x^*, and therefore $f(x^*) = f(x^{**}) = f(\hat{x})$. Replacing the generic $\hat{x}$ with x^{**}, we find that $\{G_k(x^{**}) - G_k(x^k)\}$ is decreasing to zero. We conclude that the sequence $\{\|x^* - x^k\|_2^2\}$ converges to zero, and so $\{x^k\}$ converges to x^*. This completes the proof of the theorem. ∎

Summary

- **1.** Auxiliary-function methods can be used to impose constraints, but also to provide closed-form iterations.
- **2.** When the SUMMA condition holds, the iterative sequence converges to the infimum of $f(x)$ over C.
- **3.** A wide variety of iterative methods fall into the SUMMA class.
- **4.** The alternating minimization method is an auxiliary-function method and the 5-point property is identical to the SUMMA condition.
- **6.** The SUMMA framework helps in proving convergence.

The End

THE END