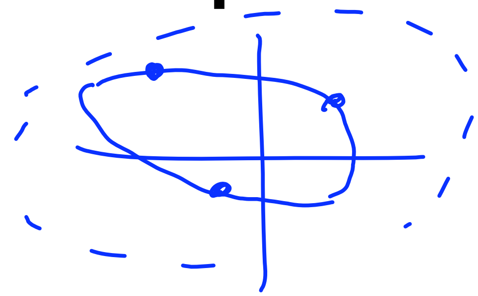
Ex: minimum volume ellipsoid



- Given points $x_1, x_2, \ldots, x_k$
- $\min_{A, x_C} \; vol(A)$ s.t.

  $(x_i - x_C)^T A (x_i - x_C) \le 1 \quad i = 1, \ldots, k$

  $A \in S^{n*n}$

  $A \succcurlyeq 0$

$vol(A) \propto \frac{1}{\sqrt{|A|}}$

$\ln vol(A) = -\frac{1}{2} \ln |A| + \text{const}$

$\min -\ln |A|$ s.t

{constraints}

Schur complement

- Symmetric block matrix $M = \begin{pmatrix} A & B \\ B^T & C \end{pmatrix}$
- Schur complement is $S = C - B^T A^{-1} B$
- $M \succ 0$ iff $A \succ 0$ and $S \succ 0$

Back to min-volume ellipsoid

- $\max_{A, x_C} \ \log |A| \ \text{ s.t.}$

  $(x_i - x_C)^T A (x_i - x_C) \leq 1 \quad i = 1, \ldots, k$

  $A = A^T, A \succcurlyeq 0$

- $\max_{A, B, u, z} \ \log |A| \ \text{ s.t.}$

Ex: manifold learning

- Given points $x_1, \ldots, x_m$ $\in \mathbb{R}^n \leftarrow$ big
- Find points $y_1, \ldots, y_m$ $\in \mathbb{R}^d \leftarrow$ $d \ll n$
- Preserving **local geometry**
  - neighbor edges N $(i,j) \in N \Longleftrightarrow$ preserve geometry btwn x_i, x_j
  - distances $\|x_i - x_j\| = \|y_i - y_j\| \quad (i,j) \in N$
- If we preserve distances among x_i, x_j, x_l
  we also preserve angles $(x_i - x_j) \cdot (x_i - x_l)$

Step 1: “embed” R^n into R^n

- While preserving local distances, move points to make manifold as flat as possible
- max

  s.t.

Step 2: reduce to R^d

- Now that manifold is flat, just use PCA:

Maximizing variance

- max s.t.

- max s.t.

Summary

- Solve SDP to “embed” R^n into R^n
- Use PCA to embed R^n into R^d
- Called “semidefinite embedding” or “maximum variance unfolding”

- Problems?

Problem: solving SDP

- Kernel matrix K
- Idea: suppose we know a subspace that preserves geometry

Side note: non-Euclidean

- If original distances are not Euclidean, might not be able to duplicate them exactly in Euclidean R^n
- Would need to soften constraints: **approximately** preserve local distances

Ex: convex games

- “Robin hood” game:
  - shoot two arrows at target simultaneously
  - Robin Hood’s cost:
  - Sheriff of Nottingham’s reward:

Convex games: definition

- Players p_1, p_2, …, p_k
- p_i chooses x_i from convex set X_i

  all simultaneously
- Cost to p_i is $f_i(x_1, x_2, \ldots, x_k)$
- Cost f_i is convex in x_i for fixed x_{-i}
- Zero-sum:

  2 players, $f_1(x_1, x_2) = -f_2(x_1, x_2)$

Equilibrium

- Minimax equilibrium: find
  - value v
  - distribution P(x)
- such that
  -
  -

Solving convex programs

- Linear programs:
- General CP:
- Interesting special cases: QP, SOCP, SDP

Separation oracle: QPs

Separation oracle: SOCPs

- SOC constraint: $||Ax + b|| \leq c'x + d$
- Given x_0 that fails:

Separation oracle: SDPs

- SDP constraint:

  $A = x_1 A_1 + x_2 A_2 + \ldots$

  $A \in S_+$

Multi-criterion optimization

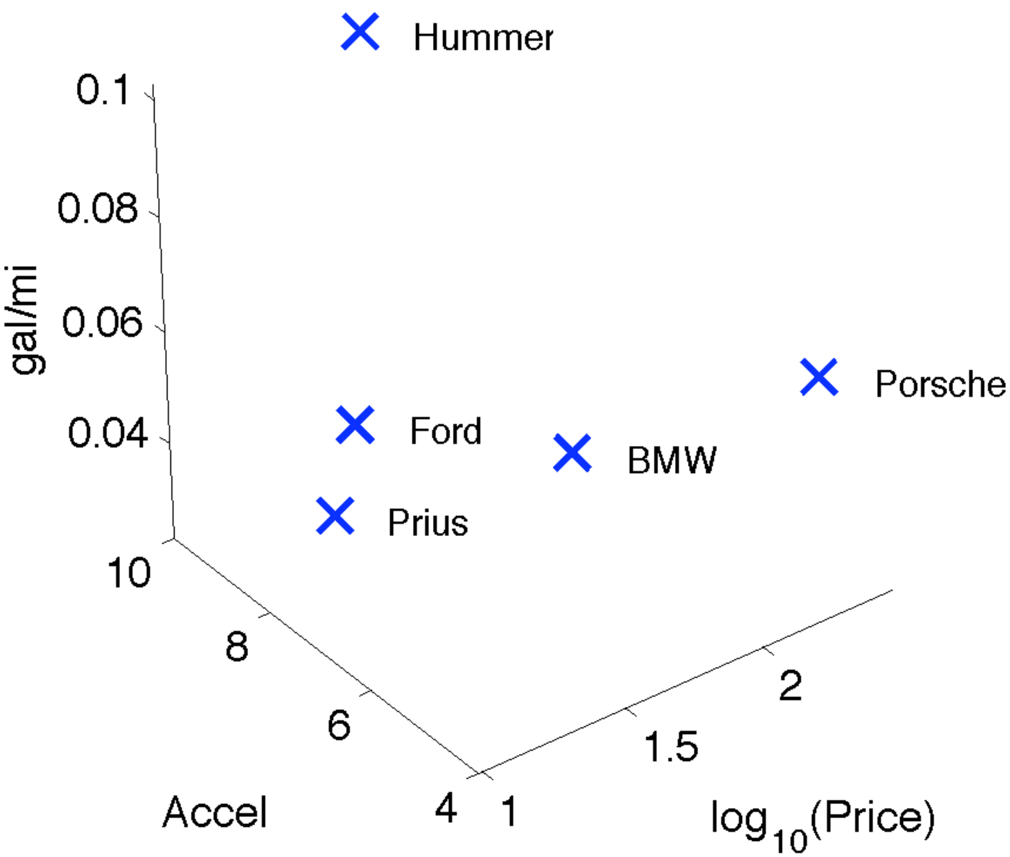
- Ordinary feasible region

- Indecisive optimizer: wants all of

Buying the perfect car

$K	0-60	MPG

Pareto optimality



x* Pareto optimal =

Pareto examples

Scalarization

- To find Pareto optima of convex problem: