LIBRARY FOR QUADRATIC PROGRAMMING

VOJTĚCH FRANC
XFRANCV AT CMP.FELK.CVUT.CZ

Introduction

LIBQP is C library which implements algorithms for solving two special instances of convex Quadratic Programming (QP):

QP task with simplex constraints. This QP task is defined as follows

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x} + \mathbf{x}^T\mathbf{f} \\ \text{subject to} \quad & \sum_{i \in I_k} x_i = b_k\,, \quad k \in \mathrm{S}_{\mathrm{equ}} \\ & \sum_{i \in I_k} x_i \leq b_k\,, \quad k \in \mathrm{S}_{\mathrm{neq}} \\ & x_i \geq 0\,, \quad i \in I\,. \end{aligned}$$

where $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$ is the optimized vector, $\mathbf{H} \in \mathbb{R}^{n \times n}$ is a symmetric positive semi-definite matrix, $\mathbf{f} \in \mathbb{R}^n$ is a vector, $I = \{1, \ldots, n\}$ is an index set, $\{I_1, \ldots, I_m\}$ are subsets of I such that $I_1 \cup \ldots \cup I_k = I$ and $I_1 \cap \ldots \cap I_k = \emptyset$, $\mathrm{S}_{\mathrm{equ}}$ and $\mathrm{S}_{\mathrm{neq}}$ are index sets such that $\mathrm{S}_{\mathrm{equ}} \cup \mathrm{S}_{\mathrm{neq}} = \{1, \ldots, \mathrm{m}\}$ and $\mathrm{S}_{\mathrm{equ}} \cap \mathrm{S}_{\mathrm{neq}} = \emptyset$, $(b_1, \ldots, b_m) \in \mathbb{R}^m$ are positive numbers.

The implemented solver (`libqp_splx.c`) is a generalization of the method proposed in [1, 2]. It is based on the Sequential Minimal Optimization (SMO) algorithm with an improved working set selection strategy. Solving instances of this QP task is required, for example, in machine learning methods like Structured SVM learning, Bundle Methods for Risk Minimization, binary SVM with L2-soft margin, etc.

QP task with box constraints and a single linear equality constraint. This QP task is defined as follows

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x} + \mathbf{x}^T\mathbf{f} \\ \text{subject to} \quad & \mathbf{x}^T\mathbf{a} = b\,, \\ & l_i \leq x_i \leq u_i\,, \qquad i = 1, \ldots, n \quad , \end{aligned}$$

where $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$ is the optimized vector, $\mathbf{H} \in \mathbb{R}^{n \times n}$ is a symmetric positive semi-definite matrix, $\mathbf{f} \in \mathbb{R}^n$ is a vector, $\mathbf{a} \in \mathbb{R}^n$ is a vector

with non-zero entries, $b \in \mathbb{R}$ is a scalar, $(l_1, \ldots, l_n) \in (\mathbb{R} \cup \{-\infty\})^n$ and $(u_1, \ldots, u_n) \in (\mathbb{R} \cup \{\infty\})^n$ are lower and upper bounds, respectively.

The solver (`libqp_gsmo.c`) is the exact implementation of the Generalized Sequential Minimal Optimizer proposed in [3]. Solving this QP task is required, for example, when training binary SVM with L1-soft margin.

Interfaces

LIBQP is implemented in C language and interfaces to Matlab.

Platforms

GNU/Linux. It should run also under Windows though not tested.

Installation

LIBQP can be downloaded from `http://cmp.felk.cvut.cz/~xfrancv/libqp/libqp.zip`.

MATLAB.

(1) Run Matlab and go to the folder `libqp_root/matlab`

```
cd libqp_root/matlab
```

(2) Compile mex files by running

```
libqp_compile
```

Now you can use `libqp_splx` and `libqp_gsmo` solvers located in `libqp_root/matlab`. To make these function visible from Matlab you need to add

```
addpath('libqp_root/matlab')
```

to your `startup.m` file.
To test the solvers run scripts

```
libqp_splx_test
libqp_gsmo_test
```

Example application.

(1) Go to the folder `libqp_root/examples`

```
cd libqp_root/examples
```

(2) Issue make

```
make
```

Now you can run test script

```
./run_test
```

License

LIBQP is licensed under the GPL version 3 (`http://gplv3.fsf.org/`).

References

[1] V. Franc, V. Hlavac. A Novel Algorithm for Learning Support Vector Machines with Structured Output Spaces. Research Report K333 22/06, CTU-CMP-2006-04. May, 2006. ftp://cmp.felk.cvut.cz/pub/cmp/articles/franc/Franc-TR-2006-04.ps

[2] R.-E. Fan, P.-H. Chen, C.-J. Lin. Working Set Selection Using Second Order Information for Training SVM. JMLR. vol 6. 2005. TBA

[3] S.-S. Keerthi, E.G.Gilbert. Convergence of a Generalized SMO Algorithm for SVM Classifier Design. Technical Report CD-00-01, Control Division, Dept. of Mechanical and Production Engineering, National University of Singapore, 2000. http://citeseer.ist.psu.edu/keerthi00convergence.html