Convex Optimization

Linear Programming Applications

Today

- This is the "What's the point?" lecture. . .
- What can be solved using linear programs?

Just an introduction. . .

Linear Programs

A linear program is written

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & x \geq 0, \end{array}$$

in the variable $x \in \mathbf{R}^n$. Or in inequality form

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax \leq b. \end{array}$$

Linear Program Applications

Linear Programming: applications

- Originally, linear programs considered "toy problems"
- Algorithm came first
- LPs could be solved efficiently, some applications were found
- Successful applications meant publicity
- Tons of applications subsequently discovered. . .
- Among the most commonly used optimization results today

Linear Programming: applications

Today, a quick look at applications of linear programming:

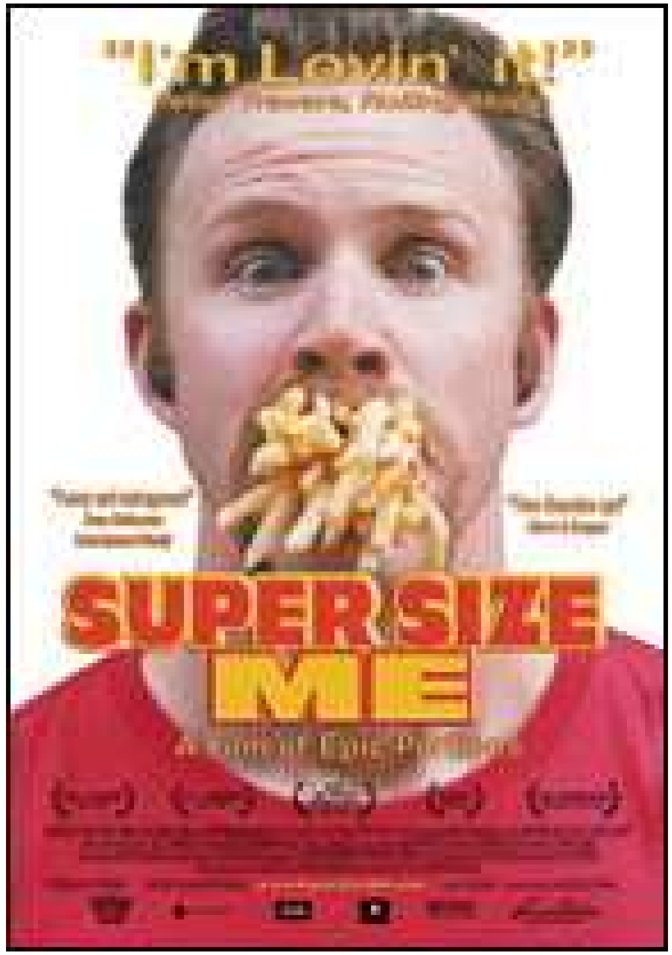
- Finance
- Statistics
- Networks
- Game theory
- Structural design
- Scheduling
- Signal processing, etc

A bit of history: the diet problem

The diet problem:

- Resource allocation problem
- Could replace, calories & nutrients by parts in a factory, etc
- Classic first example in linear programming classes
- Follow a 50 years old tradition. . .

The diet problem



Eating fast food optimally, using linear programming. . .

A bit of history: the diet problem

The diet problem:

- We're given the nutrition facts on burgers, fries, etc
- We need to design our meal so that the quantity of nutrients falls between certain values
- Objective: minimize costs
- Another possibility: minimize calories (optimally healthy fast-food meal)

Easy: this is a linear program. . .

A bit of history: the diet problem

Data (fictitious). On prices:

Quarter Pounder w/ Cheese:	1.84
McLean Deluxe w/ Cheese:	2.19
Big Mac:	1.84
Filet-O-Fish:	1.44
McGrilled Chicken:	2.29
Fries, small:	0.77
Sausage McMuffin:	1.29
1% Lowfat Milk:	0.60
Orange Juice:	0.72

A bit of history: the diet problem

Minimum and maximum values for some type of nutrients:

	Min.	Max.
Calories	2000	
Carbs	350	375
Protein	55	
VitA	100	
VitC	100	
Calc	100	
Iron	100	

A bit of history: the diet problem

Nutrition facts:

	Cal	Carbs	Protein	VitA	VitC	Calc	Iron
Quarter Pounder	510	34	28	15	6	30	20
McLean Deluxe	370	35	24	15	10	20	20
Big Mac	500	42	25	6	2	25	20
Filet-O-Fish	370	38	14	2	0	15	10
McGrilled Chicken	400	42	31	8	15	15	8
Fries, small	220	26	3	0	15	0	2
Sausage McMuffin	345	27	15	4	0	20	15
1% Lowfat Milk	110	12	9	10	4	30	0
Orange Juice	80	20	1	2	120	2	2

A bit of history: the diet problem

We can write this as a linear program:

- The variables are x_i, the quantity of item in the menu we purchase
- We let c_i be the cost of each item, the total cost of the meal is:

$$\sum_{i=1}^{9} c_i x_i$$

- Let A_{ij} be the nutrition value for nutrient i in item j, the nutrition constraints are:

$$min_i \leq \sum_{j=1}^{9} A_{ij} x_j \leq max_i \quad \text{for each nutrient } i$$

- And of course: all the quantities x_i have to be positive: $x_i \geq 0$

A bit of history: the diet problem

Minimum cost meal meeting minimum requirements

$$
\begin{array}{lll}
\text{minimize} & \sum_{i=1}^{9} c_i x_i & \\
\text{subject to} & min_i \leq \sum_{j=1}^{9} A_{ij} x_j \leq max_i & \text{for each nutrient } i \\
& x_i \geq 0, &
\end{array}
$$

Solution:

Quarter Pounder w/ Cheese	4.38525
Fries, small	6.14754
1% Lowfat Milk	3.42213

Price: 14.85, but 4000 calories. . .

A bit of history: the diet problem

2500 calories meal meeting minimum requirements

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{9} c_i x_i \\ \text{subject to} & \sum_{j=1}^{9} A_{1j} x_j = 2500 \\ & min_i \leq \sum_{j=1}^{9} A_{ij} x_j \leq max_i \quad \text{for each nutrient } i = 2, \ldots, 7 \\ & x_i \geq 0, \end{array}$$

Solution:

Quarter Pounder w/ Cheese	0.231942
McLean Deluxe w/ Cheese	3.85465
1% Lowfat Milk	2.0433
Orange Juice	9.13408

Price goes up: \$16.67. . .

A bit of history: the diet problem

Can we make a **2000 calories** meal meeting minimum requirements?

$$
\begin{array}{ll}
\text{minimize} & \sum_{i=1}^{9} c_i x_i \\
\text{subject to} & \sum_{j=1}^{9} A_{1j} x_j = 2000 \\
& min_i \leq \sum_{j=1}^{9} A_{ij} x_j \leq max_i \quad \text{for each nutrient } i = 2, \ldots, 7 \\
& x_i \geq 0,
\end{array}
$$

- No solution!
- What's the best we can do?

A bit of history: the diet problem

Minimum calories meal meeting minimum requirements

minimize $\sum_{j=1}^{9} A_{1j} x_j$

subject to $min_i \leq \sum_{j=1}^{9} A_{ij} x_j \leq max_i$ for each nutrient $i = 2, \ldots, 7$
$x_i \geq 0,$

Solution:

McLean Deluxe w/ Cheese	4.08805
1% Lowfat Milk	2.04403
Orange Juice	9.1195

Price is $16.75, minimum calories: 2467

A bit of history: the diet problem

A few interesting results from this experiment:

- Some problems are **infeasible**, how do we detect that?
- We can't ask for integer results
- Solution: **rounding**
- But we can't be certain to get the optimal integer solution. . .
  (more on this later)

Finance

Portfolio theory.

- Classic view: mean-variance tradeoff
- Portfolio management: a quadratic program (later)
- Variance is (by far) not the only measure of risk
- Other possibility: **mean absolute deviation**:

$$risk = \frac{1}{T} \sum_{t=1}^{T} |r_t - \bar{r}|$$

where $r_t = S_t - S_{t-1}$ is the return at time t and $\bar{r}$ the mean return.

Portfolio Optimization

Year	US 3-Month T-Bills	US Gov. Long Bonds	S&P 500	Wilshire 5000	NASDAQ Composite	Lehman Bros. Corp. Bonds	EAFE	Gold
1973	1.075	0.942	0.852	0.815	0.698	1.023	0.851	1.677
1974	1.084	1.020	0.735	0.716	0.662	1.002	0.768	1.722
1975	1.061	1.056	1.371	1.385	1.318	1.123	1.354	0.760
1976	1.052	1.175	1.236	1.266	1.280	1.156	1.025	0.960
1977	1.055	1.002	0.926	0.974	1.093	1.030	1.181	1.200
1978	1.077	0.982	1.064	1.093	1.146	1.012	1.326	1.295
1979	1.109	0.978	1.184	1.256	1.307	1.023	1.048	2.212
1980	1.127	0.947	1.323	1.337	1.367	1.031	1.226	1.296
1981	1.156	1.003	0.949	0.963	0.990	1.073	0.977	0.688
1982	1.117	1.465	1.215	1.187	1.213	1.311	0.981	1.084
1983	1.092	0.985	1.224	1.235	1.217	1.080	1.237	0.872
1984	1.103	1.159	1.061	1.030	0.903	1.150	1.074	0.825
1985	1.080	1.366	1.316	1.326	1.333	1.213	1.562	1.006
1986	1.063	1.309	1.186	1.161	1.086	1.156	1.694	1.216
1987	1.061	0.925	1.052	1.023	0.959	1.023	1.246	1.244
1988	1.071	1.086	1.165	1.179	1.165	1.076	1.283	0.861
1989	1.087	1.212	1.316	1.292	1.204	1.142	1.105	0.977
1990	1.080	1.054	0.968	0.938	0.830	1.083	0.766	0.922
1991	1.057	1.193	1.304	1.342	1.594	1.161	1.121	0.958
1992	1.036	1.079	1.076	1.090	1.174	1.076	0.878	0.926
1993	1.031	1.217	1.100	1.113	1.162	1.110	1.326	1.146
1994	1.045	0.889	1.012	0.999	0.968	0.965	1.078	0.990

Historical (relative) returns S_t/S_{t-1} on a few investments. . .
(EAFE: Europe, Australia, and Far East).

Portfolio Optimization

Markovitz type model:

- We look for a portfolio of N assets with coefficients x_i
- We have an initial budget of \$1
- For a given level of **risk**, we seek to maximize **return**
- When the level of risk (μ) varies, the maximum return defines a set of optimal risk/return tradeoffs: the **efficient frontier**
- We consider aboslute returns $r_t = S_t - S_{t-1}$.

Portfolio Optimization

The program to be solved can be written:

$$\begin{aligned}
&\text{maximize} && \frac{1}{T}\sum_{i=1}^{N}\sum_{t=1}^{T} x_i r_{i,t} && \text{(portfolio return)}\\
&\text{subject to} && \frac{1}{T}\sum_{t=1}^{T}\left|\sum_{i=1}^{N} x_i r_{i,t} - \sum_{i=1}^{N} x_i \bar{r}_i\right| \leq \mu && \text{(portfolio risk bounded)}\\
& && \sum_{i=1}^{N} x_i P_i = 1 && \text{(initial budget)}\\
& && x_i \geq 0 && \text{(no short sale)}
\end{aligned}$$

Is this a **linear program**?

Portfolio Optimization

- The following constraint on the absolute value:

$$|x| \leq y$$

  is equivalent to:

$$-y \leq x \leq y$$

- This means that we can replace each inequality on an absolute value by to inequalities
- We have to introduce additional variables in the original program. . .

Portfolio Optimization

The new program is written:

$$\text{maximize} \quad \frac{1}{T}\sum_{i=1}^{N}\sum_{t=1}^{T} x_i r_{i,t} \qquad \text{(portfolio return)}$$

$$\text{subject to} \quad \frac{1}{T}\sum_{t=1}^{T} y_t \leq \mu \qquad \text{(portfolio risk bounded)}$$

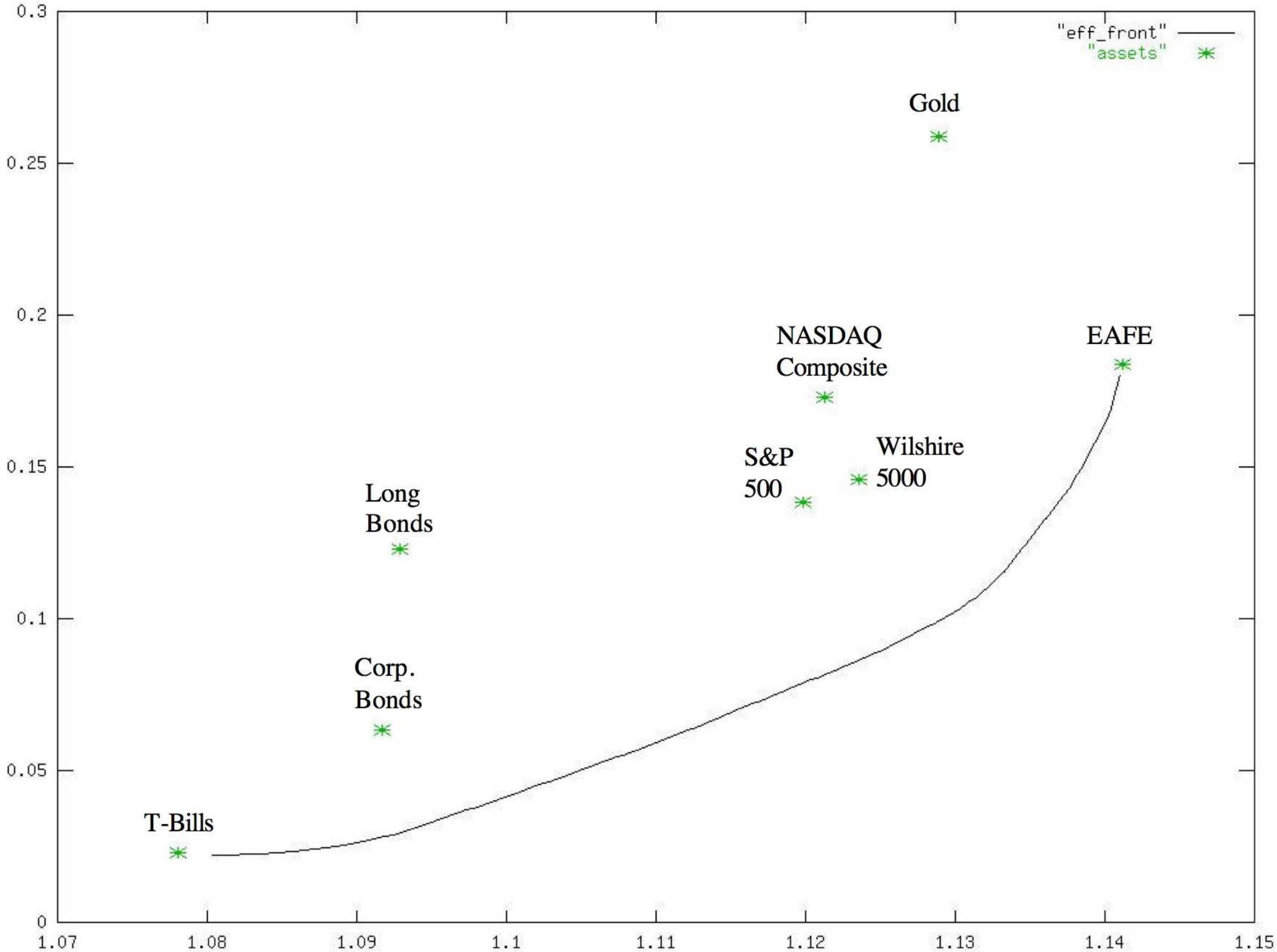
$$-y_t \leq \left(\sum_{i=1}^{N} x_i r_{i,t} - \sum_{i=1}^{N} x_i \bar{r}_i\right) \leq y_t$$

$$\sum_{i=1}^{N} x_i P_i = 1 \qquad \text{(initial budget)}$$

$$x_i \geq 0 \qquad \text{(no short sale)}$$

This is now a **linear program**!

Portfolio Optimization



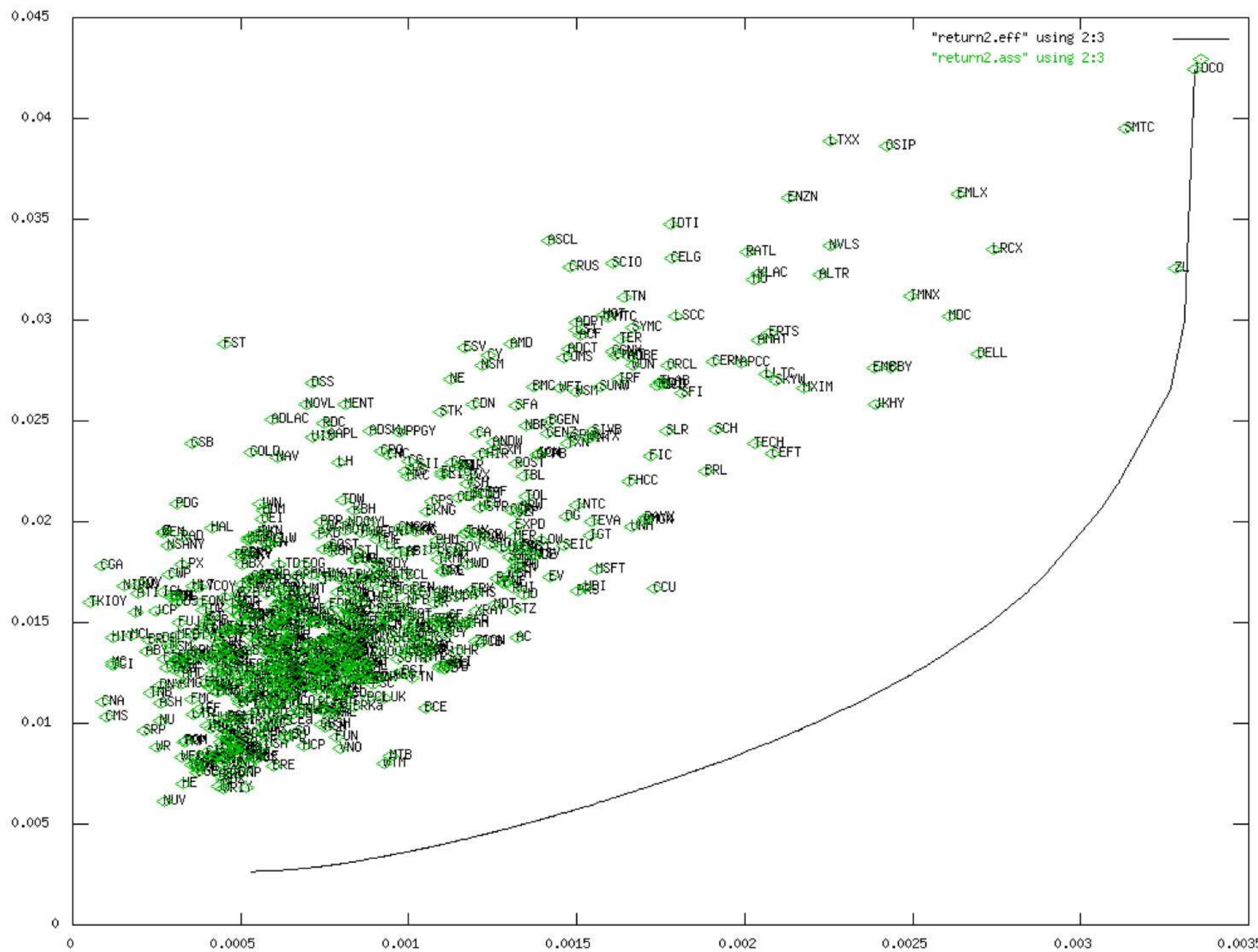
Efficient frontier for a few reference assets ($N = 8$).

Portfolio Optimization

μ	US 3-Month T-Bills	Lehman Bros. Corp. Bonds	NASDAQ Comp.	Wilshire 5000	Gold	EAFE	Reward	Risk
0.1800					0.017	0.983	1.141	0.180
0.1538					0.191	0.809	1.139	0.154
0.1275				0.119	0.321	0.560	1.135	0.128
0.1013				0.407	0.355	0.238	1.130	0.101
0.0751			0.340	0.180	0.260	0.220	1.118	0.075
0.0488	0.172	0.492			0.144	0.008	1.104	0.049
0.0226	0.815	0.100	0.037		0.041	0.008	1.084	0.022

Composition, risk and return of **optimal portfolios** for various values of μ.

Portfolio Optimization



Efficient frontier for 719 stocks.

Statistics: Regression

How far is this from the standard mean variance analysis?

- We replace the variance by the deviation
- How do these two measures of "error" compare?

Let's pick an example from statistics:

- Regress a set of data points on a few variables
- Compare least squares regression with least absolute deviation regression

Statistics: Regression

Given N data points y_i and x_i, we look for parameters a and b and compute the "best" linear model $y = ax + b$

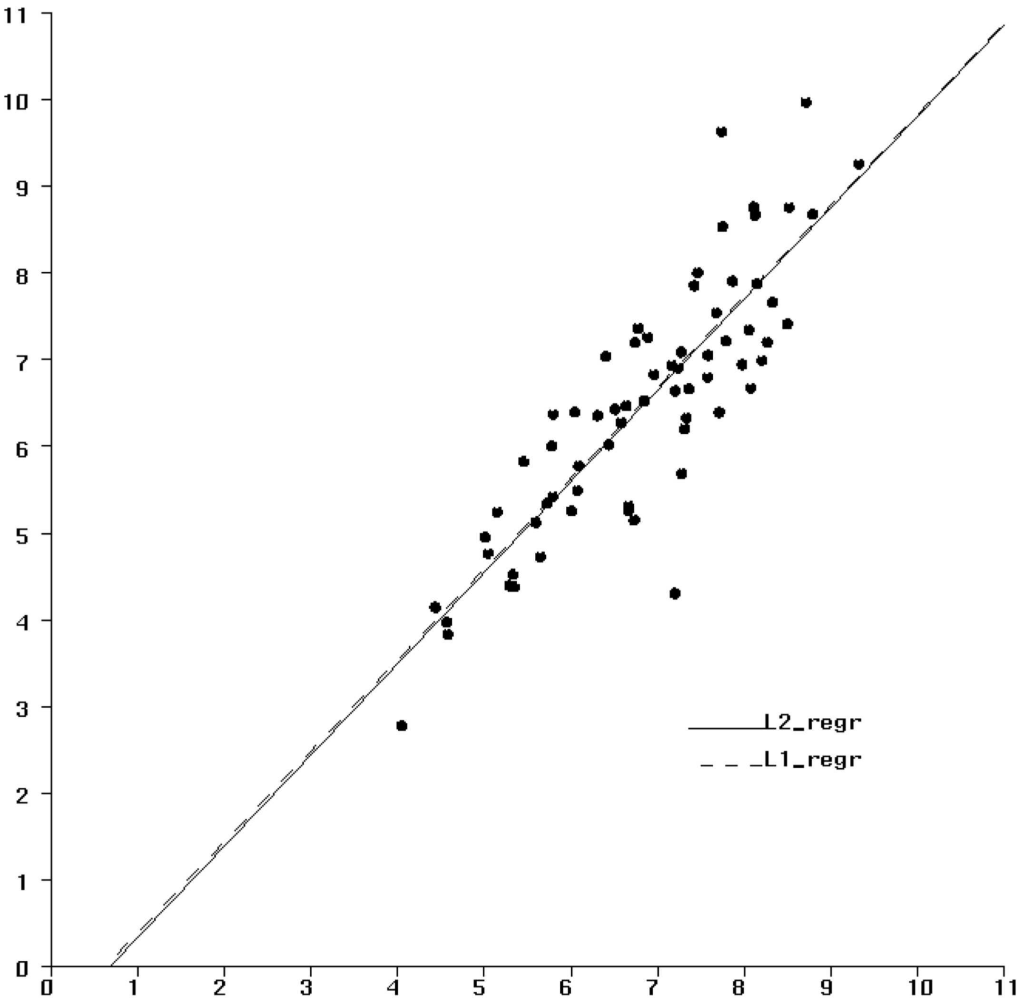
- The usual least squares regression is written:

$$\text{minimize} \sum_{i=1}^{N} \|y_i - ax_i - b\|^2$$

- The least absolute deviation regression is here:

$$\text{minimize} \sum_{i=1}^{N} |y_i - ax_i - b|$$

Portfolio Optimization



Not that different here. . .

Two person game.

- Count to three and declare:

  Paper Scissors Rock

- Winner selected according to:

  Rock beats Scissors
  Paper beats Rock
  Scissors beats Paper

- We can arrange this in a **payoff matrix**:

$$\begin{array}{c} \\ P \\ S \\ R \end{array} \quad \begin{array}{c} \begin{array}{ccc} P & S & R \end{array} \\ \begin{bmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{bmatrix} \end{array}$$

Game Theory

- Playing a fixed (deterministic, pure) strategy is bad: "always stone" is always beaten by paper. . .
- We know from game theory that there is always a Nash equilibrium involving random (mixed) strategies.
- How do we find these?
- A random strategy is simply a probability vector:

$$\sum_{i=1}^{3} x_i = 1 \text{ and } x_i \geq 0$$

- Solving for the equilibrium strategy for both players is a **linear program** (more details later).

Signal Processing

FIR filter design.

- **Finite Impulse Response** filter:

$$y_t = \sum_{\tau=0}^{n-1} h_\tau u_{t-\tau}$$

  where u_t is the input signal and h_i are the filter coefficients

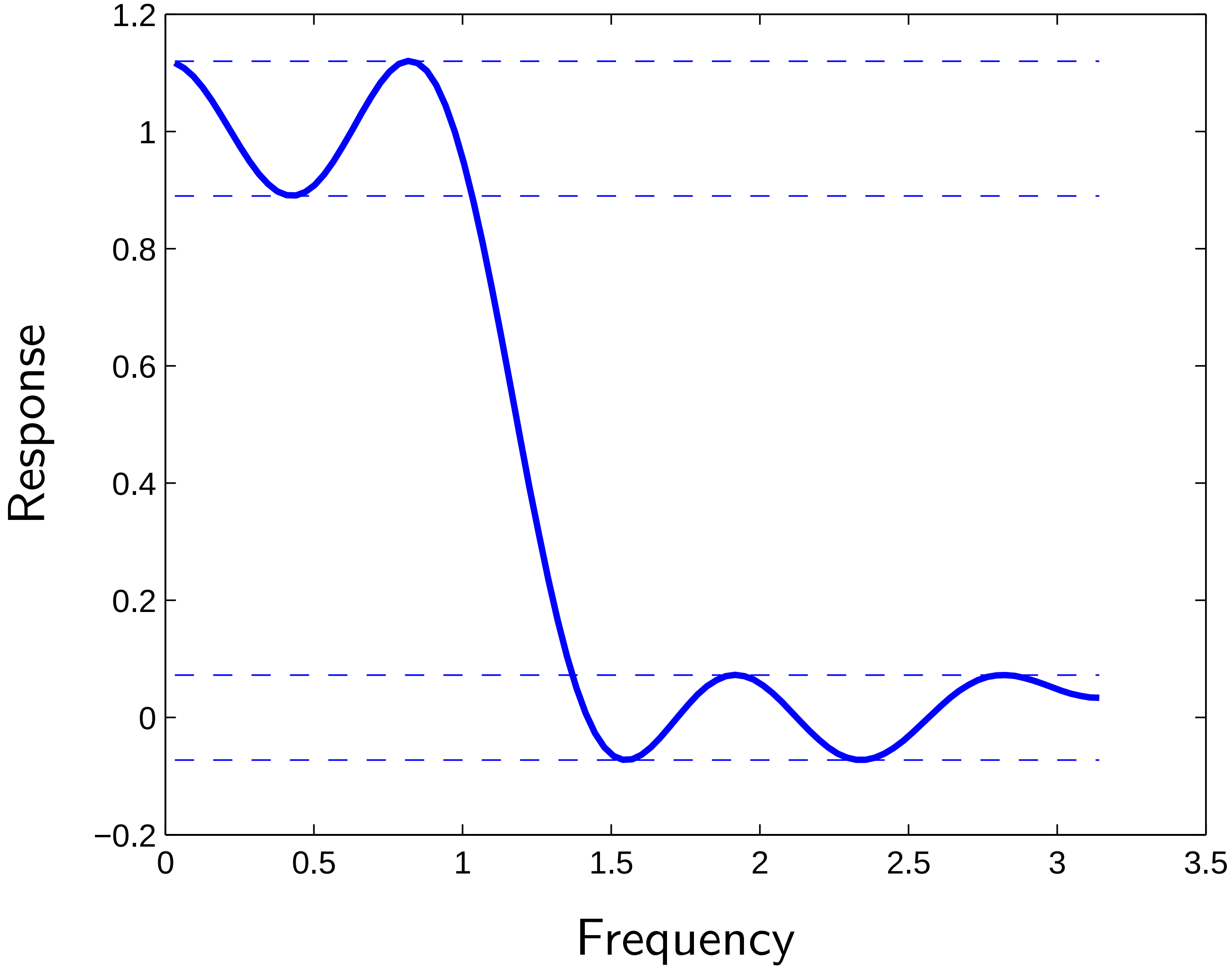
- The magnitude of the **frequency response** of the filter can be written:

$$|\tilde{H}(\omega)| = 2h_0 \cos(N\omega) + 2h_1 \cos((N-1)\omega) + \ldots + h_N$$

- For each particular frequency ω, this a **linear** function of the filter coefficients h

Designing a custom filter is just a **linear program** . . .

Signal Processing



This filter lets **bass** go through and filters out higher frequencies (low-pass)

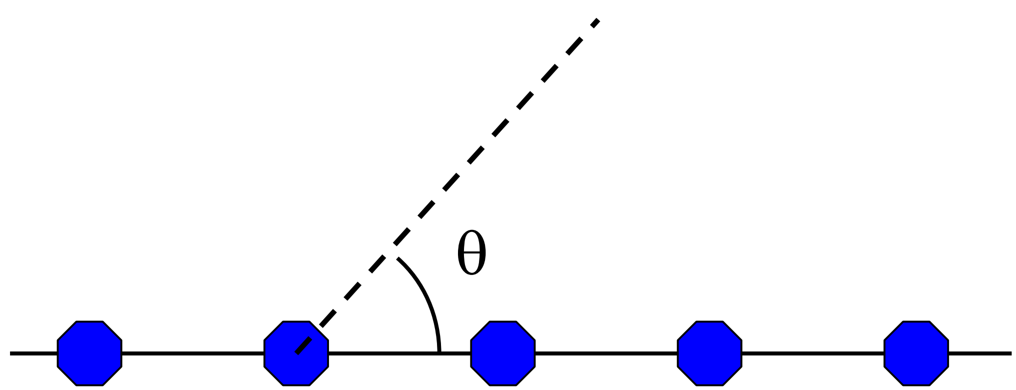
Wifi

- Wifi (802.11) is another example. . .
- Maximum allowed radiated power (EIRP) is 100mW
- Why? So you don't fry your friend next door, also avoids interferences. . .
- This power is dissipated in *all* directions. . .
- Increase the range: focus most of this power in one direction

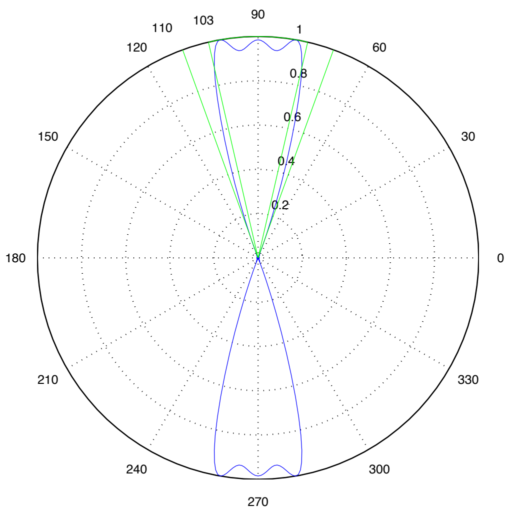
Wifi

Professional solution.

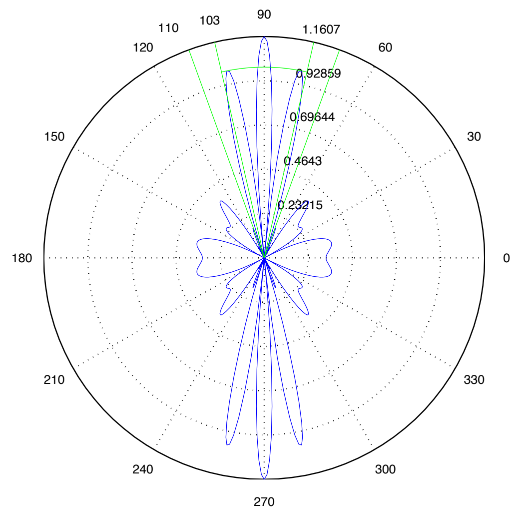
- Use multiple antennas
- Use interference patterns to focus most of the power in a particular direction
- Problem is similar to filter design: **linear program**



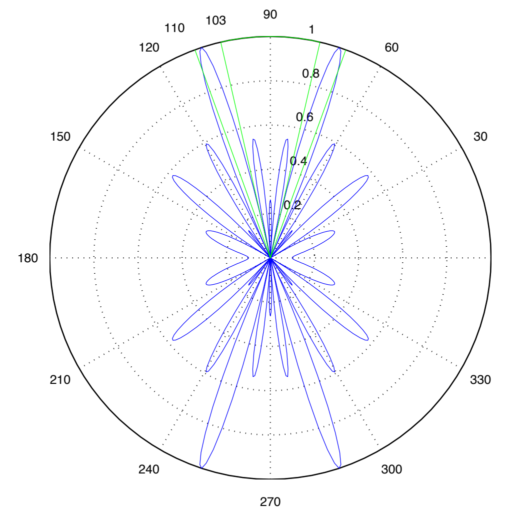
Wifi



Dream
no errors
$\|D_* - D\|_2 = 0.014$



Reality
0.1% errors
$\|D_* - D\|_2 \in [0.17, 0.89]$

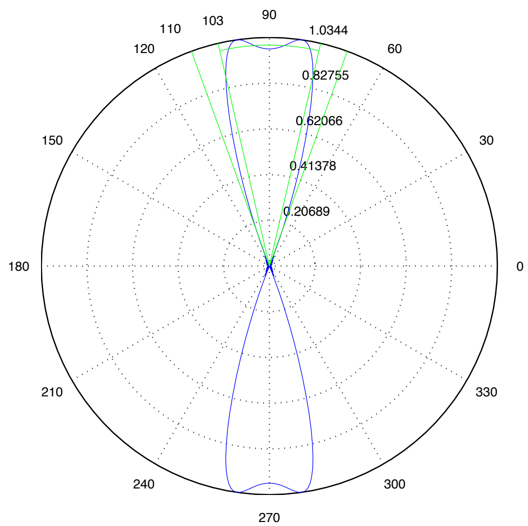


Reality
2% errors
$\|D_* - D\|_2 \in [2.9, 19.6]$

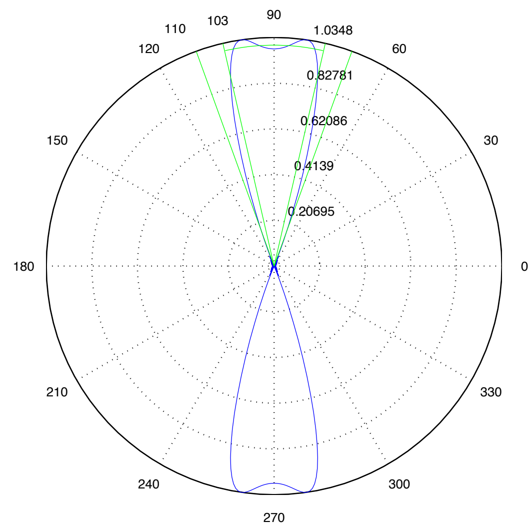
Nominal Least Squares design: dream and reality
Data over a 100-diagram sample

Implementation is tricky. . .

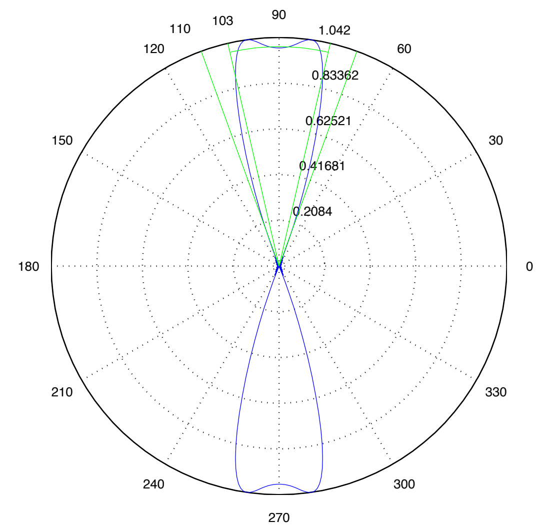
Wifi



Dream
no errors
$\|D_* - D\|_2 = 0.025$



Reality
0.1% errors
$\|D_* - D\|_2 \approx 0.025$



Reality
2% errors
$\|D_* - D\|_2 \approx 0.025$

Robust Least Squares design: dream and reality
Data over a 100-diagram sample

Convex interpolation

What's next? We will study **convex problems.**

- Much more general class of problems
- Complexity similar to linear programming
- Similar solvers
- Very very very long list of applications in statistics, engineering, finance, etc.