**Problem 1**

1. (a) size of the vectors

= [P x 1]

= [n x 1]

= [m x 1]

(b) = ∈ ℝP

(c) Optimization problem P

Min t B

S.t A

t B

Problem (1) can be reformulated as

Find

S.t P

(d) No, you can’t feed P to an LP Solver and give a solution.

1. (a) Bt0

Size of = [n x 1]

(b) = [1 1 1]’

0t B = -8 (Since the value is negative, 0 doesn’t satisfy the problem

3) (a) Min t

S.t At -

(b) Min t

S.t At -Bt

Yes, this problem can be solved using a LP solver.

(c) Converting the above dual to standard form, we get

Min t

S.t At + Bt

Substituting by

The Problem becomes,

Min t

S.t At + Bt

Since it is a minimization problem we know that value of t  0, and the feasible solution for a dual is the feasible solution for primal, therefore since is the solution to the dual problem, it is the solution to problem (1) too.

(d) First solve for x, using random matrix A & vector b.

Then pass the value of the product B and x, to the L1 norm to find the value of v.

We will come to find out that value of v is unbounded, as v can take value.

**Prob3d.m** is the mat lab code used to solve problem (1).

The value of EXITFLAG will be -3 which implies the problem is unbounded.

**Problem 2**

1. (a) No we can’t find an ***f*** *that* can exactly fit all the data points.

(b) **L1 norm**

**min** t1+ t2+ t3+ t4+ t5+ t6+ t7

**s.t**  |ax1 + b – y1| ≤ t1

|ax2 + b – y2| ≤ t2

|ax3 + b – y3| ≤ t3

|ax4 + b – y4| ≤ t4

|ax5 + b – y5| ≤ t5

|ax6 + b – y6| ≤ t6

|ax7 + b – y7| ≤ t7

t1, t2, t3, t4, t5, t6, t7 ≥ 0

**L∞ norm**

**min** t

**s.t**  |ax1 + b – y1| ≤ t

|ax2 + b – y2| ≤ t

|ax3 + b – y3| ≤ t

|ax4 + b – y4| ≤ t

|ax5 + b – y5| ≤ t

|ax6 + b – y6| ≤ t

|ax7 + b – y7| ≤ t

t≥ 0

(c) Script **prob1.m** and functions **L1norm.m, LInfnorm.m, nDegreePolynomial.m & plotGraph.m** are used to solve the given problem.

First call **prob1.m** and you will get a prompt saying “**enter the value of n:**” give 1 for this prob.



(d) For this problem call prob2.m and enter the value for n as 1.



By comparing the above two figures we could see that L**∞** norm is very sensitive and L1 norm is insensitive to presence of outliers

1. (a) **L1 norm**

**min** t1+ t2+ t3+ t4+ t5+ t6+ t7

**s.t**  |a1x1 + a2x12 + b – y1| ≤ t1

|a1x2 + a2x22 + b – y2| ≤ t2

|a1x3 + a2x32 + b – y3| ≤ t3

|a1x4 + a2x42 + b – y4| ≤ t4

|a1x5 + a2x52 + b – y5| ≤ t5

|a1x6 + a2x62 + b – y6| ≤ t6

|a1x7 + a2x72 + b – y7| ≤ t7

t1, t2, t3, t4, t5, t6, t7 ≥ 0

**L∞ norm**

**min** t

**s.t**  |a1x1 + a2x12 + b – y1| ≤ t

|a1x2 + a2x22 + b – y2| ≤ t

|a1x3 + a2x32 + b – y3| ≤ t

|a1x4 + a2x42 + b – y4| ≤ t

|a1x5 + a2x52 + b – y5| ≤ t

|a1x6 + a2x62 + b – y6| ≤ t

|a1x7 + a2x72 + b – y7| ≤ t

t≥ 0

(b) As stated above, call prob1.m and prob2.m but give the value of n as 2.





(c) Performance of the affine and the polynomial model for each norm can be inferred from the below Error Histograms

**For n = 1 (Affine Model)**



**For n=2 (Polynomial Model)**



1. **nDegreePolynomial.m** function handles polynomial models with up to a degree N that will be given as an input.