ACM 113 Homework 3

November 16, 2007

Problem 1

Using Matlab and cvx, this experiment can be done as follows:

```
m = 512; n = 128;
A = randn(m,n);
x0 = randn(n,1);
er_ind = randsample(m,round(m*.1));
y = A*x0;
y(er_ind) = -2*y(er_ind);

cvx_begin
    variable x(n);
    minimize(norm(y-A*x,1));
cvx_end

norm(x-x0,Inf)
```

In this code, we corrupt entries by multiplying by -2. However, it makes no difference how we corrupt entries. We can replace the entries with an arbitrary fixed value, add or multiply by fixed or random numbers; as long as the entries are not so large they will cause round-off errors, the algorithm will work. That is, we always observe that the computed reconstruction error $\|x^* - x_0\|_\infty$ is within the tolerance of the optimization algorithm.

In fact, if we could do all our calculations in exact arithmetic, the we would have $x^* = x_0$ with overwhelming probability (depending on the random matrix A). This surprising result is one of many of the new field of Compressed Sensing. This works because the error vector is extremely sparse (that is, it has relatively few nonzero elements), and the ℓ_1 norm is the best convex function for promoting sparsity. Of course, this will not work for arbitrary A. The encoding matrix needs special properties, basically that certain submatrices are well-conditioned, which is true with very high probability for random matrices.

Problem 2

2.1

As we have seen in problem 1, minimizing the ℓ_1 norm of a vector tends to find a sparse vector. So the program translates to: "Find the sparsest vector consistent given the noise level". That is, we allow a noise vector which is uncorrelated with the columns of A. (ie., $A^*z = 0$), and its length (ℓ_2 norm) is consistent with what you would expect from the noise level σ. The restriction $A^*z = 0$ guarantees that the vector Ax and the vector z will be orthogonal, so that for fixed y and e, the the decomposition $y - e = Ax + z$ will be unique in terms of x and z

We can recast this as an SOCP in at least 2 different ways. One by introducing a variable e which bounds the absolute value of $y - Ax - z$.

$$\begin{array}{lll} \min 1^T e & \text{subject to} & -e \leq y - Ax - z \leq e \\ & & \|z\|_2 \leq \epsilon \\ & & A^* z = 0 \end{array}$$

Alternatively, we can break apart the gross error into positive and negative parts

$$\begin{array}{lll} \min 1^T e^+ + 1^T e^- & \text{subject to} & y - Ax - z = e^+ - e^- \\ & & e^+ \geq 0, e^- \geq 0 \\ & & \|z\|_2 \leq \epsilon \\ & & A^* z = 0 \end{array}$$

In either case, we have a linear objective function and linear or second-order cone constraints, so this is an SOCP.

2.2

This reasoning here is similar as for P_2, but in this case we restrict the maximum absolute value of z to be less than what you would expect to be within the noise level. One subtle difference though, is that in this case it is essential that we restrict z to be uncorrelated with the columns of A. (That is, the algorithm P_1 will perform poorly without the constraint, whereas P_2 is not affected too much.)

We can recast this as an LP in at least 2 different ways. One by introducing a variable e which bounds the absolute value of $y - Ax - z$.

$$\begin{array}{lll} \min 1^T e & \text{subject to} & -e \leq y - Ax - z \leq e \\ & & -\delta 1 \leq z \leq \delta 1 \\ & & A^* z = 0 \end{array}$$

Alternatively, we can break apart the gross error into positive and negative parts

$$\begin{array}{lll} \min 1^T e^+ + 1^T e^- & \text{subject to} & y - Ax - z = e^+ - e^- \\ & & e^+ \geq 0, e^- \geq 0 \\ & & -\delta 1 \leq z \leq \delta 1 \\ & & A^* z = 0 \end{array}$$

In either case, we have a linear objective function and linear constraints, so this is an LP.

2.3

First of all, we assume that the components of z are i.i.d. Gaussian variables, with mean 0 and variance σ^2.

A general result in statistics is that the estimator which minimizes mean square error is the conditional expectation. Briefly, one can see this through the following manipulation:

$$\begin{aligned} E\|\hat{x}(y) - x\|^2 &= E_y \left[E\left(\|\hat{x}(y) - x\|^2 | y \right) \right] \\ &= E_y \left[\left\{ \|\hat{x}(y)\|^2 - 2\hat{x}(y)^T E(x|y) + \|E(x|y)\|^2 \right\} + \left\{ E(\|x\|^2 | y) - \|E(x|y)\|^2 \right\} \right] \\ &= E_y \left[\|\hat{x}(y) - E(x|y)\|^2 + E\left(\|x - E(x|y)\|^2 | y \right) \right] \end{aligned}$$

So the optimal choice of $\hat{x}(y)$ is to let $\hat{x}(y) = E(x|y)$ for all y.

In our case, we have $y = Ax_0 + z$ which implies $x_0 = (A^TA)^{-1}A^T(y - z)$. So the optimal estimator $\hat{x}(y)$ is given by

$$\hat{x}(y) = E(x_0|y) = E\left[(A^TA)^{-1}A^T(y - z)|y\right] = (A^TA)^{-1}A^Ty$$

Where we use linearity of expecation and the fact that $E(z|y) = E(z) = 0$. The expected mean square error with the optimal estimator is thus:

$$\begin{aligned}
E\left\|\hat{x}(y) - x_0\right\|^2 &= E\left\|(A^TA)^{-1}A^Ty - x_0\right\|^2 \\
&= E\left\|(A^TA)^{-1}A^T(Ax_0 + z) - x_0\right\|^2 \\
&= E\left\|(A^TA)^{-1}A^Tz\right\|^2 \\
&= E\left[\text{trace}(z^TA(A^TA)^{-2}A^Tz)\right] \\
&= \text{trace}\left(E[zz^T]A(A^TA)^{-2}A^T\right) \\
&= \text{trace}\left(\sigma^2A(A^TA)^{-2}A^T\right) \\
&= \sigma^2\text{trace}\left((A^TA)^{-1}\right)
\end{aligned}$$

2.4

Again, the code for these experiments in Matlab suing cvx is straightforward:

```
m = 512; n = 128;
sigma = .21; epsilon = 4.4; delta = .84;
A = randn(m,n);
x0 = randn(n,1);
er_ind = randsample(m,round(m*.1));
y = A*x0;
y(er_ind) = -2*y(er_ind);
z_true = sigma*randn(m,1);
y = y + z;
ideal_error = norm(A\z_true);

cvx_begin
    variables x(n) z(m);
    minimize(norm(y-A*x-z,1));
    subject to
        norm(z,2) <= epsilon;
        A'*z == zeros(n,1);
cvx_end

rho_2 = norm(x-x0)/ideal_error

cvx_begin
    variables x(n) z(m);
    minimize(norm(y-A*x-z,1));
    subject to
        norm(z,Inf) <= delta;
        A'*z == zeros(n,1);
cvx_end
```

```
rho_1 = norm(x-x0)/ideal_error
```

The quantity ρ gives the ratio of the error of the algorithm to the optimal estimator if we had $e = 0$. So, while we never expect ρ to be less than 1, if it is close to 1, it means that the performance is almost as good as knowing e exactly.

Unlike the case where the error vector is perfectly sparse, we expect the performance to depend slightly on the size of the gross errors. (Clearly if the nonzero elements of e are much less than σ we expect ρ to be close to 1.) But the performance should be similar for all e with nonzero elements much larger than σ. So with the same corruption procedure as in Problem 1, for the given values of ϵ and δ, we find that $\rho_{P_2} = 1.5 \pm .1$ and $\rho_{P_1} = 2.4 \pm .3$. A factor of about 2 from the optimal error is not unreasonable, so both algorithms reconstruct x_0 about as well as can be hoped for.

Note: The above results might lead one to conclude that P_2 is the superior algorithm in general. However, the performance is a bit sensitive to the threshold values ϵ and δ. (In the worst case, if we accidentally set ϵ or δ to be extremely large, then the values of x and z with $Ax - z = y$ will be feasible and our algorithms will be useless.) However, if we set $\delta = .3$, we find $\rho_{P_1} = 1.4 \pm .1$. So it is unclear without further testing which is all-around better.

Problem 5.14

Since $\tilde{x}$ is a minimizer of the differentiable convex function ϕ, we must have $\nabla\phi(\tilde{x}) = 0$. Or equivalently,

$$\nabla f_0(\tilde{x}) + 2\alpha A^T(A\tilde{x} - b) = 0$$

The Langragian associated with the original problem is

$$L(x, \nu) = f_0(x) + \nu^T(Ax - b)$$

A sufficient condition for some value of ν to be dual feasible is the existence of some finite x which minimizes $\text{Ł}(\cdot, \nu)$. If we define $\boxed{\tilde{\nu} = 2\alpha(A\tilde{x} - b)}$, then we see that $\tilde{x}$ is a minimizer of $\text{Ł}(\cdot, \tilde{\nu})$. This is because the Lagrangian is differentiable convex, and

$$\nabla_x \text{Ł}(\tilde{x}, \tilde{\nu}) = \nabla f_0(\tilde{x}) + A^T\tilde{\nu} = 0$$

Where we use the definition of $\tilde{\nu}$ and $\nabla\phi(\tilde{x}) = 0$, as above. So we conclude that $\tilde{\nu}$ is dual feasible and that $\text{g}(\tilde{\nu}) = \text{L}(\tilde{x},\tilde{\nu})$, where g is the associated Lagrange dual function. This gives a lower bound on the optimal value of the original problem:

$$\boxed{g(\tilde{\nu}) = f_0(\tilde{x}) + 2\alpha\|A\tilde{x} - b\|^2}$$