Pattern Information Processing: Sparse Methods

Masashi Sugiyama

(Department of Computer Science)

Contact: W8E-505

sugi@cs.titech.ac.jp

http://sugiyama-www.cs.titech.ac.jp/~sugi/

Sparseness and Continuous Model Choice

- Two approaches for avoiding over-fitting:

	Sparseness	Model parameter
Subset LS	Yes	Combinatorial
Quadratically constrained LS	No	Continuous

- We want to have sparseness and continuous model choice at the same time.

Today’s Plan

- Sparse learning method
- How to deal with absolute values in optimization
- Approximate gradient descent
- Standard form of quadratic programs

Non-Linear Learning for Linear / Kernel Models

- Linear / kernel models

$$f_{\boldsymbol{\alpha}}(\boldsymbol{x}) = \sum_{i=1}^{b} \alpha_i \varphi_i(\boldsymbol{x}) \qquad f_{\boldsymbol{\alpha}}(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i K(\boldsymbol{x}, \boldsymbol{x}_i)$$

- Non-linear learning

$$\hat{\boldsymbol{\alpha}} = \boldsymbol{L}(\boldsymbol{y})$$

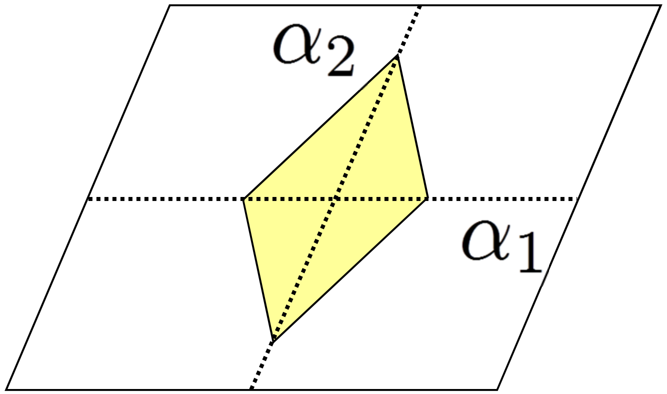
$\boldsymbol{L}(\cdot)$:Non-linear function

l1-Constrained LS

- Restrict the search space within an ℓ_1-ball.

$$\hat{\boldsymbol{\alpha}}_{\ell_1 CLS} = \mathop{\mathrm{argmin}}_{\boldsymbol{\alpha} \in \mathbb{R}^b} J_{LS}(\boldsymbol{\alpha})$$

$$\text{subject to } \|\boldsymbol{\alpha}\|_1 \leq C$$



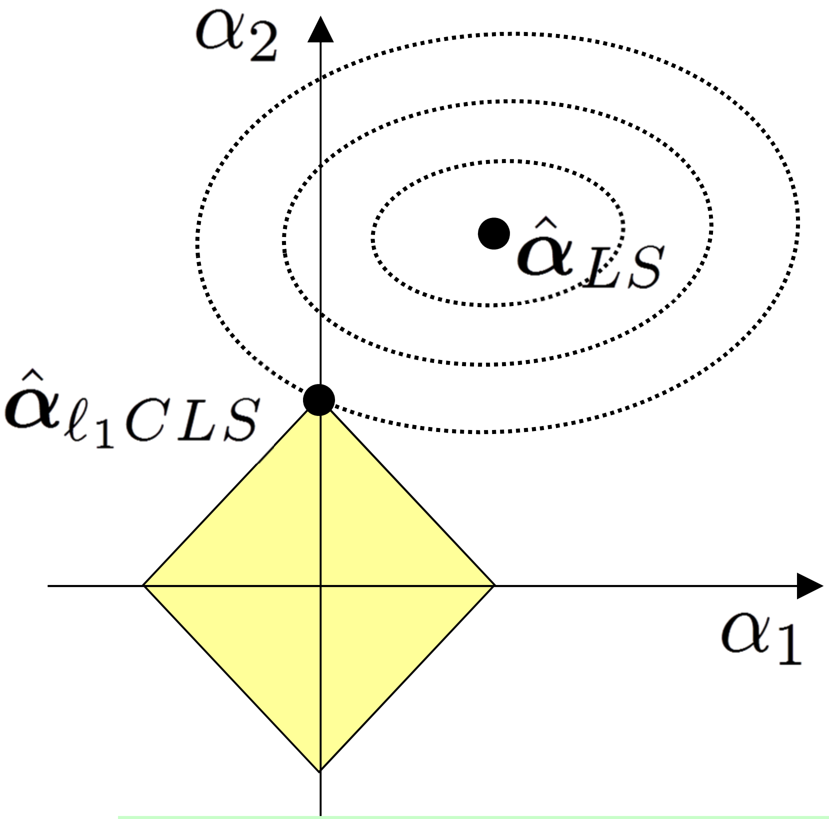
$$\ell_1 - \text{norm} \quad \|\boldsymbol{\alpha}\|_1 = \sum_{i=1}^{b} |\alpha_i|$$

$$J_{LS}(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \left(f_{\boldsymbol{\alpha}}(\boldsymbol{x}_i) - y_i\right)^2$$

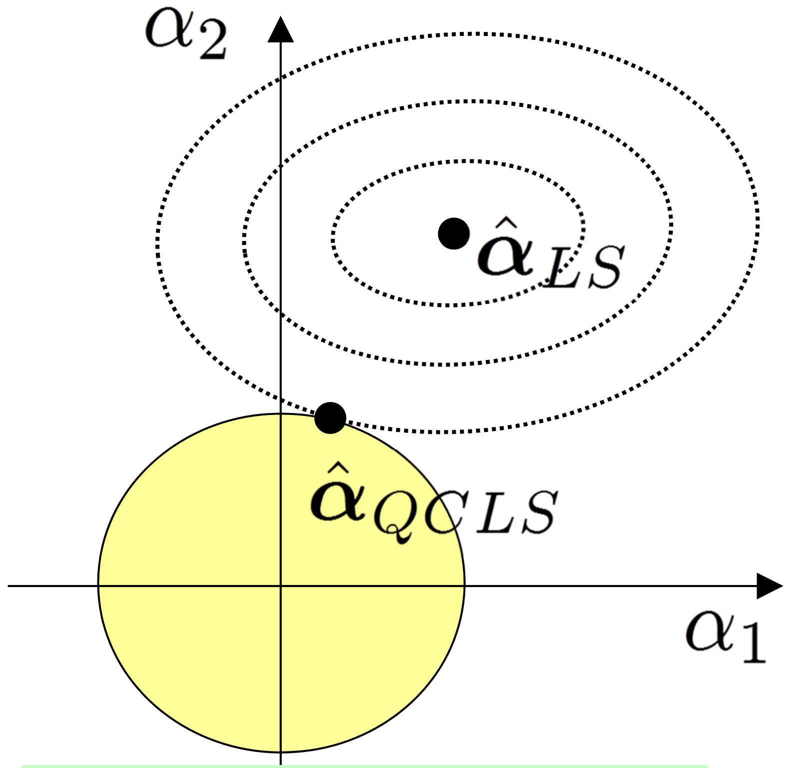
Tibshirani, Regression shrinkage and selection via the lasso,
Journal of the Royal Statistical Society, Series B, 58(1), 267-288,1996.

Why Sparse?

- The solution is often exactly on an axis.



ℓ_1 constrained LS



Quadratically constrained LS

How to Obtain A Solution

- Lagrangian:

$$J_{\ell_1 CLS}(\boldsymbol{\alpha}) = J_{LS}(\boldsymbol{\alpha}) + \lambda(\|\boldsymbol{\alpha}\|_1 - C)$$

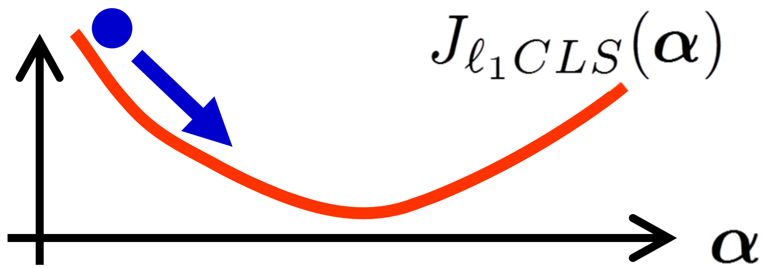
- λ :Lagrange multiplier
- Similarly to QCLS, we practically start from $\lambda\ (\geq 0)$ and solve

$$\hat{\boldsymbol{\alpha}}_{\ell_1 CLS} = \operatorname*{argmin}_{\boldsymbol{\alpha} \in \mathbb{R}^b} J_{\ell_1 CLS}(\boldsymbol{\alpha})$$

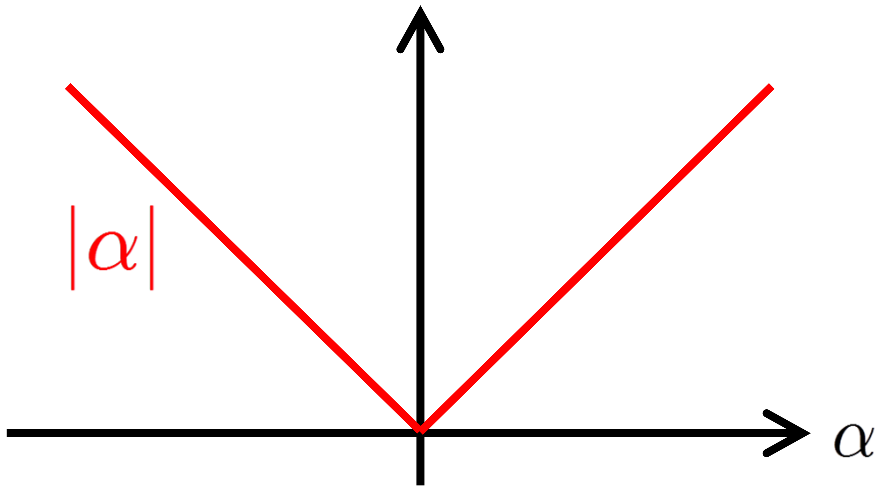
- It is often called ℓ_1-regularized LS.

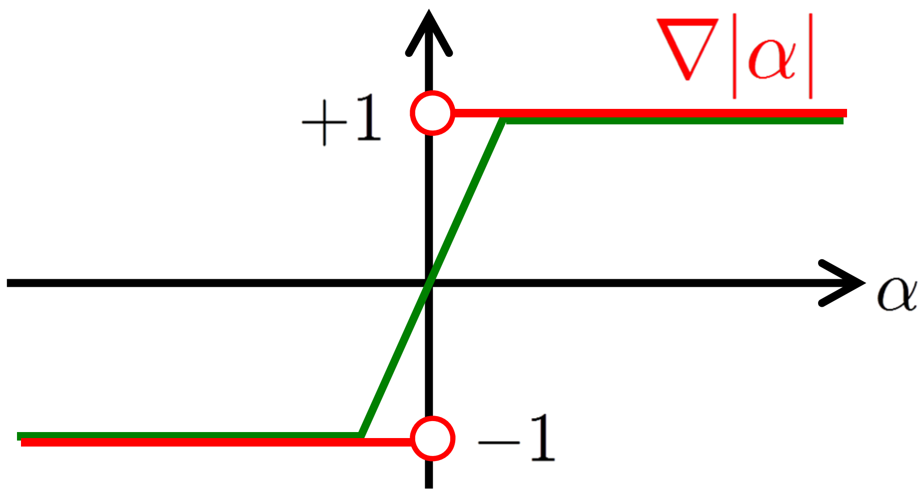
Gradient Descent

- $\boldsymbol{\alpha} \longleftarrow \boldsymbol{\alpha} - \epsilon \nabla J_{\ell_1 CLS}(\boldsymbol{\alpha})$



- However, ℓ_1-norm is not differentiable.
  - Use smooth approximation!





  - You may also use a quasi-Newton method.

Quadratic Program

- Use the following expression:

$$|\alpha| = \min_{u \in \mathbb{R}} u \quad \text{subject to} \ -u \leq \alpha \leq u$$

- Proof by contradiction:
  - Let $\widehat{u} = \operatorname*{argmin}_{u \in \mathbb{R}} u \quad \text{subject to} \ -u \leq \alpha \leq u$.
  - The constraint implies $\widehat{u} \geq |\alpha|$.
  - Assume $\widehat{u} > |\alpha|$.
  - Then such $\widehat{u}$ is not a solution because $\widetilde{u} = |\alpha|$ gives a smaller value.
  - This implies that the solution should satisfy $\widehat{u} = |\alpha|$.

How to Obtain A Solution (cont.)

$$\hat{\boldsymbol{\alpha}}_{\ell_1 CLS} = \mathop{\mathrm{argmin}}_{\boldsymbol{\alpha} \in \mathbb{R}^b} J_{LS}(\boldsymbol{\alpha}) + \lambda \|\boldsymbol{\alpha}\|_1$$

- $\hat{\boldsymbol{\alpha}}_{\ell_1 CLS}$ is given as the solution of

$$\min_{\boldsymbol{\alpha}, \boldsymbol{u} \in \mathbb{R}^b} \left[J_{LS}(\boldsymbol{\alpha}) + \lambda \sum_{i=1}^{b} u_i \right]$$

$$\text{subject to } -\boldsymbol{u} \leq \boldsymbol{\alpha} \leq \boldsymbol{u},$$

Note: Inequality for vectors is component-wise

$$J_{LS}(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \left(f_{\boldsymbol{\alpha}}(\boldsymbol{x}_i) - y_i \right)^2 = \|\boldsymbol{X}\boldsymbol{\alpha} - \boldsymbol{y}\|^2$$

Linearly-Constrained Quadratic Program (QP)

- Standard optimization software can solve QP:

$$\min_{\boldsymbol{\beta}} \left[\frac{1}{2} \langle \boldsymbol{Q}\boldsymbol{\beta}, \boldsymbol{\beta} \rangle + \langle \boldsymbol{\beta}, \boldsymbol{q} \rangle \right]$$

$$\text{subject to } \boldsymbol{H}\boldsymbol{\beta} \leq \boldsymbol{h}$$

$$\boldsymbol{G}\boldsymbol{\beta} = \boldsymbol{g}$$

Transformation into Standard Form

- Let

$$\boldsymbol{\beta} = \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{u} \end{pmatrix} \qquad \begin{aligned} \boldsymbol{\Gamma}_{\boldsymbol{\alpha}} &= (\boldsymbol{I}_b, \boldsymbol{O}_b) \\ \boldsymbol{\Gamma}_{\boldsymbol{u}} &= (\boldsymbol{O}_b, \boldsymbol{I}_b) \end{aligned}$$

- Then

$$\begin{aligned} \boldsymbol{\alpha} &= \boldsymbol{\Gamma}_{\boldsymbol{\alpha}}\boldsymbol{\beta} \\ \boldsymbol{u} &= \boldsymbol{\Gamma}_{\boldsymbol{u}}\boldsymbol{\beta} \end{aligned}$$

- Use these expressions and replace all $\boldsymbol{\alpha}, \boldsymbol{u}$ with $\boldsymbol{\beta}$.

Standard Form

$$\min_{\boldsymbol{\beta}} \left[\frac{1}{2} \langle \boldsymbol{Q}\boldsymbol{\beta}, \boldsymbol{\beta} \rangle + \langle \boldsymbol{\beta}, \boldsymbol{q} \rangle \right] \quad \text{subject to } \boldsymbol{H}\boldsymbol{\beta} \leq \boldsymbol{h}, \quad \boldsymbol{G}\boldsymbol{\beta} = \boldsymbol{g}$$

- ℓ_1-constrained LS can be expressed as

$$\begin{aligned}
\boldsymbol{Q} &= 2\boldsymbol{\Gamma}_{\boldsymbol{\alpha}}^\top \boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{\Gamma}_{\boldsymbol{\alpha}} \\
\boldsymbol{q} &= -2\boldsymbol{\Gamma}_{\boldsymbol{\alpha}}^\top \boldsymbol{X}^\top \boldsymbol{y} + \lambda \boldsymbol{\Gamma}_{\boldsymbol{u}}^\top \boldsymbol{1}_b \\
\boldsymbol{H} &= \begin{pmatrix} -\boldsymbol{\Gamma}_{\boldsymbol{\alpha}} - \boldsymbol{\Gamma}_{\boldsymbol{u}} \\ \boldsymbol{\Gamma}_{\boldsymbol{\alpha}} - \boldsymbol{\Gamma}_{\boldsymbol{u}} \end{pmatrix} \\
\boldsymbol{h} &= \boldsymbol{0}_{2b} \\
\boldsymbol{G} &= \boldsymbol{O}_{2b} \\
\boldsymbol{g} &= \boldsymbol{0}_{2b}
\end{aligned}$$

$$\boldsymbol{\beta} = \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{u} \end{pmatrix}$$

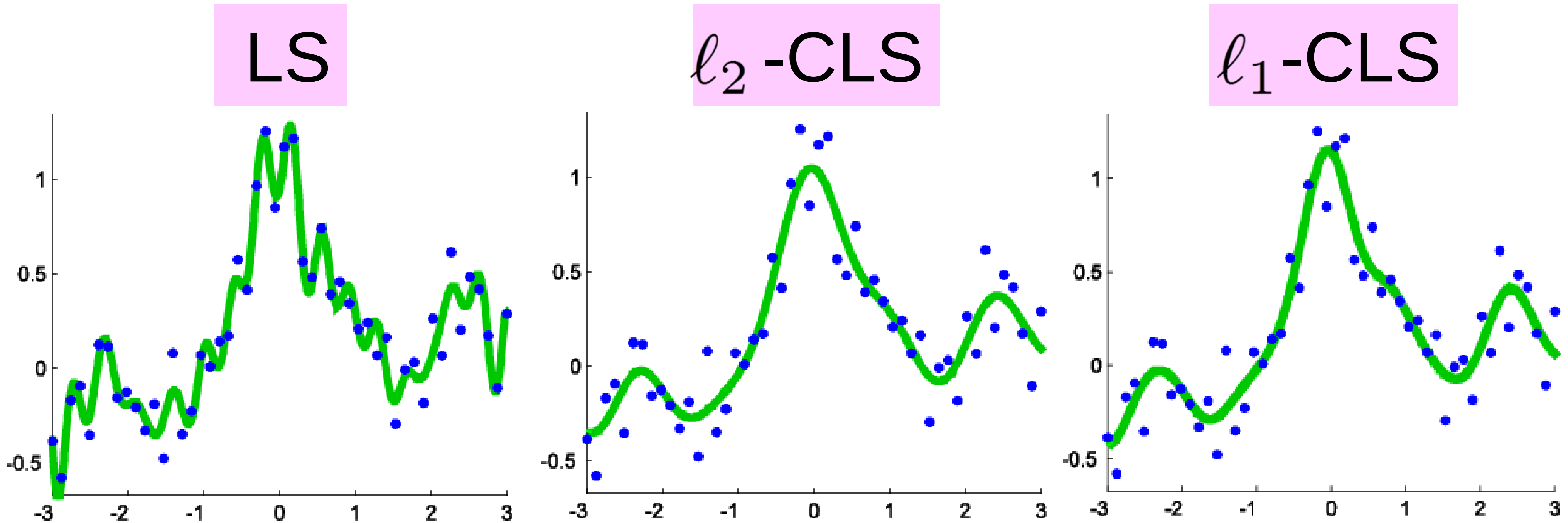
$$\begin{aligned}
\boldsymbol{\Gamma}_{\boldsymbol{\alpha}} &= (\boldsymbol{I}_b, \boldsymbol{O}_b) \\
\boldsymbol{\Gamma}_{\boldsymbol{u}} &= (\boldsymbol{O}_b, \boldsymbol{I}_b)
\end{aligned}$$

Proof: Homework!

Example of Sparse Learning

- Gaussian kernel model:

$$f_{\boldsymbol{\alpha}}(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}_i\|^2}{2}\right)$$



- ℓ_2-CLS and ℓ_1-CLS give similar results.
- 37 out of 50 parameters are exactly zero in ℓ_1.

Feature Selection

- If ℓ_1-CLS is combined with linear model with respect to input,

$$f_{\boldsymbol{\alpha}}(\boldsymbol{x}) = \boldsymbol{\alpha}^\top \boldsymbol{x} \qquad \boldsymbol{x} = (x^{(1)}, x^{(2)}, \ldots, x^{(d)})^\top$$

  some input variables are not used for prediction.

  Important features are automatically selected

- Example: Gene selection
- Generally, 2^d combinations need to be compared for feature selection (cf. subset LS).
- On the other hand, ℓ_1-CLS only involves a continuous model parameter λ.

Constrained LS

	Sparseness	Model parameter	Parameter learning
Subset LS	Yes	Combina-torial	Analytic (Linear)
Quadratically constrained LS	No	Continuous	Analytic (Linear)
ℓ_1 constrained LS	Yes	Continuous	Iterative (Non-linear)

Notification of Final Assignment

1. Apply supervised learning techniques to your data set and analyze it.

- Final report deadline: Aug 1st (Fri.) 17:00
- Bring your report to W8E-404.

Mini-Workshop on Data Mining

- On July 15th and 22nd, we will have a mini-workshop on data mining.
- Several students present their own data mining results.
- Those who give a talk at the workshop will have very good grades!

Mini-Workshop on Data Mining

- Application (just to declare that you want to give a presentation) deadline: July 1st.
- Presentation: 10-15 minutes (?).
  - Specification of your dataset
  - Methods used
  - Outcome
- Slides should be in English.
- Better to speak in English, but Japanese is also allowed.

Homework

1. Derive the standard quadratic programming form of ℓ_1 -constrained LS.

$$\min_{\boldsymbol{\beta}} \left[\frac{1}{2} \langle \boldsymbol{Q}\boldsymbol{\beta}, \boldsymbol{\beta} \rangle + \langle \boldsymbol{\beta}, \boldsymbol{q} \rangle \right]$$

$$\text{subject to } \boldsymbol{H}\boldsymbol{\beta} \leq \boldsymbol{h}$$

$$\boldsymbol{G}\boldsymbol{\beta} = \boldsymbol{g}$$

$$\boldsymbol{\beta} = \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{u} \end{pmatrix}$$

$$\begin{aligned} \boldsymbol{\Gamma}_{\boldsymbol{\alpha}} &= (\boldsymbol{I}_b, \boldsymbol{O}_b) \\ \boldsymbol{\Gamma}_{\boldsymbol{u}} &= (\boldsymbol{O}_b, \boldsymbol{I}_b) \end{aligned}$$

$$\begin{aligned} \boldsymbol{Q} &= 2\boldsymbol{\Gamma}_{\boldsymbol{\alpha}}^\top \boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{\Gamma}_{\boldsymbol{\alpha}} \\ \boldsymbol{q} &= -2\boldsymbol{\Gamma}_{\boldsymbol{\alpha}}^\top \boldsymbol{X}^\top \boldsymbol{y} + \lambda \boldsymbol{\Gamma}_{\boldsymbol{u}}^\top \mathbf{1}_b \\ \boldsymbol{H} &= \begin{pmatrix} -\boldsymbol{\Gamma}_{\boldsymbol{\alpha}} - \boldsymbol{\Gamma}_{\boldsymbol{u}} \\ \boldsymbol{\Gamma}_{\boldsymbol{\alpha}} - \boldsymbol{\Gamma}_{\boldsymbol{u}} \end{pmatrix} \\ \boldsymbol{h} &= \mathbf{0}_{2b} \\ \boldsymbol{G} &= \boldsymbol{O}_{2b} \\ \boldsymbol{g} &= \mathbf{0}_{2b} \end{aligned}$$

Homework (cont.)

2. For your own toy 1-dimensional data, perform simulations using
   - Gaussian kernel models
   - ℓ_1-constraint least-squares learning

   and analyze the results, e.g., by changing
   - Target functions
   - Number of samples
   - Noise level

   Use 5-fold cross-validation for choosing
   - Width of Gaussian kernel
   - Regularization parameter

   Compare the results of QCLS and ℓ_1CLS, e.g., in terms of sparseness and accuracy.

Solving QP Problems

- R: “quadprog”
- Octave: “qp”
- MATLAB: “quadprog”
  (you need Optimization Toolbox)
  - Various free software seems available, for example, “quadprog2”. http://www.mathworks.com.au/matlabcentral/fileexchange/7860-quadprog2-convex-qp-solver