ORF307 – Optimization
8. Piecewise Linear Optimization
Ed Forum question:

Software workflow

```python
import cvxpy as cp

# make decision variable
x = cp.Variable(2)

# define objective
objective = x[0]

# define constraints
constraints = [cp.norm_inf(x) <= 1, # inequalities
               x[0] + x[1] == -1]  # equalities

# make problem and solve
prob = cp.Problem(cp.Minimize(objective), constraints)
prob.solve()
```

• What is happening here?
Today: Piecewise Linear Optimization

- Vector norms
- Turning vector norm problems in LPs
- Piecewise linear optimization
- Support vector machines
- Sparse regression
Vector norms

Euclidean norm:

$$\|x\|_2 = \sqrt{\sum_i x_i^2}$$

1–norm (Manhattan or taxicab norm):

$$\|x\|_1 = \sum_i |x_i|$$

∞–norm:

$$\|x\|_\infty = \max_i |x_i|$$
A regression example (revisited)

Fit a linear function $f(z) = a + bz$ to $m$ data points $(z_i, f_i)$:

Approximation problem $Ax \approx b$ where

$$\begin{bmatrix}
1 & z_1 \\
\vdots & \vdots \\
1 & z_m
\end{bmatrix}
\begin{bmatrix}
a \\
b
\end{bmatrix} \approx
\begin{bmatrix}
f_1 \\
\vdots \\
f_m
\end{bmatrix}$$

Recall our regression problem:

$$\text{minimize } \sum_{i=1}^{n} |Ax - b|_i = \|Ax - b\|_1.$$ 

Why is this a linear program?
A simple example

Find a point in the unit box $X$, and restricted to the line $L$, that is as far left as possible.

minimize $x_2$

subject to $x_1 + x_2 = -1$

$\|x\|_{\infty} \leq 1$

The (nonlinear) norm function now appears in the constraints.

Why is this a linear program?
Linear, affine and convex functions

Linear function: \( f(x) = a^T x \)

Affine function: \( f(x) = a^T x + b \)
Linear, affine and convex functions

Convex function:

\[ f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y), \quad \forall x, y \in \mathbb{R}^n, \; \alpha \in [0, 1] \]
Convex piecewise affine functions

\[ f(x) = \max_{i=1,\ldots,m} (c_i^T x + d_i) \]
Piecewise affine minimization

minimize \[ \max_{i=1,\ldots,m} \left( c_i^T x + d_i \right) \]

As a linear program:

minimize \( t \)

subject to \( c_i^T x + d_i \leq t \)

\( \vdots \)

\( c_m^T x + d_m \leq t \)
Piecewise affine minimization

minimize \[ \max_{i=1, \ldots, m} (c_i^T x + d_i) \]
subject to \[ \sum_{i=1}^m c_i^T x \leq \sum_{i=1}^m d_i \]

As a linear program (matrix form):

minimize \( \tilde{c}^T \tilde{x} \)
subject to \( \tilde{C} \tilde{x} \leq \tilde{d} \)

\[ \begin{array}{c}
\tilde{x} = \begin{bmatrix} x \\ t \end{bmatrix}, \quad \tilde{c} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \\
\tilde{C} = \begin{bmatrix} c_1^T & -1 \\ \vdots & \vdots \\ c_m^T & -1 \end{bmatrix}, \quad \tilde{d} = \begin{bmatrix} -d_1 \\ \vdots \\ -d_m \end{bmatrix}
\end{array} \]
Infinity norm approximation

minimize $\|Ax - b\|_\infty$

$\begin{aligned}
(Ax - b)_i & \leq t \\
-(Ax - b)_i & \leq t
\end{aligned}$

The $\infty$-norm of $m$-vector $y$ is:

$$
\|y\|_\infty = \max_{i=1,\ldots,m} |y_i| = \max_{i=1,\ldots,m} \max\{y_i, -y_i\}
$$

Equivalent problem

minimize $t$
subject to $-t1 \leq Ax - b \leq t1$

$\begin{aligned}
Ax - 1t & \leq b \\
-Ax - 1t & \leq -b
\end{aligned}$

Matrix notation

minimize
$$
\begin{bmatrix}
0 & T \\
1 & t
\end{bmatrix}
$$
subject to
$$
\begin{bmatrix}
A & -1 \\
-A & -1
\end{bmatrix}
\begin{bmatrix}
x \\
t
\end{bmatrix}
\leq
\begin{bmatrix}
+b \\
-b
\end{bmatrix}
$$
The sum of piecewise affine functions

$$\text{minimize} \left( \left[ \max_{i=1,\ldots,p} (a_i^T x + b_i) \right] + \left[ \max_{i=1,\ldots,m} (c_i^T x + d_i) \right] \right)$$

As a linear program:

$$\text{minimize} \quad t_1 + t_2$$

subject to

$$a_i^T x + b_i \leq t_1 \quad i = 1, \ldots, p$$
$$c_i^T x + d_i \leq t_2 \quad i = 1, \ldots, m$$
1—norm approximation

\[
\text{minimize } \|Ax - b\|_1
\]

Rewrite as a sum of piecewise linear functions. Assume \( A \) is \( m \times n \).

\[
\text{minimize } \sum_{i=1}^{m} \max\left\{ (Ax - b)_i, -(Ax - b)_i \right\}
\]
1—norm approximation

\[
\text{minimize } \| Ax - b \|_1
\]

minimize \( t_1 + \ldots + t_m \)

subject to \( (Ax - b)_i \leq t_i \quad i = 1, \ldots, m \)
\[
-(Ax - b)_i \leq t_i \quad i = 1, \ldots, m
\]

\[
\iff
\]

minimize \( 1^T t \)

subject to \( -t \leq (Ax - b) \leq t \)

\( t \in \mathbb{R}^m \)
Summary: regression in 1 and infinity norms

1-norm minimization:

minimize $\|Ax - b\|_1$

Equivalent to:

minimize $1^T t$

subject to $-t \leq (Ax - b) \leq t$

Absolute value of every element $(Ax - b)_i$ is bounded by a component of the vector $t$

∞-norm minimization:

minimize $\|Ax - b\|_\infty$

Equivalent to:

minimize $t$

subject to $-1t \leq (Ax - b) \leq 1t$

Absolute value of every element $(Ax - b)_i$ is bounded by the same scalar $t$
Example: converting to an LP

minimize $\|Ax - b\|_\infty$

subject to $\|x\|_1 \leq k$

\[
\min t \\
-1t \leq Ax - b \leq 1t \\
-1 \leq x \leq 1 \\
1^T s \leq k
\]
Comparison with least-squares

Histogram of residuals $Ax - b$ with randomly generated $A \in \mathbb{R}^{200 \times 80}$

$x_2 = \text{argmin } \|Ax - b\|_2^2, \quad x_1 = \text{argmin } \|Ax - b\|_1$

1-norm distribution is wider with a high peak at zero
Sparse signal recovery via $\ell_1$-norm minimization

$\hat{x} \in \mathbb{R}^n$ is unknown signal, known to be sparse

We make linear measurements $y = Ax$ with $A \in \mathbb{R}^{m \times n}, m < n$

Estimate signal with smallest $\ell_1$-norm, consistent with measurements

minimize $\|x\|_1$
subject to $Ax = y$
Sparse signal recovery via $1$–norm minimization

$\hat{x} \in \mathbb{R}^n$ is unknown signal, known to be sparse

We make linear measurements $y = A\hat{x}$ with $A \in \mathbb{R}^{m \times n}, m < n$

Estimate signal with smallest $\ell_1$-norm, consistent with measurements

$$\begin{align*}
\text{minimize} & \quad \|x\|_1 \\
\text{subject to} & \quad Ax = y
\end{align*}$$

Equivalent linear optimization

$$\begin{align*}
\text{minimize} & \quad 1^T u \\
\text{subject to} & \quad -u \leq x \leq u \\
& \quad Ax = y
\end{align*}$$
Sparse signal recovery via $1$–norm minimization

Example

Exact signal $\hat{x} \in \mathbb{R}^{1000}$
10 nonzero components
Random $A \in \mathbb{R}^{100 \times 1000}$

The least squares estimate cannot recover the sparse signal

The $1$-norm estimate is **exact**
Software does most of this for you!

**1-norm minimization:**

\[
\text{minimize } \|Ax - b\|_1
\]

```python
import numpy as np
import cvxpy as cp

m = 200; n = 80

A = np.random.randn(200, 80)
b = np.random.randn(200)
x = cp.Variable(80)

cost = cp.norm(A @ x - b, 1)
prob = cp.Problem(cp.Minimize(cost))
prob.solve()
```

**∞-norm minimization:**

\[
\text{minimize } \|Ax - b\|_{\infty}
\]

```python
import numpy as np
import cvxpy as cp

m = 200; n = 80

A = np.random.randn(200, 80)
b = np.random.randn(200)
x = cp.Variable(80)

cost = cp.norm(A @ x - b, np.inf)
prob = cp.Problem(cp.Minimize(cost))
prob.solve()
```
Support vector machines

Problem data:

- A collection of **feature vectors** $x_1, \ldots, x_N$
- For each point, a **label** $s_i \in \{-1, +1\}$

Find a line (i.e. a **hyperplane**) that separates the two groups.
Support vector machines

As a feasibility problem:

find \((a, b)\)

subject to \(a^T x_i + b > 0, \text{ if } s_i = +1\)

\(a^T x_i + b < 0, \text{ if } s_i = -1\)

Important! Inequalities are homogeneous in \((a, b)\).
Support vector machines

As a feasibility problem:

find \((a, b)\)
subject to \(a^T x_i + b > 0, \text{ if } s_i = +1\)
\(a^T x_i + b < 0, \text{ if } s_i = -1\)

As a linear program:

minimize \(0\)
subject to \(s_i(a^T x_i + b) \geq 1, \quad i = 1, \ldots, N\)
Which separator is best?

Find that line that gives maximum separation.

As a quadratic program:

minimize $\|a\|^2$

subject to $s_i(a^T x_i + b) \geq 1$, $i = 1, \ldots, N$
What if the points can’t be separated?

Each of our constraints was:

\[ s_i(a^T x_i + b) \geq 1 \]

\[ |1 - s_i(a^T x_i + b)| \leq 0 \]

The constraint violation of any point is:

\[ \max\{0, 1 - s_i(a^T x_i + b)\} \]
Approximate linear separation

Minimize the sum of the violations:

$$\text{minimize} \sum_{i=1}^{N} \max\{0, 1 - s_i(a^T x_i + b)\}$$

As a linear program?:

$$\min \sum_{i=1}^{N}$$

$$0 \leq t_i$$

$$1 - s_i(a^T x_i + b) \leq t_i$$

$$i = 1 \ldots N$$
What if they really can’t be separated?
What if they really can’t be separated?

\[ x_3 = x_1^2 + x_2^2 \]
What if they really can’t be separated?
Lasso methods
Least absolute shrinkage and selection operator (Lasso)

How do I find a function fits this data?:

- The function is unknown
- I want to use regression
- I don’t want to overfit
Lasso methods

Least absolute shrinkage and selection operator (Lasso)

How do I find a function fits this data?:

- The function is unknown
- I want to use regression
- I don’t want to overfit

Let’s try this:

\[ f(t) = c_0 + c_1 t + c_2 \sin(t) + c_3 \cos(t) + c_4 \sin(2t) + c_5 \cos(2t) + c_6 t^2 + c_7 \sqrt{t} + \ldots \]
Lasso methods

\[ f(t) = c_0 + c_1 t + c_2 \sin(t) + c_3 \cos(t) + c_4 \sin(2t) + c_5 \cos(2t) + c_6 t^2 + c_7 \sqrt{t} + \ldots \]
Lasso methods

\[ f(t) = c_0 + c_1 t + c_2 \sin(t) + c_3 \cos(t) + c_4 \sin(2t) + c_5 \cos(2t) + c_6 t^2 + c_7 \sqrt{t} + \ldots \]

Data points \( \{f_i, t_i\} \)
Next time

- Geometry of Polyhedra and Linear Programs
- Simple optimality conditions
References

• Bertsimas and Tsitsiklis: Introduction to Linear Programming
  • Chapter 1.3: piecewise linear optimization

• R. Vanderbei: Linear Programming — Foundations and Extensions
  • Chapter 12.4,12.7: 1-norm regression and SVMs