Pattern Recognition: Assignment 11

Due on Tuesday, July 17 2012, 23:59

Bernhard X. Kausler (TA), Summer Term 2012

patternrecognition@hci.iwr.uni-heidelberg.de

`http://hci.iwr.uni-heidelberg.de/MIP/Teaching/pr/`

Linear Programming

In this exercise you will explore linear programming (LP). Firstly, you will formulate and solve a LP for yourself and learn about the dual of your program. Secondly, you will use LP to implement the L^1-regression method, which will be more robust against outliers than ordinary least-squares regression.

Prob. 1: Problem Solving using Linear Programming

(a) Formulate a linear program (LP) (2 point)

Suppose you want to earn some money at the flea market next weekend with the fruit trees in your garden. You can either harvest cherries or apples at the following rates:

Cherries	2 kg/hour
Apples	1.4 kg/hour

They also earn different profits at your local flea market:

Cherries	2.5 Eur/kg
Apples	3.0 Eur/kg

Based on your previous flea market experience you place some upper bounds on the demand at the flea market next weekend:

Cherries	60 kg
Apples	40 kg

Given that there are 40 hours of harvesting time available this week, how much cherries and how much apples should you pluck to maximize your profit? Formulate that question as a linear program.

(b) Solve the LP (2 point)

Solve your LP with the online LP solver at `http://www.zweigmedia.com/RealWorld/simplex.html`. What's the maximal profit. How much cherries and apples do you have to sell then?

(c) Dual problem (4 point)

Write down the dual linear program to your primal program. Solve it using the online solver and give the values of the objective and the dual variables at optimality.
Compare the optimal primal and dual objective values. Did you expect this result? Why?
Finally, give the dual of the dual program. Is the result conform with what you expected from your knowledge of duality theory? Why or why not?

Prob. 2: Robust Regression Using Linear Programming

The file `cpu_load.csv` contains real world data of the cpu load `CPU_UTILIZE` (in arbitrary units) depending on the number of request to a database `BUFFER_GETS`. The relationship of the two is governed by a linear model. Unfortunately, we observe some outliers in the cpu load.

(a) Try a least-squares regression (2 point)

Make a scatter plot of the cpu load depending on the number of database requests. You should see a linear dependence with some outliers. Fit a linear model using ordinary least squares regression and add the fitted line to the plot. Is the fit a good model for the data? Comment.

(b) L^1-regression as a LP (6 point)

As an alternative to the mean, the median is another point estimate of the "average", which is more robust against outliers (think about it!). Similarly, least-absolute-deviations or L^1-regression is more robust against outliers than ordinary least-squares:

$$\hat{x} = \operatorname*{argmin}_x ||y - Ax||$$

There is no closed form solution to this minimization problem, but it can be solved as a linear program. Reformulate L^1-regression as a LP using the tricks you learned in the lecture.

(c) Implement and try L^1-regression (4 point)

Implement L^1-regression using the program you found in the previous subsection and fit a linear model to the cpu-load data. Make a plot as before. Compare your result with the least-squares fit and comment on the robustness against outliers.
[Use `linprog` in Matlab.]

Regulations

Please hand in the matlab code, figures and explanations (describing clearly which belongs to which). Non-trivial sections of your code should be explained with short comments, and variables should have self-explanatory names. Plots should have informative axis labels, legends and captions. Please enclose everything into a single PDF document (e.g. use the publish command of MATLAB for creating a LaTeX document and run latex, dvips and ps2pdf or copy and paste everything into an

office document and convert to PDF). Please email the PDF to patternrecognition@hci.iwr.uni-heidelberg.de before the deadline specified below. You may hand in the exercises in teams of two people, which must be clearly named on the solution sheet (one email is sufficient). Discussions between different teams about the exercises are encouraged, but the code must not be copied verbatim (the same holds for any implementations which may be available on the WWW). Please respect particularly this rule, otherwise we cannot give you a passing grade.