A cvx Experiment in Compressive Sensing

Consider the following problem

$$\text{minimize } \|x\|_1$$
$$\text{subject to } y = \Phi x$$

where Φ is a $k \times n$ matrix with $k \ll n$, and x is a vector with only S nonzero elements. The interpretation is that y constitutes our k measurements of a sparse vector x, through a "measurement matrix" Φ. Since Φ is fat, there are infinitely many vectors x that satisfy the equality constraint in the problem, so we cannot determine x uniquely, only from the constraint. But if we know that x is sparse (i.e. only S nonzero elements out of n), and if k is large enough (i.e. we have enough `projections' of x), then the optimization problem above can uniquely determine x. Note that we do not need to know the sparsity pattern (which elements of x are nonzero).

For the purposes of this problem, we will assume that Φ consists a matrix with random entries chosen as independent, plus or minus one with equal probability. Recall that k < n, so we have much fewer "samples" (inner products with Bernoulli vectors) than the length of x, hence the name compressed sampling or compressed sensing. The idea is that for almost all selection of k samples, perfect reconstruction of the sparse x is possible with much fewer than n, but more than CS log n samples, where C is just some constant, independent of n and k. Loosely speaking, if k is sufficiently large, the Bernoulli matrix with very high probability yield a sampling matrix Φ that will generate a y which can be used to recover x perfectly using the optimization problem above. This probability will approach 1 very rapidly especially if n is large as well.

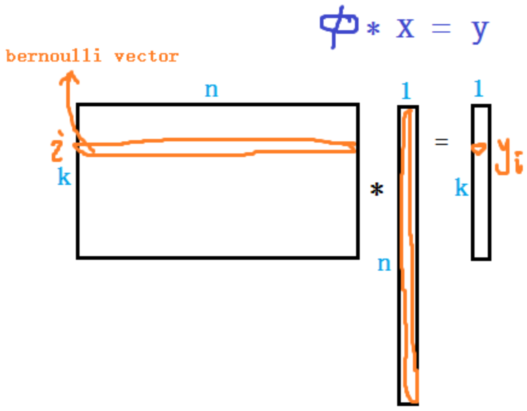
Toward this goal, select n, S, and generate Φ randomly using the description above, Use cvx to recover x. For values of n = 50, n = 100, and n = 500, and values of S (small) that you chose, plot the probability of perfect recovery versus k. To estimate this probability, you need to run this experiment many times and count how many times you get perfect reconstruction. In other words, you need to use a Monte Carlo approach to estimate this probability.

Analysis of the problem:

Original x is a vector of length n. Therefore, in an ideal setting, we takes n measurements (i.e. n many y s) to perfectly recover x.

However, in this case, we can only take k measurements (k < n). Hence the event "perfectly recovering x from y" is not a certain event. It becomes a stochastic event. There is a probability for "recovering x using k many y s". We want to quantify this probability versus k.

x: n*1 vector of original signal (which is sparse)
y: k*1 vector of output values
Φ: k*n measurement matrix



When we get a measurement vector y, we want to guess the original vector x.

Element y_i is obtained by inner production of the i throw of matrix Φ and vector x. The i throw of matrix Φ is a Bernoulli vector (randomly 1 and -1). x is a sparse vector, very few elements of x are non-zero.

Say there are only S=3 elements in x are non-zero. Then all the element wise products on other indexes are definitely zero. For those 3 indexes, there are 2^S=2^3=8 possible combinations of values of i throw of matrix Φ on those three indexes (see chart below). The larger k we have, the more likely we have all these combinations. Those 8 rows are dependent. Rank is 3. i.e. we can obtain all the information if we have at least 3 of them. Therefore, k is positive propositional to S.

Non-zero indexes	7	15	38
value	12	28	9
i throw of matrix Φ	1	1	1
	1	1	-1
	1	-1	1
	1	-1	-1
	-1	1	1
	-1	1	-1
	-1	-1	1
	-1	-1	-1

As matrix Φ is fat, there are multiple x vectors satisfying the constraint $y = \Phi x$. However, as k goes larger ($k > CS \log n$), we pick the one (x vector) with the smallest 1-norm, we are very likely to recover the original x.