Basis pursuit L1 minimization and its biomedical applications

[Code Documentation]

Script	Language	Source
`Python Intro/projected_gradient.py`	Python	/

Overview

Preliminary experiments with constrained optimization problems and procedures by implementing projected gradient descent.

$\min f(x)\ s.t.\ Ax \leq b$

Parameters

- *objfun*: lambda function returning objective function value and gradient
- *A:* inequality constraint matrix
- *b:* inequality observations vector
- x_0: starting feasible vector

Returns

- x_k: solution vector

Script	Language	Source
`Python Intro/signal_theory.py`	Python	/

Overview

Simulation script comparing a relaxation of the problem using L_2 minimization and basis pursuit solved by Interior-Point-Method from *scipy*.

Script	Language	Source
`Python Intro/sumofsinesfitting.py`	Python	/

Overview

Simulation script solving the second application covering ECG signal reconstruction by Interior-Point-Method from *scipy*.

Script	Language	Source
`Python Intro/wave_plotting.py`	Python	/

Overview

Testing script containing the simulation of different waves (square, sawtooth, triangle).

Script	Language	Source
`Python Intro/waves.py`	Python	/

Overview

Utility functions providing the implementation of different waves (square, sawtoot, triangle)

Parameters

- *f*: frequency parameter
- *t:* time input

Returns

- a: amplitude

Script	Language	Source
MatLab/buildSignal.py	Matlab	/

Overview

Utility function uniformly sampling from a signal over time in order to build the constraint matrix A and the vector of observations b, providing also the sampled points and its value over time.

Parameters

- *m*: number of observations
- *n:* number of variables
- *sign_function:* lambda function outputting amplitude based on time input
- *interval:* time interval for the sampled points

Returns

- *A:* constraint matrix
- *b:* observations vector
- x_m: sampled time coordinates
- y_m: amplitude coordinates

Script	Language	Source
MatLab/forrest_tomlin.m	Matlab	/

Overview

Forrest-Tomlin update procedures producing a new version of the LU decomposition of a matrix undergoing a column update. The adaptation for row updates in dual simplex is based on the fact that $LU = A \; \Rightarrow \mathbf{U}^{\mathbf{T}}\mathbf{L}^{\mathbf{T}} = \mathbf{A}^{\mathbf{T}}$

Parameters

- A_q: updated column vector
- *q:* column index that has to be updated
- *L:* old lower triangular matrix from LU decomposition
- *U:* old upper triangular matrix from LU decomposition

Returns

- *L_up:* updated upper triangular matrix from LU decomposition
- *U_up:* updated upper triangular matrix from LU decomposition

Script	Language	Source
`MatLab/projgrad.py`	Python	/

Overview

Preliminary experiments with constrained optimization problems and procedures by implementing projected gradient descent.

$\min f(x)\ s.t.\ Ax \leq b$

Parameters

- *objfun*: lambda function returning objective function value and gradient
- *A:* inequality constraint matrix
- *b:* inequality observations vector
- x_0: starting feasible vector

Returns

- x_k: solution vector

Script	Language	Source
`MatLab/reconstruct.m`	Matlab	/

Overview

Procedure for uniformly random sampling *n* points over time based on a solution vector *x*.

Parameters

- *x*: solution vector
- *n_points:* number of points to sample
- *interval:* interval over time

Returns

- x_{res}: sampled time coordinates
- y_{res}: amplitude coordinates

Script	Language	Source
`MatLab/dual_simplex.m`	Python	/

Overview

Revised dual simplex implementation including FT update producing primal and dual solution vectors.

Parameters

- *A:* constraint matrix
- *b:* observations vector
- *B:* basis projection
- *tol_set:* tolerance for checking emptiness of primal set
- *tol_opt:* tolerance for optimal set condition
- *MAX_ITER:* maximum number of iterations allowed
- *state:* state of the problem
- *verbose:* verbosity flag

Returns

- x: primal solution vector
- y: dual solution vector

Script	Language	Source
`MatLab/tools/metrics.m`	Python	/

Overview

Utility script for easily calculating mean squared error and Pearson correlation coefficient based on the values of two vectors

Parameters

- *sol:* solution vector
- *tar:* target vector

Returns

- *mse*: mean squared error
- *corr*: pearson correlation coefficient

Script	Language	Source
`MatLab/ECG/simulate_ecg.m`	Python	[1]

Overview

Utility script for easily generating fake ECG signals providing beats per minute, frequency of sampling and the width of time interval.

Parameters

- *bpm:* beats per minute
- *rate:* frequency of sampling
- *sec:* seconds for the acquisition

Returns

- *x*: sampled time coordinates
- *y*: relative amplitude coordinates

Sources

[1] karthik raviprakash (2021). ECG simulation using MATLAB (https://www.mathworks.com/matlabcentral/fileexchange/10858-ecg-simulation-using-matlab), MATLAB Central File Exchange. Retrieved March 11, 2021.