525 Computing Project 2, Fall 2011*
DMTF with Compressed Sensing: The Past Meets the Future

Way back in the day when we all had land lines, the phone company figured out who you were calling using a clever scheme called *dual-tone multi-frequency* signaling (DTMF). When pressed, each key would trigger tones corresponding to the row and a key corresponding to the column. This would give a total of 12 different sounds but only 7 different frequencies (remember, there are the "*" and "#" keys). The tones for keys 1-9 are given in Figure 1.

Much more recently, the signal processing community has been enamored with *compressed sensing*, a technique to recover sparse signals from highly incomplete and noisy information. In this project, we'll combine both of these ideas, trying to recover DTMF signals from very few noisy samples. Since each DTMF signal is 2-sparse in a basis of 6 sine waves, let's explore the limits of sensing DTMF signals in noise.

A sine wave is given by

$$s(t) = \sin(2\pi f t)$$

where f is the *frequency* of the sine wave. To make a discrete time approximation, you pick a *sampling frequency* F_s and define a discrete time vector with components

$$t_k = k/F_s$$

the units of this vector are in seconds. Then a discrete sine wave s_k is given by

$$s_k = \sin(2\pi f t_k)$$

*Extra Credit. Due in my office by 5PM, Dec 22, 2011.

	1209	1336	1447
697	**1**	**2**	**3**
770	**4**	**5**	**6**
852	**7**	**8**	**9**

Figure 1: The DTMF pattern, each digit triggers the sum of two sine waves. The frequencies are the corresponding column and row headers.

To make a DTMF tone, we have 6 frequencies:

$$\begin{aligned} f_1^{lo} &= 697, & f_2^{lo} &= 770, & f_3^{lo} &= 852 \\ f_1^{hi} &= 1209, & f_2^{hi} &= 1336, & f_3^{hi} &= 1447 \end{aligned}$$

Each key of the DTMF signal table produces the signal

$$\begin{aligned} s_k^{(1)} &= \tfrac{1}{2}\sin\left(2\pi f_1^{(lo)} t_k\right) + \tfrac{1}{2}\sin\left(2\pi f_1^{(hi)} t_k\right) \\ s_k^{(2)} &= \tfrac{1}{2}\sin\left(2\pi f_1^{(lo)} t_k\right) + \tfrac{1}{2}\sin\left(2\pi f_2^{(hi)} t_k\right) \\ s_k^{(3)} &= \tfrac{1}{2}\sin\left(2\pi f_1^{(lo)} t_k\right) + \tfrac{1}{2}\sin\left(2\pi f_3^{(hi)} t_k\right) \\ s_k^{(4)} &= \tfrac{1}{2}\sin\left(2\pi f_2^{(lo)} t_k\right) + \tfrac{1}{2}\sin\left(2\pi f_1^{(hi)} t_k\right) \\ s_k^{(5)} &= \tfrac{1}{2}\sin\left(2\pi f_2^{(lo)} t_k\right) + \tfrac{1}{2}\sin\left(2\pi f_2^{(hi)} t_k\right) \\ s_k^{(6)} &= \tfrac{1}{2}\sin\left(2\pi f_2^{(lo)} t_k\right) + \tfrac{1}{2}\sin\left(2\pi f_3^{(hi)} t_k\right) \\ s_k^{(7)} &= \tfrac{1}{2}\sin\left(2\pi f_3^{(lo)} t_k\right) + \tfrac{1}{2}\sin\left(2\pi f_1^{(hi)} t_k\right) \\ s_k^{(8)} &= \tfrac{1}{2}\sin\left(2\pi f_3^{(lo)} t_k\right) + \tfrac{1}{2}\sin\left(2\pi f_2^{(hi)} t_k\right) \\ s_k^{(9)} &= \tfrac{1}{2}\sin\left(2\pi f_3^{(lo)} t_k\right) + \tfrac{1}{2}\sin\left(2\pi f_3^{(hi)} t_k\right) \end{aligned}$$

To phrase this as a matrix, define a $T \times 6$ matrix with rows

$$S_{kj} = [\sin(2\pi f_j t_k)]$$

where

$$f = \left[\begin{array}{cccccc} f_1^{(lo)} & f_2^{(lo)} & f_3^{(lo)} & f_1^{(hi)} & f_2^{(hi)} & f_3^{(hi)} \end{array}\right]$$

Then we can rewrite the DTMF signals for all of the digits as

$$\begin{aligned}
s_k^{(1)} &= S_{k\cdot}\left(\tfrac{1}{2}e_1 + \tfrac{1}{2}e_4\right)\\
s_k^{(2)} &= S_{k\cdot}\left(\tfrac{1}{2}e_1 + \tfrac{1}{2}e_5\right)\\
s_k^{(3)} &= S_{k\cdot}\left(\tfrac{1}{2}e_1 + \tfrac{1}{2}e_6\right)\\
s_k^{(4)} &= S_{k\cdot}\left(\tfrac{1}{2}e_2 + \tfrac{1}{2}e_4\right)\\
s_k^{(5)} &= S_{k\cdot}\left(\tfrac{1}{2}e_2 + \tfrac{1}{2}e_5\right)\\
s_k^{(6)} &= S_{k\cdot}\left(\tfrac{1}{2}e_2 + \tfrac{1}{2}e_6\right)\\
s_k^{(7)} &= S_{k\cdot}\left(\tfrac{1}{2}e_3 + \tfrac{1}{2}e_4\right)\\
s_k^{(8)} &= S_{k\cdot}\left(\tfrac{1}{2}e_3 + \tfrac{1}{2}e_5\right)\\
s_k^{(9)} &= S_{k\cdot}\left(\tfrac{1}{2}e_3 + \tfrac{1}{2}e_6\right)
\end{aligned}$$

where e_i are the standard basis vectors

Thus, we can model a DTMF signal as a length T vector, y, given by the expression

$$y = Sx + n\,.$$

Here S is the signalling matrix, and x is a vector with 2 components equal to 1/2 and all other components equal to zero. n is a vector of noise.

We are going to compare the performance of two different regularization problems:

$$\text{minimize}_x \|y - Sx\|_2^2 + \mu\|x\|_1 \tag{1}$$

versus

$$\text{minimize}_x \|y - Sx\|_2^2 + \lambda\|x\|_2^2 \tag{2}$$

In all of the problems, use $F_S = 22050$.

1. Generate signals y for each digit with $T = 2000$. Plot your signals over time. (for fun, try the command `sound(y,Fs)`). Now generate and plot the same signals but add a noise vector `n = sigma*randn(T,1)` with $\sigma = 10$.

2. We consider a signal "decoded" if the two largest components of x correspond to the true DTMF components. For example, if the true signal is an 8, the correct DTMF components are 3 and 5.

For each digit, run ℓ_1 minimization (Problem 1) with the following parameter settings:

$$T = 10, 20, 40, 80, 160$$
$$\sigma = 0, 0.1, 1, 10, 100$$

For each instance, determine a value of the parameter μ that gives you the most correct decodings. In all of the cases, report the average number of correct decodings for your best value of μ (over ten instances of the random noise). Is there a difference between the digits?

3. Repeat Problem 1, but now use ℓ_2 decoding (Problem 2). Are there any regimes where ℓ_2 performs better than ℓ_1?

4. Repeat Problem 1, but this time us the simpler decoding scheme $x = S^T y$. Note that in this case, you do not have to choose a regularization parameter. How often does this scheme succeed for the given values of T and σ?

This project was inspired by the Matlab article "'Magic' Reconstruction: Compressed Sensing" by Cleve Moler, `http://www.mathworks.com/company/newsletters/articles/clevescorner-compressed-sensing.html`.