# 15-780 – Numerical Optimization

J. Zico Kolter

January 29, 2014

# Overview

- Introduction to mathematical programming problems
- Applications
- Classification of optimization problems
- (Linear algebra review)
- Convex optimization problems
- Nonconvex optimization problems
- Solving optimization problems

# Overview

- Introduction to mathematical programming problems
- Applications
- Classification of optimization problems
- (Linear algebra review)
- Convex optimization problems
- Nonconvex optimization problems
- Solving optimization problems

# Introduction to mathematical optimization

- Casting AI problems as optimization / mathematical programming problems has been one of the primary trends of the last 15 years

- A topic *not* highlighted in textbook (see website for additional readings)

- A seemingly remarkable fact:

| | **Search problems** | **Mathematical programs** |
|---|---|---|
| **Variable type** | Discrete | Continuous |
| **# of possible solutions** | Finite | Infinite |
| **"Difficulty" of solving** | Exponential | Polynomial (often) |

# Formal definition

- Mathematical programs are problems of the form

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \leq 0 \quad i = 1, \ldots, m \end{aligned}$$

  - $x \in \mathbb{R}^n$ is the *optimization variable*
  - $f : \mathbb{R}^n \to \mathbb{R}$ is *objective function*
  - $g_i : \mathbb{R}^n \to \mathbb{R}$ are (inequality) *constraint functions*

- *Feasible region*: $\mathcal{C} = \{x : g_i(x) \leq 0, \forall i = 1, \ldots, m\}$

- $x^\star \in \mathbb{R}^n$ is an *optimal solution* if $x^\star \in \mathcal{C}$, and $f(x^\star) \leq f(x), \;\; \forall x \in \mathcal{C}$

# Note on naming

- *Mathematical program* (or *mathematical optimization problem*, and sometimes just *optimization problem*) refer to the problem of the form

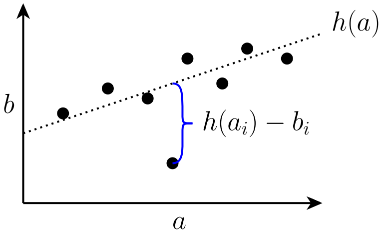
$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \leq 0 \;\; i = 1, \ldots, m \end{aligned}$$

- *Numerical optimization* (or *mathematical optimization*, or just *optimization*) refers to the general study of these problems, as well as methods for solving them on a computer

# Overview

- Introduction to mathematical programming problems
- Applications
- Classification of optimization problems
- (Linear algebra review)
- Convex optimization problems
- Nonconvex optimization problems
- Solving optimization problems

# Least-squares fitting

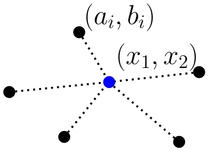



- Given $a_i, b_i, i = 1, \ldots, m$, find $h(a) = x_1 a + x_2$ that optimizes

$$\underset{x}{\text{minimize}} \quad \sum_{i=1}^{m} (x_1 a_i + x_2 - b_i)^2$$

($x_1$ is slope, $x_2$ is intercept)

# Weber point




- Given $m$ points in 2D space $(a_i, b_i),\ i = 1, \ldots m$, find the point that minimizes the sum of the Euclidean distances

$$\underset{x}{\text{minimize}} \quad \sum_{i=1}^{m} \sqrt{(x_1 - a_i)^2 + (x_2 - b_i)^2}$$

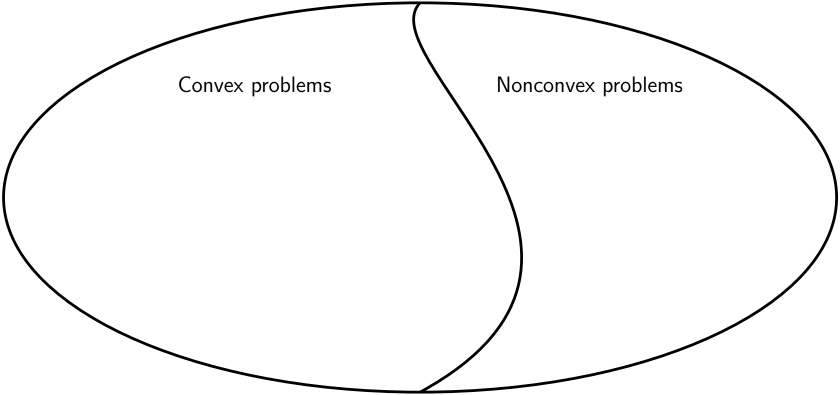
- Many modifications, e.g. keep $x$ within range $(a_l, b_l)$, $(a_u, b_u)$

$$\begin{array}{ll} \underset{x}{\text{minimize}} & \sum_{i=1}^{m} \sqrt{(x_1 - a_i)^2 + (x_2 - b_i)^2} \\ \text{subject to} & a_l \le x_1 \le a_u,\ b_l \le x_2 \le b_u \end{array}$$

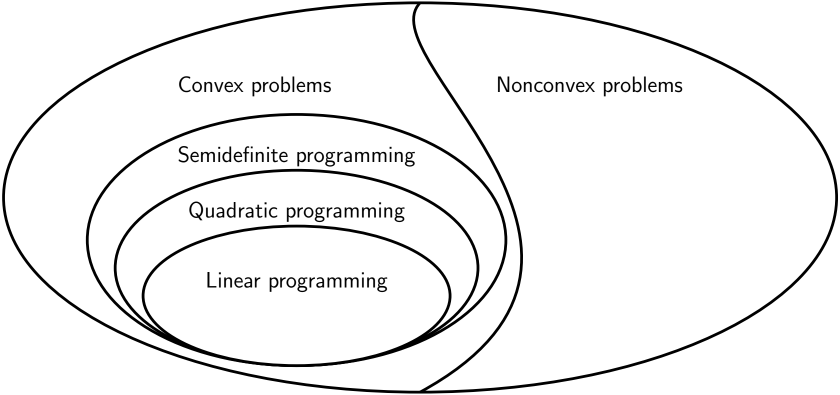
# Overview

- Introduction to mathematical programming problems
- Applications
- Classification of optimization problems
- (Linear algebra review)
- Convex optimization problems
- Nonconvex optimization problems
- Solving optimization problems

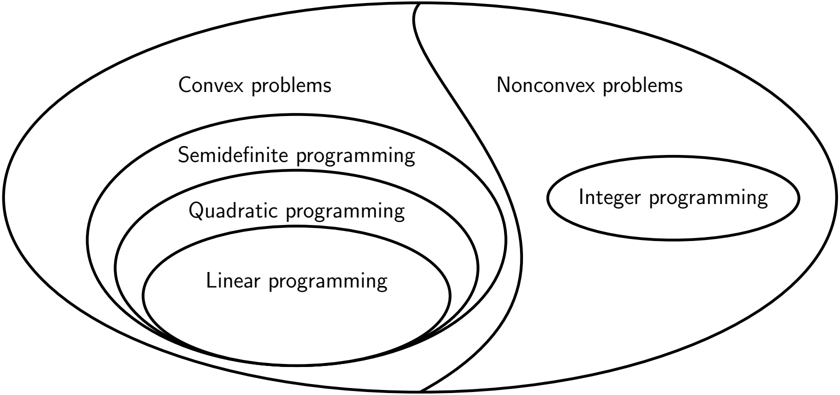
**Optimization problems**



**Optimization problems**



**Optimization problems**



# Overview

- Introduction to mathematical programming problems
- Applications
- Classification of optimization problems
- (Linear algebra review)
- Convex optimization problems
- Nonconvex optimization problems
- Solving optimization problems

# Notation

- A (column) vector with $n$ real-valued entries

$$x \in \mathbb{R}^n$$

$x_i$ denotes the $i$th entry

- A matrix with real-valued entries, $m$ rows, and $n$ columns

$$A \in \mathbb{R}^{m \times n}$$

$A_{ij}$ denotes the entry in the $i$th row and $j$th column

- The transpose operator $A^T$ switches rows and columns of a matrix

$$A_{ij} = (A^T)_{ji}$$

# Operations

- **Addition**: For $A, B \in \mathbb{R}^{m \times n}$,

$$(A + B)_{ij} = A_{ij} + B_{ij}$$

- **Multiplication**: For $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{n \times p}$

$$(AB)_{ij} = \sum_{k=1}^{n} A_{ik} B_{kj}, \quad AB \in \mathbb{R}^{m \times p}$$

associative: $A(BC) = (AB)C = ABC$
distributive: $A(B + C) = AB + AC$
*not* commutative: $AB \neq BA$
transpose of product: $(AB)^T = B^T A^T$

- **Inner product**: For $x, y \in \mathbb{R}^n$

$$x^T y = \langle x, y \rangle = \sum_{i=1}^{n} x_i y_i$$

- **Identity matrix**: $I \in \mathbb{R}^{n \times n}$ has ones on diagonal and zeros elsewhere, has property that $IA = A$

- **Matrix inverse**: For $A \in \mathbb{R}^{n \times n}$, $A^{-1}$ is unique matrix such that

$$AA^{-1} = I = A^{-1}A$$

(may not exist for all square matrices)
inverse of product: $(AB)^{-1} = B^{-1}A^{-1}$ *when $A$ and $B$ both invertible*

# Overview

- Introduction to mathematical programming problems
- Applications
- Classification of optimization problems
- (Linear algebra review)
- Convex optimization problems
- Nonconvex optimization problems
- Solving optimization problems

# Convex optimization problems

- An extremely powerful subset of all optimization problems
- Roughly speaking, allow for efficient (polynomial time) *global* solutions
- Beautiful theory
- *Lots* of applications (but be careful, lots of problems we'd like to solve are definitely *not* convex)
- At this point, a fairly mature technology (off-the-shelf libraries for medium-sized problems)

# Formal definition

- A convex optimization problem is a specialization of a mathematical programming problem

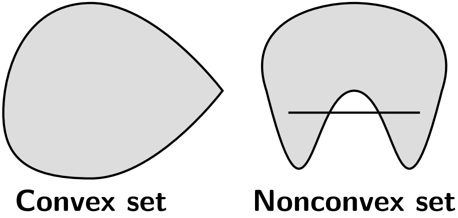
$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & x \in \mathcal{C} \end{aligned}$$

where $x \in \mathbb{R}^n$ is optimization variable, $f : \mathbb{R}^n \to \mathbb{R}$ is a *convex function* and feasible region $\mathcal{C}$ is a *convex set*

# Convex sets

- A set $\mathcal{C}$ is convex if, for a $x, y \in \mathcal{C}$ and $\theta \in [0,1]$,

$$\theta x + (1-\theta)y \in \mathcal{C}$$




- All of $\mathbb{R}^n$: $x, y \in \mathbb{R}^n \Longrightarrow \theta x + (1-\theta)y \in \mathbb{R}^n$

- Intervals: $\mathcal{C} = \{x \in \mathbb{R}^n : l \le x \le u\}$ (elementwise inequality)

- Linear inequalities: $\mathcal{C} = \{x \in \mathbb{R}^n : Ax \leq b\},\ A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$
  **Proof:**

$$\begin{aligned} x, y \in \mathbb{R}^n \in \mathcal{C} &\Longrightarrow Ax \leq b,\ Ay \leq b \\ &\Longrightarrow \theta Ax \leq \theta b, (1-\theta)Ay \leq (1-\theta)b \\ &\Longrightarrow \theta Ax + (1-\theta)Ay \leq b \\ &\Longrightarrow A(\theta x + (1-\theta)y) \leq b \\ &\Longrightarrow \theta x + (1-\theta)y \in \mathcal{C} \end{aligned}$$

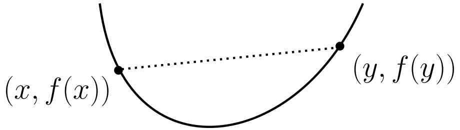
- Linear equalities $\mathcal{C} = \{x \in \mathbb{R}^n : Ax = b\},\ A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$

- Intersection of convex sets: $\mathcal{C} = \bigcap_{i=1}^m \mathcal{C}_i$ for convex sets $\mathcal{C}_i$, $i = 1, \ldots, m$

# Convex functions

- A function $f : \mathbb{R}^n \to \mathbb{R}$ is *convex* if, for any $x, y \in \mathbb{R}^n$ and $\theta \in [0, 1]$,

$$f(\theta x + (1-\theta)y) \le \theta f(x) + (1-\theta) f(y)$$




- $f$ is *concave* if $-f$ is convex

- $f$ is *affine* if $f$ is convex and concave, must have form

$$f(x) = a^T x + b$$

for $a \in \mathbb{R}^n$, $b \in \mathbb{R}$.

# Testing for convexity

- Convex function must "curve upwards" everywhere

- For functions with scalar input $f : \mathbb{R} \to \mathbb{R}$, equivalent to condition that $f''(x) \geq 0, \;\; \forall x$

- For vector-input functions, corresponding condition is that

$$\nabla_x^2 f(x) \succeq 0$$

where $\nabla_x^2 f(x)$ is the *Hessian* of $f$,

$$(\nabla_x^2 f(x))_{ij} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}$$

and $\succeq 0$ denotes *positive definiteness*, the condition that for all $z \in \mathbb{R}^n$,

$$z^T \left(\nabla_x^2 f(x)\right) z \geq 0$$

# Examples

- Exponential: $f(x) = \exp(ax)$, $[f''(x) = a^2 \exp(ax) \geq 0 \forall x]$

- Negative logarithm: $f(x) = -log(x)$, $[f''(x) = 1/x^2 \geq 0 \forall x]$

- Squared Euclidean distance: $f(x) = x^T x$ [After some derivations, you'll find that $\nabla_x^2 f(x) = I, \forall x$, which is positive semidefinite since $z^T I z = z^T z \geq 0$]

- Euclidean distance: $f(x) = \|x\|_2 = \sqrt{x^T x}$

- Non-negative weighted sum of convex functions:

$$f(x) = \sum_{i=1}^{m} w_i f_i(x)$$

where $w_i \geq 0$ and $f_i$ convex, $\forall i = 1, \ldots, m$

- Negative square root: $f(x) = -\sqrt{x}$ convex for $x \geq 0$

- Log-sum-exp: $f(x) = \log(e^{x_1} + e^{x_2} + \ldots + e^{x_n})$

- Maximum of convex functions:

$$f(x) = \max_{i=1}^{m} f_i(x)$$

  where $f_i$ convex, $\forall i = 1, \ldots, m$

- Composition of convex and affine function: if $f(y)$ convex in $y$, $f(Ax - b)$ is convex in $x$

- Sublevel sets: for $f$ convex, $\mathcal{C} = \{x : f(x) \leq c\}$ is a convex set

# Convex optimization problems (again)

- Using sublevel sets property, it's more common to write generic convex optimization problems as

$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \leq 0 \;\; i = 1, \ldots, m \\ & h_i(x) = 0 \;\; i = 1, \ldots, p \end{aligned}$$

  where $f$ (objective) is convex, $g_i$'s (inequality constraints) are convex, and $h_i$'s (equality constraints) are affine

- Key property of convex optimization problems: *all local solutions are global solutions*

- **Definition**: a point $x$ is *globally optimal* if $x$ is feasible and there is no feasible $y$ such that $f(y) < f(x)$

- **Definition**: a point $x$ is *locally optimal* if $x$ is feasible and there exists some $R > 0$ such that for all feasible $y$ with that $\|x - y\|_2 \leq R$, $f(x) \leq f(y)$

- **Theorem**: for a convex optimization problem, all locally optimal points are globally optimal

  **Proof** (by contradiction): Suppose there exists feasible $y$ such that $f(y) < f(x)$. Pick the point

$$z = \theta x + (1 - \theta) y, \quad \theta = 1 - \frac{R}{2\|x - y\|_2}.$$

  We will show $\|x - z\|_2 < R$, $z$ is feasible, and $f(z) < f(x)$, contradicting local optimality.

**Proof** (cont)

$$\begin{aligned}\|x - z\|_2 &= \left\| x - \left( \left( 1 - \frac{R}{2\|x-y\|_2} \right) x + \frac{R}{2\|x-y\|_2} y \right) \right\|_2 \\ &= \left\| \frac{R}{2\|x-y\|_2}(x-y) \right\|_2 = R/2 < R\end{aligned}$$

Furthermore, by convexity of feasible set, $z$ is feasible. This shows all the points above, contradicting the supposition that there was a non-local feasible $y$ with $f(y) < f(x)$. □

# Example: least-squares

- General (potentially multi-variable) least-squares problems is given by

$$\underset{x}{\text{minimize}} \quad \|Ax - b\|_2^2$$

for optimization variable $x \in \mathbb{R}^n$, data matrices $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, and where $\|y\|_2^2$ is the $\ell_2$ norm of vector $y \in \mathbb{R}^n$ is

$$\|y\|_2^2 = y^T y = \sum_{i=1}^{m} y_i^2.$$

- A convex optimization problem (why?)

# Example: Weber point

- Weber point in $n$ dimensions

$$\underset{x}{\text{minimize}} \quad \sum_{i=1}^{m} \|x - a_i\|_2$$

  where $x \in \mathbb{R}^n$ is optimization variable and $a_1, \ldots, a_m$ are problem data.

- A convex optimization problem (why?)

# Example: linear programming

- General form of a linear program

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & c^T x \\ \text{subject to} \quad & Ax = b \\ & Fx \leq g \end{aligned}$$

  where $x \in \mathbb{R}^n$ is optimization variable, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $F \in \mathbb{R}^{p \times n}$, $g \in \mathbb{R}^p$ are problem data

- You may have also seen this with just the inequality constraint $x \geq 0$ (these are equivalent, why?)

- A convex problem (affine objective, convex constraints)

# Example: quadratic programming

- General form of quadratic program

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & x^T Q x + r^T x \\ \text{subject to} \quad & Ax = b \\ & Fx \leq g \end{aligned}$$

  where $x \in \mathbb{R}^n$ is optimization variable, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $F \in \mathbb{R}^{p \times n}$, $g \in \mathbb{R}^p$ are problem data

- A convex problem if $Q$ is semidefinite

# Overview

- Introduction to mathematical programming problems
- Applications
- Classification of optimization problems
- (Linear algebra review)
- Convex optimization problems
- Nonconvex optimization problems
- Solving optimization problems

# Nonconvex problems

- Any optimization problem

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \leq 0 \;\; i = 1, \ldots, m \\ & h_i(x) = 0 \;\; i = 1, \ldots, p \end{aligned}$$

  where $f$ or any $g_i$ not convex, any $h_i$ not affine, is a nonconvex problem

- Two classes of approaches for solving non-convex problems: *local* and *global*

- **Local methods**: Given some initial point $x_0$, repeatedly search "nearby" points until finding a (feasible) solution $\hat{x}$ that is better than all it's nearby points
  - Typically, same approximate computational complexity as convex methods (often just use convex methods directly)
  - Can fail to find any feasible solution at all
- **Global methods**: Find actual optimal solution $x^\star$ over the entire domain of feasible solution
  - Typically, exponential time complexity
- Both are used extremely frequently in practice, for different types of problems

# Integer programming

- A class of nonconvex problems that we will look at in more detail later:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & c^T x \\ \text{subject to} \quad & Ax = b \\ & Fx \leq g \\ & x \in \mathbb{Z} \end{aligned}$$

  with optimization variable $x$, problem data $A, b, F, g$ as in linear programming

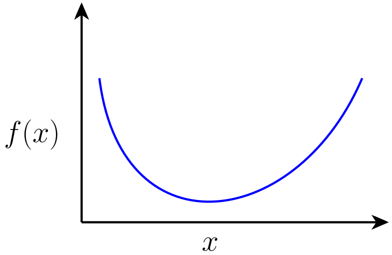
- A nonconvex problem (why?)

# Overview

- Introduction to mathematical programming problems
- Applications
- Classification of optimization problems
- (Linear algebra review)
- Convex optimization problems
- Nonconvex optimization problems
- Solving optimization problems

# Solving optimization problems

- Starting with the unconstrained, one dimensional case




- To find minimum point $x^\star$, we can look at the derivative of the function $f'(x)$: any location where $f'(x) = 0$ will be a "flat" point in the function

- For convex problems, this is guaranteed to be a minimum (instead of a maximum)

- Generalization for multivariate function $f : \mathbb{R}^n \to \mathbb{R}$: the *gradient* of $f$ must be zero

$$\nabla_x f(x) = 0$$

- For $f$ defined as above, gradient is a $n$-dimensional vector containing partial derivatives with respect to each dimension

$$(\nabla_x f(x))_i = \frac{\partial f(x)}{\partial x_i}$$

- Important: only a sufficient condition for unconstrained optimization

# How do we find $\nabla_x f(x) = 0$?

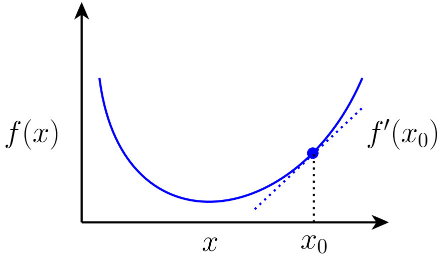
- **Direct solution**: Analytically compute $\nabla_x f(x)$ and set resulting equation to zero
  - Example: quadratic function

    $$f(x) = x^T Q x + r^T x \implies \nabla_x f(x) = 2Qx + r \implies x^\star = -\frac{1}{2} Q^{-1} r$$

    for $Q \in \mathbb{R}^{n \times n}$, $r \in \mathbb{R}^n$

  - Will be a minimum assuming $Q$ is positive definite

- **Gradient descent**: Take small steps in the direction of the current (negative) gradient
  - Intuition: negative gradient of a function always points "downhill" (actually points downhill in the steepest possible direction, for multivariate function)




  - Repeat: $x \leftarrow x - \alpha \nabla_x f(x)$ where $\alpha \in \mathbb{R} > 0$ is a *step size*

- **Newton's method**: Use root-finding algorithm to find solution to (nonlinear) equation $\nabla_x f(x) = 0$
  - Newton's method: given $g : \mathbb{R} \to \mathbb{R}$, find $g(x) = 0$ by iteration

$$x \leftarrow x - \frac{f(x)}{f'(x)}$$

  - Multivariate generalization for finding $\nabla_x f(x) = 0$

$$x \leftarrow x - (\nabla_x^2 f(x))^{-1} \nabla_x f(x)$$

  - Each iteration is more expensive than gradient descent, but can converge to numerical precision much faster

# Constrained optimization

- Now consider problem with constraints

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \leq 0 \;\; i = 1, \ldots, m \end{aligned}$$

- **Projected gradient**: if we can come up with an "easy" *projection operator* $\mathcal{P}(x)$ such that for any $x$, $\mathcal{P}(x)$ returns "closest" feasible point

$$x \leftarrow \mathcal{P}(x - \alpha \nabla_x f(x))$$

  - Example $\mathcal{C} = x \geq 0$,

$$\mathcal{P}(x) = \max\{x, 0\} \text{ (elementwise)}$$

- **Barrier method**: Approximate problem via unconstrained optimization

$$\underset{x}{\text{minimize}} \quad f(x) - t\sum_{i=1}^{m} \log(-g_i(x))$$

as $t \to 0$, this approaches original problem

  - Can quickly lead to numerical instability, requires a lot of care to get right
  - Equality constraints need to be handled separately

# Practically solving optimization problems

- The good news: for many classes of optimization problems, people have already done all the “hard work” of developing numerical algorithms

- A wide range of tools that can take optimization problems in “natural” forms and compute a solution

- Some well-known libraries: CVX (MATLAB), YALMIP (MATLAB), AMPL (custom language), GAMS (custom language)

# cvxpy

- We'll be using a relative newcomer to the game: **cvxpy** (`http://github.com/cvxgrp/cvxpy`)

- Python library for specifying and solving convex optimization problems

- Very much "alpha" software, under active development

- Constrained Weber point, given $a_i \in \mathbb{R}^2$, $i = 1, \ldots, m$

$$\underset{x}{\text{minimize}} \quad \sum_{i=1}^{m} \|x - a_i\|_2, \quad \text{subject to} \quad x_1 + x_2 = 0$$

- cvxpy code

```
import cvxpy as cp
import cvxopt

n = 2
m = 10
A = cvxopt.normal(n,m)

x = cp.Variable(n)
f = sum([cp.norm(x - A[:,i],2) for i in range(m)])
constraints = [sum(x) == 0]
result = cp.Problem(cp.Minimize(f), constraints).solve()
print x.value
```

# Take home points

- Optimization (and formulating AI tasks as optimization problems) has been one of the primary themes in AI of the past 15 years

- Convex optimization problems are a useful (though restricted) subset that can be solved efficiently and which still find a huge number of applications

- Many algorithms for solving optimization problems, but a lot of the "hard work" has been done by freely available libraries