Master's Thesis

An accelerated proximal gradient method for Fenchel-type dual problems of general support vector regressions

Guidance

Professor Nobuo YAMASHITA

Toshiaki HAGA

Department of Applied Mathematics and Physics

Graduate School of Informatics

Kyoto University

February 2015

Abstract

A regression is one of the major machine learning techniques, and it is used in various fields. A Gaussian process regression (GPR) and a support vector regression (SVR) are standard regression methods. A regression function given by GPR usually has the better generalization ability than those by SVR and other regressions. On the other hand, the regression function of SVR is represented by a few data points, while GPR usually uses all data. Therefore SVR is more suitable than GPR when we need quick evaluation of the regression function with low memory.

In this paper, we propose a new general SVR that has the above advantages of both GPR and SVR. The original SVR and GPR are special cases of the proposed general SVR. We then consider its Fenchel-type problem in order to apply the kernel trick to the proposed general SVR. Moreover, we adapt the accelerated proximal gradient method to solve the Fenchel-type dual problem. We present concrete implementation for applying the method effectively, which exploits the special structure of the general SVR. We give some numerical results for the general SVR. These results show that proposed regression has the advantages of both GPR and SVR.

Contents

1 Introduction

With the developments in information technology, it is necessary to use big data effectively in various fields. Then, machine learning techniques based on statistics and optimization theory has been drawn much attention. Especially, a considerable number of studies on a regression has been conducted since it is highly useful and widely applied [10, 14, 16].

The purpose of the regression is to predict a function y that expresses a relation between input and target by using given datasets $\{(\boldsymbol{x}_i, t_i)\}$, where $\boldsymbol{x}_i \in \mathbb{R}^n$ $(i = 1, \ldots, N)$ and $t_i \in \mathbb{R}$ $(i = 1, \ldots, N)$ are given input and target dataset, respectively. As we get such a regression function y, we can predict target $\hat{t}$ for a new input data $\hat{\boldsymbol{x}} \in \mathbb{R}^n$ as $\hat{t} = y(\hat{\boldsymbol{x}})$.

There are several approaches in regressions. Examples of regression methods are a least-squares regression, a Ridge regression, a Gaussian process regression (GPR) and a support vector regression (SVR) [2]. Most of these regression methods get a regression function y as an optimal solution of the following minimization problem:

$$\min_{y} \quad \sum_{i=1}^{N} F_l\left(y(\boldsymbol{x}_i) - t_i\right) + \frac{1}{C} F_r(y), \tag{1.1}$$

where $F_l : \mathbb{R} \to \mathbb{R}$ is a loss function that attains its minimum value at 0, C is a regularization parameter, F_r is a regularization function. The loss function F_l is a fundamental of a regression method. Moreover, the properties of problem (1.1) and its solution y^* mainly depend on the function. If y^* is a minimizer of only a loss function, it sometimes fits too closely to dataset $(\boldsymbol{x}_i,\, t_i)$ $(i = 1, \ldots, N)$, which is called over fitting. The regularization function F_r is a functional with respect to y for avoiding over-fitting. However, it is difficult to minimize problem (1.1) with respect to y itself. Therefore, we suppose that $y(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x})$, where $\boldsymbol{w} \in \mathbb{R}^M$ is a weight vector and $\boldsymbol{\phi} = [\phi_1, \ldots, \phi_M]^\top$ is a basis function vector whose elements are nonlinear functions $\phi_i : \mathbb{R}^n \to \mathbb{R}$ $(i = 1, \ldots, M)$. Then, problem (1.1) is reduced to the following problem with respect to the weight vector $\boldsymbol{w}$:

$$\min_{\boldsymbol{w}} \quad \sum_{i=1}^{N} F_l\left(\boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x}_i) - t_i\right) + \frac{1}{2C}\|\boldsymbol{w}\|^2. \tag{1.2}$$

In this paper, we propose a general support vector regression that includes SVR and GPR. SVR is closely related to a support vector machine (SVM), which was proposed by Vapnik, and it is a high accuracy method in a classification. SVR adopts ε-insensitive loss function as the loss function F_l in problem (1.1). Since the loss function ignores an error of a certain range from the true value, most of $F_l\left(\boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x}_i) - t_i\right) = 0$ $(i = 1, \ldots, N)$. The decision variables of a dual problem of SVR corresponds to data points in the primal problem. Then a solution of the dual problem is usually sparse. By exploiting the sparsity, we can solve the problem fast by using a method such as a conjugate gradient method or a sequential minimal optimization (SMO) [3, 12].

GPR is given by considering a noise in the target value. Models equivalent to GPR have been widely studied in many different fields. For instance, it is known as kriging in the geostatistics literature [5]. Williams and Rasmussen adopted GPR for a machine learning [17]. Nguen-Tuong, Seeger and Peters showed that a regression function of GPR usually has the better generalization ability than those by

SVR and other regressions [11]. As seen in Section 3, the regression function by GPR is a solution of a dual problem of the Ridge regression. The loss function of a Ridge regression is $F_l\left(\boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x}_i) - t_i\right) = \|\boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x}_i) - t_i\|^2$ $(i = 1, \ldots, N)$. The problem (1.1) for the Ridge regression or its dual problem is unconstrained convex quadratic problem, and hence it is equivalent to N linear equations.

The loss function of a Ridge regression is $F_l\left(y(\boldsymbol{x}_i) - t_i\right) = \|y - t_i\|^2$ $(i = 1, \ldots, N)$, and problem (1.1) is reduced to N linear equations. Therefore, for large training datasets, the direct application of GPR is difficult, and serval approximation methods have been proposed [8, 18].

The purpose of this paper is the following two points. First, we propose a general support vector regression that includes SVR and GPR. The regression has the advantages of both SVR and GPR. Second, we construct a concrete implementation for applying an accelerated proximal gradient method to solve the dual problem of the proposed regression.

In this paper, we propose a loss function F_l such that it concludes features of various regression methods. Then, we consider a Fenchel-type dual problem [13] of problem (1.2) in order to apply the kernel trick to the proposed general SVR.

Moreover, we adapt the accelerated proximal gradient method to solve the Fenchel-type dual problem. The proximal gradient method that combines a proximal point method and a gradient method. When an objective function can be separated into a differentiable function and a function with a special structure, we apply the proximal point method to the special structured function and apply the gradient method to the differentiable one. Beck and Teboulle proposed an accelerated proximal gradient algorithm which has a convergence rate $O(1/k^2)$ [1]. We present concrete implementation of this algorithm for the Fenchel-type dual problem, which exploits the special structure of the general SVR. We conduct numerical experiments by this algorithm, and investigate the validity of the general SVR.

This paper is organized as follows. In Section 2, we introduce kernel functions, the existing regressions, a Fenchel-type dual problem, and an accelerated proximal gradient method. In Section 3, we generalize a regression problem by exploiting a Fenchel-type dual problem. In Section 4, we propose a general SVR which can be solved by the accelerated proximal gradient method. In Section 5, we give some numerical results to investigate the validity of the proposed regression. In Section 6, we conclude the paper with some remarks.

2 Preliminaries

In this section, we introduce kernel functions, the existing regressions (SVR and GPR), a Fenchel-type dual problem, and an accelerated proximal gradient method.

2.1 Regression functions and Kernel functions

First, we introduce a linear model for regression function [2, Chapter 3]. The model is given by a linear combination of an input variable $\boldsymbol{x} \in \mathbb{R}^n$ as follows :

$$\hat{y}^0(\boldsymbol{x}; \hat{\boldsymbol{w}}) = \hat{\boldsymbol{w}}^\top \boldsymbol{x}, \tag{2.1}$$

where $\hat{\boldsymbol{w}} \in \mathbb{R}^n$ is a weight vector. We may also consider the following model that has a bias parameter b:

$$\hat{y}^b(\boldsymbol{x}; \hat{\boldsymbol{w}}, b) = \hat{\boldsymbol{w}}^\top \boldsymbol{x} + b. \tag{2.2}$$

Models (2.1) and (2.2) are expressed as a linear function of the input variable, and this fact imposes significant limitations on these regressions. Therefore, by exploiting basis function vector $\boldsymbol{\phi} = [\phi_1, \ldots, \phi_M]^\top$, we extend models (2.1) and (2.2) as follows:

$$\begin{aligned} y^0(\boldsymbol{x}; \boldsymbol{w}) &= \boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x}), & (2.3) \\ y^b(\boldsymbol{x}; \boldsymbol{w}, b) &= \boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x}) + b, & (2.4) \end{aligned}$$

where $\boldsymbol{w} \in \mathbb{R}^M$ is a weight vector and $\phi_i : \mathbb{R}^n \to \mathbb{R}$ $(i = 1, \ldots, M)$ are nonlinear functions.

Next, we describe kernel functions. We define the following kernel function $k(\boldsymbol{x}, \boldsymbol{x}')$ with respect to two different input variables $\boldsymbol{x}$ and $\boldsymbol{x}'$ by using a basis function vector $\boldsymbol{\phi}$:

$$k(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{\phi}(\boldsymbol{x}').$$

The well-known kernel function are the following polynomial kernel function and Gaussian kernel function:

$$\begin{aligned} &\text{(polynomial kernel function)} && k(\boldsymbol{x}, \boldsymbol{x}') = (\boldsymbol{x}^\top \boldsymbol{x}' + c)^m, \\ &\text{(Gaussian kernel function)} && k(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{2\sigma^2}\right). \end{aligned} \tag{2.5}$$

Note that $n = \inf$ for Gaussian kernel function.

We also define the following Gram matrix $\boldsymbol{K}$ with respect to input variables $\boldsymbol{x}_i$ $(i = 1, \ldots, N)$ by using kernel functions:

$$\boldsymbol{K} = \begin{pmatrix} k(\boldsymbol{x}_1, \boldsymbol{x}_1) & k(\boldsymbol{x}_1, \boldsymbol{x}_2) & \ldots & k(\boldsymbol{x}_1, \boldsymbol{x}_N) \\ k(\boldsymbol{x}_2, \boldsymbol{x}_1) & k(\boldsymbol{x}_2, \boldsymbol{x}_2) & \ldots & k(\boldsymbol{x}_2, \boldsymbol{x}_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(\boldsymbol{x}_N, \boldsymbol{x}_1) & k(\boldsymbol{x}_N, \boldsymbol{x}_2) & \ldots & k(\boldsymbol{x}_N, \boldsymbol{x}_N) \end{pmatrix}. \tag{2.6}$$

Note that the Gram matrix $\boldsymbol{K}$ is positive semidefinite matrix.

2.2 Existing regressions

In this subsection, we introduce a support vector regression (SVR) and a Gaussian process regression (GPR). See [6, Chapter 6] for details of SVR and [2, Chapter 6] for those of GPR. Let $\boldsymbol{x}_i \in \mathbb{R}^n$ $(i = 1 \ldots N)$ and $t_i \in \mathbb{R}$ $(i = 1 \ldots N)$ be given input and target datasets, respectively. Moreover, we define $\boldsymbol{t} = [t_1, \ldots, t_N]^\top$.

2.2.1 A Support vector regression with linear ε-insensitive loss function

We introduce SVR with the linear ε-insensitive loss function defined by

$$F_l\left(\boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x}) + b - t\right) = L_1^\varepsilon(\boldsymbol{x}, t; \boldsymbol{w}, b) = \max\left\{0, \left|\boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x}) + b - t\right| - \varepsilon\right\}.$$

We adopt the model (2.4), that is, $y^b(\boldsymbol{x}; \boldsymbol{w}) = \boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x}) + b$ to SVR (linear ε-insensitive loss function). We consider the following unconstrained minimization problem with the regularization $\frac{1}{2C}\|\boldsymbol{w}\|^2$:

$$\min \quad \sum_{i=1}^N L_1^\varepsilon(\boldsymbol{x}_i, t_i; \boldsymbol{w}, b) + \frac{1}{2C}\|\boldsymbol{w}\|^2, \tag{2.7}$$

where C is a regularization parameter. The problem (2.7) is reduced to the following problem by using slack variables $\xi_i \geq 0$ and $\hat{\xi}_i \geq 0$:

$$\begin{aligned} \min \quad & \sum_{i=1}^N (\xi_i + \hat{\xi}_i) + \frac{1}{2C}\|\boldsymbol{w}\|^2 \\ \text{s.t.} \quad & t_i \leq \boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x}_i) + b + \varepsilon + \xi_i \quad (i = 1, \ldots, N), \\ & t_i \geq \boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x}_i) + b - \varepsilon - \xi_i \quad (i = 1, \ldots, N), \\ & \xi_i, \hat{\xi}_i \geq 0 \quad (i = 1, \ldots, N). \end{aligned} \tag{2.8}$$

A Lagrange dual problem of (2.8) is expressed as

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \tfrac{1}{2}\boldsymbol{\alpha}^\top \boldsymbol{K}\boldsymbol{\alpha} + \varepsilon\|\boldsymbol{\alpha}\|_1 - \boldsymbol{t}^\top \boldsymbol{\alpha} \\ \text{s.t.} \quad & \sum_{i=1}^N \alpha_i = 0, \\ & -C \leq \alpha_i \leq C \quad (i = 1, \ldots, N), \end{aligned} \tag{2.9}$$

where $\boldsymbol{K}$ is the Gram matrix defined by (2.6). Let $\boldsymbol{\alpha}^* \in \mathbb{R}^N$ be an optimal solution of problem (2.9). Then, we set $b^* = t_j - \epsilon - \sum_{i=1}^N \alpha_i^* k(\boldsymbol{x}_j, \boldsymbol{x}_i)$ when j is an index such that $-C < \alpha_j^* < 0$. Then, we obtain the following regression function by using $\boldsymbol{\alpha}^*$ and b^*:

$$y^b(\boldsymbol{x}) = \sum_{i=1}^N \alpha_i^* k(\boldsymbol{x}_i, \boldsymbol{x}) + b^*. \tag{2.10}$$

Note that most of α_i^* are 0, that is , the solution $\boldsymbol{\alpha}^*$ is sparse. We can evaluate the function (2.10) by calculating only $k(\boldsymbol{x}_i, \boldsymbol{x})$ for i such that $\alpha_i \neq 0$.

2.2.2 A Support vector regression with quadratic ε-insensitive loss function

We introduce SVR with the quadratic ε-insensitive loss function defined by

$$F_l(\boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x}) + b - t) = L_2^\varepsilon(\boldsymbol{x}, t; \boldsymbol{w}, b) = \max\left\{0, \left|\boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x}) + b - t\right| - \varepsilon\right\}^2.$$

We adopt the model (2.4), that is, $y^b(\boldsymbol{x};\boldsymbol{w}) = \boldsymbol{w}^\top\boldsymbol{\phi}(\boldsymbol{x}) + b$ to SVR (quadratic ε-insensitive loss function). We consider the following unconstrained minimization problem with the regularization $\frac{1}{2C}\|\boldsymbol{w}\|^2$:

$$\min \quad \sum_{i=1}^{N} L_2^\varepsilon(\boldsymbol{x}_i, t_i; \boldsymbol{w}, b) + \frac{1}{2C}\|\boldsymbol{w}\|^2, \tag{2.11}$$

where C is a regularization parameter. Problem (2.11) is reduced to the following problem by using slack variables $\xi_i \geq 0$ and $\hat{\xi}_i \geq 0$:

$$\begin{array}{ll} \min & C\sum_{i=1}^{N}(\xi_i^2 + \hat{\xi}_i^2) + \frac{1}{2}\|\boldsymbol{w}\|^2 \\ \text{s.t.} & t_i \leq \boldsymbol{w}^\top\boldsymbol{\phi}(\boldsymbol{x}_i) + b + \varepsilon + \xi_i \quad (i = 1, \ldots, N), \\ & t_i \geq \boldsymbol{w}^\top\boldsymbol{\phi}(\boldsymbol{x}_i) + b - \varepsilon - \xi_i \quad (i = 1, \ldots, N), \\ & \xi_i, \hat{\xi}_i \geq 0 \quad (i = 1, \ldots, N). \end{array} \tag{2.12}$$

A Lagrange dual problem of (2.12) is expressed as

$$\begin{array}{ll} \min\limits_{\boldsymbol{\alpha}} & \frac{1}{2}\boldsymbol{\alpha}^\top\left(\boldsymbol{K} + \frac{1}{2C}\boldsymbol{I}\right)\boldsymbol{\alpha} + \varepsilon\|\boldsymbol{\alpha}\|_1 - \boldsymbol{t}^\top\boldsymbol{\alpha} \\ \text{s.t.} & \sum_{i=1}^{N}\alpha_i = 0. \end{array} \tag{2.13}$$

where $\boldsymbol{K}$ is the Gram matrix defined in (2.6). Let $\boldsymbol{\alpha}^* \in \mathbb{R}^N$ be an optimal solution of problem (2.13), we have the regression function as (2.10) by using $\boldsymbol{\alpha}^* \in \mathbb{R}^N$.

2.2.3 A Gaussian process regression

We introduce a Gaussian process regression (GPR). We adopt the model (2.3), that is, $y^0(\boldsymbol{x};\boldsymbol{w}) = \boldsymbol{w}^\top\boldsymbol{\phi}(\boldsymbol{x})$. The prediction by GPR is expressed as a distribution of target data t. The distribution $p(t|\boldsymbol{t})$ by GPR is a Gaussian distribution that has the following mean function $m : \mathbb{R}^n \to \mathbb{R}$ and covariance function $\sigma^2 : \mathbb{R}^n \to \mathbb{R}$ [2, Chapter 6]:

$$m(\boldsymbol{x}) = \boldsymbol{t}^\top\left(\beta_1^{-1}\boldsymbol{K} + \beta_2^{-1}\boldsymbol{I}\right)\hat{\boldsymbol{k}}(\boldsymbol{x}), \tag{2.14}$$

$$\sigma^2(\boldsymbol{x}) = \hat{k}(\boldsymbol{x},\boldsymbol{x}) + \beta_2^{-1} - \hat{\boldsymbol{k}}(\boldsymbol{x})^\top\left(\beta_1^{-1}\boldsymbol{K} + \beta_2^{-1}\boldsymbol{I}\right)\hat{\boldsymbol{k}}(\boldsymbol{x}), \tag{2.15}$$

where $\hat{\boldsymbol{k}}$ is a vector whose elements are $\hat{k}(\boldsymbol{x}_i,\boldsymbol{x}) = \frac{1}{\beta_1}k(\boldsymbol{x}_i,\boldsymbol{x})$ $(i = 1, \ldots, N)$, β_1 and β_2 are hyperparameters of a prior distribution of $\boldsymbol{w}$ and noise in the target data, respectively. The mean of predictive distribution (2.14) corresponds to the regression function by the model (2.1). The covariance of the predictive distribution (2.15) shows the uncertainty of the regression function. When the covariance value is small, the regression has good generalization ability.

Now, let $\boldsymbol{\alpha}^* \in \mathbb{R}^N$ be defined by

$$\boldsymbol{\alpha}^* = \left(\beta_1^{-1}\boldsymbol{K} + \beta_2^{-1}\boldsymbol{I}\right)\boldsymbol{t}.$$

Then, the regression function y^0 by GPR is written as

$$y^0(\boldsymbol{x}) = m(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i^* \hat{k}(\boldsymbol{x}_i, \boldsymbol{x}). \tag{2.16}$$

Note that the function (2.16) is a regression function by using a kernel function as well as SVR. Moreover, $\boldsymbol{\alpha}^*$ is an optimal solution of the following problem:

$$\min_{\boldsymbol{\alpha}} \quad \frac{1}{2}\boldsymbol{\alpha}^\top \left(\frac{1}{\beta_1}\boldsymbol{K} + \frac{1}{\beta_2}\boldsymbol{I}\right)\boldsymbol{\alpha} - \boldsymbol{t}^\top \boldsymbol{\alpha}. \tag{2.17}$$

Therefore, we can get the regression function y^0 in GPR by solving problem (2.17).

2.3 A Fenchel-type duality

We describe a Fenchel-type duality. We consider the following primal problem:

$$(\mathrm{P_F}) \qquad \min_{\boldsymbol{x}} \quad f(\boldsymbol{x}) + g(\boldsymbol{A}\boldsymbol{x}), \tag{2.18}$$

where $f : \mathbb{R}^n \to (-\infty, +\infty]$ and $g : \mathbb{R}^m \to (-\infty, +\infty]$ are closed, proper, and convex functions, and A is an $m \times n$ matrix. To describe a dual problem of problem (2.18), we define the following function $F : \mathbb{R}^{n+m} \to (-\infty, +\infty]$:

$$F(\boldsymbol{x}, \boldsymbol{u}) = f(\boldsymbol{x}) + g(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{u}). \tag{2.19}$$

Clearly, F is a closed, proper, and convex function. Furthermore, we define a function $\psi : \mathbb{R}^m \to [-\infty, +\infty]$ as follows:

$$\psi(\boldsymbol{u}) = \inf\left\{F(\boldsymbol{x}, \boldsymbol{u}) | \boldsymbol{x} \in \mathbb{R}^n\right\}.$$

Then, $\inf(P) = \psi(\boldsymbol{0})$, where $\inf(P)$ is the infimum of problem (2.18). Let $L : \mathbb{R}^{n+m} \to [-\infty, +\infty]$ be a Lagrangian function defined by

$$\begin{aligned} L(\boldsymbol{x}, \boldsymbol{\lambda}) &= \inf\left\{F(\boldsymbol{x}, \boldsymbol{u}) + \langle \boldsymbol{\lambda}, \boldsymbol{u}\rangle | \, \boldsymbol{u} \in \mathbb{R}^m\right\} \\ &= \inf\left\{f(\boldsymbol{x}) + g(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{u}) + \langle \boldsymbol{\lambda}, \boldsymbol{u}\rangle | \, \boldsymbol{u} \in \mathbb{R}^m\right\} \\ &= f(\boldsymbol{x}) - g^*(-\boldsymbol{\lambda}) - \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x}\rangle, \end{aligned}$$

where $g^* : \mathbb{R}^m \to [-\infty, +\infty]$ is a conjugate function of g defined by

$$g^*(\boldsymbol{\xi}) = \sup\left\{\langle \boldsymbol{z}, \boldsymbol{\xi}\rangle - g(\boldsymbol{z}) | \boldsymbol{z} \in \mathbb{R}^m\right\}. \tag{2.20}$$

Therefore, an objective function of the Fenchel-type dual problem $\omega : \mathbb{R}^m \to [-\infty, +\infty)$ is a closed and concave function defined by

$$\begin{aligned} \omega(\boldsymbol{\lambda}) &= \inf\{L(\boldsymbol{x}, \boldsymbol{\lambda}) | \boldsymbol{x} \in \mathbb{R}^n\} \\ &= \inf\{f(\boldsymbol{x}) - g^*(-\boldsymbol{\lambda}) - \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x}\rangle | \boldsymbol{x} \in \mathbb{R}^n\} \\ &= -f^*(\boldsymbol{A}^\top \boldsymbol{\lambda}) - g^*(-\boldsymbol{\lambda}), \end{aligned} \tag{2.21}$$

where $f^* : \mathbb{R}^n \to [-\infty, +\infty]$ is a conjugate function of f. Thus, a Fenchel-type dual problem of problem $(\mathrm{P_F})$ is written as

$$(\mathrm{D_F}) \qquad \max_{\boldsymbol{\lambda}} \quad -f^*(\boldsymbol{A}^\top \boldsymbol{\lambda}) - g^*(-\boldsymbol{\lambda}).$$

Clearly, the dual problem $(\mathrm{D_F})$ is equivalent to the following problem:

$$\min_{\boldsymbol{\lambda}} \quad f^*(\boldsymbol{A}^\top \boldsymbol{\lambda}) + g^*(-\boldsymbol{\lambda}).$$

The following theorem guarantees a duality between the primal problem $(\mathrm{P_F})$ and the dual problem $(\mathrm{D_F})$.

Theorem 2.1. *[13, PART VI] If* $\inf(\mathrm{P_F})$ *is finite and*

$$\text{ri dom } g \cap \boldsymbol{A} \text{ ri dom } f \neq \emptyset,$$

there exists an optimal solution in the dual problem $(\mathrm{D_F})$*. Moreover, the following relation holds:*

$$\inf\ (\mathrm{P_F}) = \sup\ (\mathrm{D_F}).$$

□

2.4 An accelerated proximal gradient method

We introduce an accelerated proximal gradient method. We consider the following problem:

$$\min_{\boldsymbol{x}} \quad p(\boldsymbol{x}) + q(\boldsymbol{x}), \tag{2.22}$$

where $p : \mathbb{R}^n \to \mathbb{R}$ is a differentiable and convex function, $q : \mathbb{R}^n \to \mathbb{R}$ is a convex function with a special structure. A proximal gradient method is that combines a proximal point method and a gradient method. When we solve the problem (2.22), we apply the proximal point method to the function q and a gradient method to the function p. An iterative scheme of a proximal gradient method is as follows:

$$\boldsymbol{x}^{k+1} = \operatorname*{argmin}_{\boldsymbol{x}} \left\{ p(\boldsymbol{x}^k) + \langle \boldsymbol{x} - \boldsymbol{x}^k, \nabla p(\boldsymbol{x}^k)\rangle + \frac{1}{2\eta_k} \boldsymbol{B}_\psi(\boldsymbol{x}, \boldsymbol{x}^k) + q(\boldsymbol{x}) \right\}, \tag{2.23}$$

where η_k is a stepsize, $\boldsymbol{B}_\psi : \mathbb{R}^n \to \mathbb{R}$ is a Bregman function defined by $\boldsymbol{B}_\psi(\boldsymbol{x}, \boldsymbol{y}) = \psi(\boldsymbol{x}) - \psi(\boldsymbol{y}) - \langle \nabla\psi(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y}\rangle$ with a differentiable and strongly convex function $\psi : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$. In this paper, we suppose that $\boldsymbol{B}_\psi(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2$.

The iterative scheme (2.23) is divided into the following 2 steps:

$$\begin{aligned} &\text{(step 1)} \qquad \boldsymbol{u}^{k+1} = \boldsymbol{x}^k + \frac{1}{\eta_k}\nabla p(\boldsymbol{x}^k), \\ &\text{(step 2)} \qquad \boldsymbol{x}^{k+1} = \operatorname*{argmin}_{\boldsymbol{x}} \left\{ q(\boldsymbol{x}) + \frac{\eta_k}{2}\|\boldsymbol{x} - \boldsymbol{u}^{k+1}\|^2 \right\}. \end{aligned}$$

We apply the gradient method to the function p in step 1, and apply the proximal point method to function q at neighborhood of $\boldsymbol{u}^{k+1}$ in step 2.

In the problem (2.22), if q is l_1-regularized function $q(\boldsymbol{x}) = \lambda\|\boldsymbol{x}\|_1$, function (2.23) is expressed as

$$\boldsymbol{x}_i^{k+1} = \mathcal{T}_{\lambda\eta_k}\left(x_i^{k-1} - \eta_k \nabla p(\boldsymbol{x}^{k-1})\right) \quad (i = 1, \ldots, n),$$

where the function $\mathcal{T}_\nu : \mathbb{R}^n \to \mathbb{R}$ is the following shrinkage operator [4]:

$$\mathcal{T}_\nu(\boldsymbol{x})_i = \max\left\{(|x_i| - \nu), 0\right\}\operatorname{sgn}(x_i) \quad (i = 1, \ldots, n),$$

where the function sgn: $\mathbb{R} \to \mathbb{R}$ is sign function defined by

$$\operatorname{sgn}(a) = \begin{cases} +1 & (a > 0), \\ 0 & (a = 0), \\ -1 & (a < 0). \end{cases}$$

Now, we introduce an accelerated algorithm of the proximal gradient method proposed by Beck and Teboulle [1]. This algorithm is called Fast Iterative Shrinkage Thresholding Algorithm (FISTA). FISTA with a backtracking stepsize rule is presented in the following.

Algorithm 2.1. FISTA with backtracking

Step 0 Take $L_0 > 0$, some $\eta > 1$, and $\boldsymbol{x}_0 \in \mathbb{R}^n$. Set $y_1 = \boldsymbol{x}_0, t_1 = 1$.

Step k ($k \geq 1$) Find the smallest nonnegative integers i_k such that
with $\bar{L} = \eta^{i_k} L_{k-1}$

$$p(\boldsymbol{x}^k) \leq p(\boldsymbol{y}^k) + \langle \boldsymbol{x}^k - \boldsymbol{y}^k, \nabla p(\boldsymbol{y}^k)\rangle + \frac{\bar{L}}{2}\|\boldsymbol{x}^k - \boldsymbol{y}^k\|^2.$$

Set $L_k = \eta^{i_k} L_{k-1}$ and compute

$$\boldsymbol{x}^k = \operatorname*{argmin}_{\boldsymbol{x}} \left\{ \langle \boldsymbol{x} - \boldsymbol{y}^k, \nabla p(\boldsymbol{y}^k)\rangle + \frac{L_k}{2}\|\boldsymbol{x} - \boldsymbol{y}^k\|^2 + q(\boldsymbol{x}) \right\},$$

$$t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2},$$

$$\boldsymbol{y}^{k+1} = \boldsymbol{x}^k + \left(\frac{t_k - 1}{t_{k+1}}\right)(\boldsymbol{x}^k - \boldsymbol{x}^{k-1}).$$

---

3 Regression problems by Fenchel-type dual problems

In this section, we consider a Fenchel-type dual problem of general problem (1.1). We consider the following regression models introduced in Section 2:

$$\text{(i)} \qquad y^0(\boldsymbol{x}; \boldsymbol{w}) = \boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x}), \tag{3.1}$$

$$\text{(ii)} \qquad y^b(\boldsymbol{x}; \boldsymbol{w}, b) = \boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x}) + b. \tag{3.2}$$

First, we describe the model (i). Given a training dataset $(\boldsymbol{x}_i, t_i)$ $(i = 1, \ldots, N)$, we consider the following problem that minimizes a loss function $g : \mathbb{R}^N \to (-\infty, +\infty]$ with respect to $\boldsymbol{w}$ in order to fit the function (i) to the training dataset:

$$\min_{\boldsymbol{w}} \quad f(\boldsymbol{w}) + g(\boldsymbol{\Phi}\boldsymbol{w} - \boldsymbol{t}), \tag{3.3}$$

where the function $f : \mathbb{R}^M \to (-\infty, +\infty]$ is a regularization function for avoiding over-training, $\boldsymbol{\Phi}$ is a matrix whose elements is $\Phi_{ij} = \phi_j(\boldsymbol{x}_i)$, and $\boldsymbol{t} = [t_1, \ldots, t_N]^\top$. We define a function $g_1 : \mathbb{R}^N \to (-\infty, +\infty]$ by

$$g_1(\boldsymbol{x}) = g(\boldsymbol{x} - \boldsymbol{t}).$$

Then, the problem (3.3) is expressed as

$$(\mathrm{P_i}) \qquad \min_{\boldsymbol{w}} \quad f(\boldsymbol{w}) + g_1(\boldsymbol{\Phi}\boldsymbol{w}).$$

Next, we describe the model (ii). Given a training dataset $(\boldsymbol{x}_i, t_i)$ $(i = 1, \ldots, N)$, we also consider the following problem minimizing a loss function $g : \mathbb{R}^N \to (-\infty, +\infty]$ with respect to $\boldsymbol{w}$:

$$\min_{\boldsymbol{w}} \quad f(\boldsymbol{w}) + \hat{g}_2(\boldsymbol{\Phi}\boldsymbol{w} + b\boldsymbol{e} - \boldsymbol{t}), \tag{3.4}$$

where $\boldsymbol{e} \in \mathbb{R}^N$ is a vector whose elements are all 1. We define functions $\hat{g}_2 : \mathbb{R}^N \to (-\infty, +\infty]$ and $g_2 : \mathbb{R}^N \to (-\infty, +\infty]$ by

$$\begin{aligned} \hat{g}_2(\boldsymbol{x}) &= \inf_{b \in \mathbb{R}} \left\{ g(\boldsymbol{x} + b\boldsymbol{e}) \right\}, \\ g_2(\boldsymbol{x}) &= \hat{g}_2(\boldsymbol{x} - \boldsymbol{t}). \end{aligned}$$

Then problem (3.4) is written as

$$(\mathrm{P_{ii}}) \qquad \min_{\boldsymbol{w}} \quad f(\boldsymbol{w}) + g_2(\boldsymbol{\Phi}\boldsymbol{w} + b\boldsymbol{e} - \boldsymbol{t}).$$

Now, we give the following propositions related to the loss functions g_1 and g_2.

Proposition 3.1. *The conjugate function of the function g_1 is expressed as*

$$g_1^*(\boldsymbol{\xi}) = g^*(\boldsymbol{\xi}) + \langle \boldsymbol{t}, \boldsymbol{\xi} \rangle.$$

Proof. From the definition of the conjugate function,

$$\begin{aligned} g_1^*(\boldsymbol{\xi}) &= \sup_{\boldsymbol{x}\in\mathbb{R}^N} \{\langle \boldsymbol{x}, \boldsymbol{\xi}\rangle - g(\boldsymbol{x}-\boldsymbol{t})\} \\ &= \sup_{\boldsymbol{x}\in\mathbb{R}^N} \{\langle \hat{\boldsymbol{x}}+\boldsymbol{t}, \boldsymbol{\xi}\rangle - g(\hat{\boldsymbol{x}})\} \quad (\text{where } \hat{\boldsymbol{x}} = \boldsymbol{x}-\boldsymbol{t}) \\ &= \sup_{\boldsymbol{x}\in\mathbb{R}^N} \{\langle \hat{\boldsymbol{x}}, \boldsymbol{\xi}\rangle - g(\hat{\boldsymbol{x}})\} + \langle \boldsymbol{t}, \boldsymbol{\xi}\rangle \\ &= g^*(\boldsymbol{\xi}) + \langle \boldsymbol{t}, \boldsymbol{\xi}\rangle. \end{aligned}$$

□

Proposition 3.2. *The conjugate function of the function $\hat{g}_2$ is expressed as*

$$g_2^*(\boldsymbol{\xi}) = \begin{cases} g^*(\boldsymbol{\xi}) + \langle \boldsymbol{t}, \boldsymbol{\xi}\rangle & (\sum_{i=1}^N \xi_i = 0), \\ \infty & (\text{otherwise}). \end{cases}$$

Proof. From the definition of a conjugate function,

$$\begin{aligned} g_2^*(\boldsymbol{\xi}) &= \sup_{\boldsymbol{x}\in\mathbb{R}^N} \{\langle \boldsymbol{x}, \boldsymbol{\xi}\rangle - g_2(\boldsymbol{x}-\boldsymbol{t})\} \\ &= \sup_{\boldsymbol{x}\in\mathbb{R}^N} \left\{\langle \boldsymbol{x}, \boldsymbol{\xi}\rangle - \inf_{b\in\mathbb{R}} \{g(\boldsymbol{x}+b\boldsymbol{e}-\boldsymbol{t})\}\right\} \\ &= \sup_{\boldsymbol{x}\in\mathbb{R}^N,\ b\in\mathbb{R}} \{\langle \boldsymbol{x}, \boldsymbol{\xi}\rangle - g(\boldsymbol{x}+b\boldsymbol{e}-\boldsymbol{t})\} \\ &= \sup_{\boldsymbol{x}\in\mathbb{R}^N,\ b\in\mathbb{R}} \{\langle \hat{\boldsymbol{x}}-b\boldsymbol{e}+\boldsymbol{t}, \boldsymbol{\xi}\rangle - g(\hat{\boldsymbol{x}})\} \quad (\text{where } \hat{\boldsymbol{x}} = (\boldsymbol{x}+b\boldsymbol{e})-\boldsymbol{t}) \\ &= \sup_{\boldsymbol{x}\in\mathbb{R}^N,\ b\in\mathbb{R}} \{\langle \hat{\boldsymbol{x}}, \boldsymbol{\xi}\rangle - g(\hat{\boldsymbol{x}}) - b\langle \hat{\boldsymbol{e}}, \boldsymbol{\xi}\rangle\} + \langle \boldsymbol{t}, \boldsymbol{\xi}\rangle \\ &= \begin{cases} g^*(\boldsymbol{\xi}) + \langle \boldsymbol{t}, \boldsymbol{\xi}\rangle & (\sum_{i=1}^N \xi_i = 0), \\ \infty & (\text{otherwise}). \end{cases} \end{aligned}$$

□

We now give the Fenchel-type dual problem. In this paper, we assume that the regularization function is $f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x}\|^2$. From Proposition 3.1, the Fenchel-type dual problem of ($\mathrm{P_i}$) is given by

$$\min_{\boldsymbol{\lambda}} \quad \frac{1}{2}\|\boldsymbol{\Phi}^\top \boldsymbol{\lambda}\|^2 + g^*(-\boldsymbol{\lambda}) - \langle \boldsymbol{t}, \boldsymbol{\lambda}\rangle.$$

which is written as

$$(\mathrm{D_i}) \qquad \min_{\boldsymbol{\lambda}} \quad \frac{1}{2}\boldsymbol{\lambda}^\top \boldsymbol{K}\boldsymbol{\lambda} + g^*(-\boldsymbol{\lambda}) - \langle \boldsymbol{t}, \boldsymbol{\lambda}\rangle,$$

Where $\boldsymbol{K}$ is the Gram matrix defined by (2.6).

Now, let $\boldsymbol{w}^*$ and $\boldsymbol{\lambda}^*$ be an optimal solutions of the primal problem ($\mathrm{P_i}$) and the dual problem ($\mathrm{D_i}$),

respectively. From the right-hand side of equation (2.21), the following equation holds:

$$\begin{aligned} \boldsymbol{w}^* &= \operatorname*{argmin}_{\boldsymbol{w}} \left\{ \frac{1}{2}\|\boldsymbol{w}\|^2 - g^*(-\boldsymbol{\lambda}^*) - \langle \boldsymbol{\lambda}^*, \boldsymbol{\Phi}\boldsymbol{w} \rangle \right\} \\ &= \boldsymbol{\Phi}^\top \boldsymbol{\lambda}^*. \end{aligned}$$

Therefore, the regression function is expressed as

$$y^0(\boldsymbol{x}) = \sum_{i=1}^{N} \lambda_i^* k(\boldsymbol{x}_i, \boldsymbol{x})$$

In a similar way, from proposition (3.2), the Fenchel-type dual problem ($\mathrm{D_{ii}}$) of ($\mathrm{P_{ii}}$) is given by

$$(\mathrm{D_{ii}}) \qquad \begin{aligned} \min \quad & \tfrac{1}{2}\boldsymbol{\lambda}^\top \boldsymbol{K}\boldsymbol{\lambda} + g^*(-\boldsymbol{\lambda}) - \langle \boldsymbol{t}, \boldsymbol{\lambda} \rangle \\ \text{s.t.} \quad & \sum_{n=1}^{N} \lambda_i = 0. \end{aligned}$$

The regression function is expressed as

$$y^b(\boldsymbol{x}) = \sum_{i=1}^{N} \lambda_i^* k(\boldsymbol{x}_i, \boldsymbol{x}) + b,$$

where $\boldsymbol{\lambda}_{\mathrm{ii}}^*$ is an optimal solution of the dual problem ($\mathrm{D_{ii}}$).

In the following, we provide examples of a loss functions and its conjugate functions [9]. Here, let β_1, β_2 and C be a hyperparameter of a prior distribution of $\boldsymbol{w}$, a hyperparameter of noise in the target data and a regularization parameter, respectively.

Example 3.1. *SVR (linear ε-insensitive loss function)*

The loss function of SVR (linear ε-insensitive loss function) is given by

$$g_\varepsilon(\boldsymbol{x}) = C \sum_{i=1}^{N} \max\left\{0, \ \ |x_i| - \varepsilon\right\}.$$

The conjugate function of g_ε is expressed as

$$\begin{aligned} g_\varepsilon^*(\boldsymbol{\xi}) &= \sum_{i=1}^{N} h_\varepsilon(\xi_i), \\ h_\varepsilon(\xi_i) &= \begin{cases} \varepsilon|\xi_i| & (|\xi_i| \le C), \\ \infty & (\text{otherwise}). \end{cases} \end{aligned}$$

Example 3.2. *SVR (quadratic ε-insensitive loss function)*

The loss function of SVR (quadratic ε-insensitive loss function) is given by

$$g_{\varepsilon^2}(\boldsymbol{x}) = C\sum_{i=1}^{N} \max\left\{0,\ \ |x_i| - \varepsilon\right\}^2.$$

The conjugate function of g_{ε^2} is expressed as

$$\begin{aligned} g^*_{\varepsilon^2}(\boldsymbol{\xi}) &= \sum_{i=1}^{N} h_{\varepsilon^2}(\xi_i), \\ h_{\varepsilon^2}(\xi_i) &= \frac{1}{4C}\xi_i^2 + \varepsilon|\xi_i|. \end{aligned}$$

Example 3.3. *Robust regression (Huber loss function)*

The loss function of a Robust regression (Huber loss function) is given by

$$\begin{aligned} g_H(\boldsymbol{x}) &= C\sum_{i=1}^{N} h(x_i), \\ h(x_i) &= \begin{cases} \frac{1}{2}x_i^2 & (|x_i| \le \varepsilon), \\ \varepsilon|x_i| - \frac{1}{2}\varepsilon^2 & (\text{otherwise}). \end{cases} \end{aligned}$$

The conjugate function of g_H is expressed as

$$\begin{aligned} g^*_H(\boldsymbol{\xi}) &= \sum_{i=1}^{N} h^*(\xi_i), \\ h^*(\xi_i) &= \begin{cases} \frac{1}{2C}\xi_i^2 & (|\xi_i| \le \varepsilon C), \\ \infty & (\text{otherwise}). \end{cases} \end{aligned}$$

Example 3.4. *Ridge regression*

The loss function of a Ridge regression is given by

$$g_{\mathrm{R}}(\boldsymbol{x}) = C\|\boldsymbol{x}\|^2.$$

The conjugate function of g_R is expressed as

$$g^*_R(\boldsymbol{\xi}) = \frac{1}{4C}\|\boldsymbol{\xi}\|^2.$$

The optimization problem of a Ridge regression is given by

$$(\mathrm{D}_R) \qquad \min_{\boldsymbol{\lambda}} \quad \frac{1}{2}\boldsymbol{\lambda}^\top \boldsymbol{K}\boldsymbol{\lambda} + \frac{1}{4C}\|\boldsymbol{\lambda}\|^2 - \langle \boldsymbol{t}, \boldsymbol{\lambda}\rangle.$$

Let $C = \frac{\beta_2}{2\beta_1}$, and let $\hat{\boldsymbol{\lambda}} = \beta_1\boldsymbol{\lambda}$. Then the problem (3.5) is expressed as

$$(\mathrm{D}'_R) \qquad \min_{\hat{\boldsymbol{\lambda}}} \quad \frac{1}{2}\hat{\boldsymbol{\lambda}}^\top \left(\frac{1}{\beta_1}\boldsymbol{K} + \frac{1}{\beta_2}\boldsymbol{I}\right)\hat{\boldsymbol{\lambda}} - \langle \boldsymbol{t}, \hat{\boldsymbol{\lambda}}\rangle.$$

Problem (3.5) corresponds to GPR problem (2.17). Therefore, a Ridge regression is essensially same as GPR.

4 A general support vector regression and the accelerated proximal gradient method

In this section, we propose a general support vector regression that includes SVR and GPR. Furthermore, we present concrete implementation for applying an accelerated proximal gradient method to the Fenchel-type dual problem of the proposed regression.

4.1 A general loss function and regression function

We propose the following loss function $g_p : \mathbb{R}^n \to \mathbb{R}$ with three parameters ε, β and C:

$$
\begin{aligned}
g_p(\boldsymbol{x}) &= \sum_{i=1}^{N} h_p(x_i), \\
h_p(x_i) &= \begin{cases} 0 & (|x_i| \le \varepsilon), \\ \frac{1}{2\beta}(|x_i| - \varepsilon)^2 & (\varepsilon < |x_i| < \varepsilon + \beta C), \\ C(|x_i| - \varepsilon) - \frac{\beta}{2}C^2 & (\varepsilon + \beta C \le |x_i|). \end{cases}
\end{aligned}
$$

This loss function includes the loss functions of the existing regression methods by choosing the parameters as Table 4.1.

Table 4.1: Corresponding parameters

Regression method	ε	β	C
GPR	0	β_2/β_1	∞
SVR(ε)	ε	0	C
SVR(ε^2)	ε	$2C$	∞
robust regression (Huber loss)	0	1	ε
Ridge regression	0	$2C$	∞

We give the conjugate function of the loss function g_p.

Theorem 4.1. *The conjugate function of the loss function $g_p : \mathbb{R}^n \to \mathbb{R}$ is written as*

$$
\begin{aligned}
g_p^*(\boldsymbol{\xi}) &= \sum_{i=1}^{N} h_p^*(\xi_i), \\
h_p^*(\xi_i) &= \begin{cases} \frac{\beta}{2}\xi_i^2 + \varepsilon|\xi_i| & (|\xi_i| \le C), \\ \infty & (\text{otherwise}). \end{cases}
\end{aligned}
$$

Proof. First, note that the conjugate function of g_p is a linear combination of the conjugate functions

of $h_i : \mathbb{R} \to \mathbb{R}$ $(i = 1, \ldots, N)$, that is,

$$\begin{aligned} g^*(\boldsymbol{\xi}) &= \sup_{\boldsymbol{x}\in\mathbb{R}^n} \{\langle \boldsymbol{x}, \boldsymbol{\xi}\rangle - g(\boldsymbol{x})\} \\ &= \sup_{\boldsymbol{x}\in\mathbb{R}^n} \left\{ \sum_{i=1}^N (\xi_i x_i - h_p(x_i)) \right\} \\ &= \sum_{i=1}^N h_p^*(\xi_i). \end{aligned}$$

Next, we give an explicit formula of h^*. First, note that

$$\begin{aligned} h^*(\xi) &= \sup_{x\in\mathbb{R}} \{\xi x - h(x)\} \\ &= \max\Bigg\{ \sup_{-\varepsilon<x<0} \{\xi x\}, \sup_{0<x<\varepsilon} \{\xi x\}, \\ &\qquad \sup_{\substack{-(\varepsilon+\beta C)<x,\\ x<-\varepsilon}} \left\{ \xi x - \frac{1}{2\beta}(|x|-\varepsilon)^2 \right\}, \sup_{\substack{\varepsilon<x,\\ x<\varepsilon+\beta C}} \left\{ \xi x - \frac{1}{2\beta}(|x|-\varepsilon)^2 \right\}, \\ &\qquad \sup_{x\le -(\varepsilon+\beta C)} \left\{ \xi x - \left(C(|x|-\varepsilon) - \frac{\beta}{2}C^2 \right) \right\}, \sup_{\varepsilon+\beta C\le x} \left\{ \xi x - \left(C(|x|-\varepsilon) - \frac{\beta}{2}C^2 \right) \right\} \Bigg\}. \end{aligned}$$

For each supremum, we get

$$\begin{aligned} \sup_{-\varepsilon<x<0} \{\xi x\} &= -\varepsilon\xi, \\ \sup_{0<x<\varepsilon} \{\xi x\} &= \varepsilon\xi, \\ \sup_{\substack{-(\varepsilon+\beta C)<x,\\ x<-\varepsilon}} \left\{ \xi x - \frac{1}{2\beta}(|x|-\varepsilon)^2 \right\} &= \begin{cases} -\varepsilon\xi & (\xi < 0), \\ \frac{\beta}{2}\xi^2 - \varepsilon\xi & (-C \le \xi \le 0), \\ -(\varepsilon+\beta C)\xi - \frac{\beta C^2}{2} & (\xi \le C), \end{cases} \\ \sup_{\substack{\varepsilon<x,\\ x<\varepsilon+\beta C}} \left\{ \xi x - \frac{1}{2\beta}(|x|-\varepsilon)^2 \right\} &= \begin{cases} \varepsilon\xi & (\xi < 0), \\ \frac{\beta}{2}\xi^2 + \varepsilon\xi & (-C \le \xi \le 0), \\ (\varepsilon+\beta C)\xi - \frac{\beta C^2}{2} & (\xi \le C), \end{cases} \\ \sup_{x<-(\varepsilon+\beta C)} \left\{ \xi x - \left(C(|x|-\varepsilon) - \frac{\beta}{2}C^2 \right) \right\} &= \begin{cases} \infty & (\xi < -C), \\ -(\varepsilon+\beta C)\xi - \frac{\beta C^2}{2} & (-C \le \xi), \end{cases} \\ \sup_{\varepsilon+\beta C<x} \left\{ \xi x - \left(C(|x|-\varepsilon) - \frac{\beta}{2}C^2 \right) \right\} &= \begin{cases} (\varepsilon+\beta C)\xi - \frac{\beta C^2}{2} & (\xi < C), \\ \infty & (C \le \xi). \end{cases} \end{aligned}$$

Consequently, we obtain

$$h_p^*(\xi_i) = \begin{cases} \frac{\beta}{2}\xi_i^2 + \varepsilon|\xi_i| & (|\xi_i| \le C), \\ \infty & (\text{otherwise}). \end{cases}$$

□

We give a Fenchel-type dual problem with the loss function g_p. First, we consider the case where a regression model is (i) $y^0(\boldsymbol{x};\boldsymbol{w}) = \boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x})$. From Theorem 4.1, a Fenchel-type dual problem with the loss function g_p is written as

$$\min \quad \frac{1}{2}\boldsymbol{\lambda}^\top (\boldsymbol{K} + \beta \boldsymbol{I})\, \boldsymbol{\lambda} + \varepsilon \|\boldsymbol{\lambda}\|_1 - \boldsymbol{t}^\top \boldsymbol{\lambda} + \delta(\boldsymbol{\lambda}), \tag{4.1}$$

where $\delta : \mathbb{R}^n \to \mathbb{R}$ is an indicator function defined by

$$\delta(\boldsymbol{\lambda}) = \begin{cases} 0 & (|\lambda_i| \leq C \ \text{ for all } i), \\ \infty & (\text{otherwise}). \end{cases}$$

Problem (4.1) can be written as

$$\begin{array}{ll} \min & \frac{1}{2}\boldsymbol{\lambda}^\top (\boldsymbol{K} + \beta \boldsymbol{I})\, \boldsymbol{\lambda} + \varepsilon \|\boldsymbol{\lambda}\|_1 - \boldsymbol{t}^\top \boldsymbol{\lambda} \\ \text{s.t.} & -C \leq \lambda_i \leq C \quad (i = 1, \ldots, N). \end{array} \tag{4.2}$$

Let $\boldsymbol{\lambda}^*$ be an optimal solution of problem (4.2). We have the following regression function by using $\boldsymbol{\lambda}^*$:

$$y^0(\boldsymbol{x}) = \sum_{i=1}^{N} \lambda_i^* k(\boldsymbol{x}_i, \boldsymbol{x}). \tag{4.3}$$

Next, we consider the case that a regression model is (ii) $y^b(\boldsymbol{x};\boldsymbol{w}) = \boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x}) + b$. In a similar way, a Fenchel-type dual problem with the loss function g_p is expressed as

$$\begin{array}{ll} \min & \frac{1}{2}\boldsymbol{\lambda}^\top (\boldsymbol{K} + \beta \boldsymbol{I})\, \boldsymbol{\lambda} + \varepsilon \|\boldsymbol{\lambda}\|_1 - \boldsymbol{t}^\top \boldsymbol{\lambda} \\ \text{s.t.} & -C \leq \lambda_i \leq C \quad (i = 1, \ldots, N), \\ & \displaystyle\sum_{i=1}^{N} \lambda_i = 0. \end{array} \tag{4.4}$$

Let $\boldsymbol{\lambda}^*$ be the optimal solution of problem (4.4). We have the following regression function by using $\boldsymbol{\lambda}^*$:

$$y^b(\boldsymbol{x}) = \sum_{i=1}^{N} \lambda_i^* k(\boldsymbol{x}_i, \boldsymbol{x}) + b.$$

4.2 The accelerated proximal gradient method for general support vector regression

We apply the accelerated proximal gradient method to the proposed problem (4.2). It is described as follows.

Algorithm 4.1. The accelerated proximal gradient method for problem (4.1)

Step 0. Choose parameters t_0, η such that $0 < t_0$ and $0 < \eta < 1$. Choose an initial point $\boldsymbol{x}^0 \in \mathbb{R}^N$.

Set $\boldsymbol{y}^1 = \boldsymbol{x}^0, \theta_1 = 1, k = 1$.

Step 1. Execute the following steps:

Step 1-0 Set $\bar{t} = t_{k-1}$.

Step 1-1 Compute $u_i^k = \mathcal{T}_{\varepsilon\bar{t}}\left(x_i^{k-1} - \bar{t}\nabla f(\boldsymbol{x}^{k-1})\right)$ $(i = 1, \ldots, N)$.

Step 1-2 Compute $x_i^k = \mathrm{mid}\left\{u_i^k, -C, C\right\}$ $(i = 1, \ldots, N)$.

Step 1-3 If $f(\boldsymbol{x}^k) \leq f(\boldsymbol{y}^k) + \langle \boldsymbol{x}^k - \boldsymbol{y}^k, \nabla f(\boldsymbol{y}^k)\rangle + \frac{1}{2\bar{t}}\|\boldsymbol{x}^k - \boldsymbol{y}^k\|^2$, then go to Step 2. Otherwise update $\bar{t} = \eta\bar{t}$, and go to Step 1-1.

Step 2. If some stopping criteria are satisfied, then terminate. Otherwise, update $\theta_{k+1} = \frac{1+\sqrt{1+4\theta_k^2}}{2}$, $\boldsymbol{y}^{k+1} = \boldsymbol{x}^k + \left(\frac{\theta_k - 1}{\theta_{k+1}}\right)(\boldsymbol{x}^k - \boldsymbol{x}^{k-1})$.
Set $k = k + 1$, and go to Step 1.

---

5 Numerical experiments

In this section, we report numerical experiments to investigate the validity of the proposed general SVR. All computations are carried out on a machine with Intel(R) Core(TM) i7-4650U 1.70GHz CPU and 8.00GB memory, and the program is coded in MATLAB R2013a.

We use two datasets, *abalone* and *pumadyn-8nh* [7, 19]. Table 5.1 shows the dimension n of the input data, the number N of training data and the number M of test data. In the experiments, we

Table 5.1: Datasets in the experiments

dataset	n (dimension of the input data)	N (training data)	M (test data)
1. *pumadyn-8nh*	8	4499	3693
2. *abalone*	7	3000	1177

use the Gaussian kernel (2.5) with $\sigma = 1$.

We solve general support vector regression (4.2) with various parameters (ε, β, C) by Algorithm 4.1, and get the regression function (4.3). We set an initial point as $\boldsymbol{x}^0 = [0, \ldots, 0]^\top$. We choose parameters in Algorithm 4.1 as $t_0 = 1, \eta = 1$. Moreover, we adopt the termination criteria as follows:

$$\frac{\|\boldsymbol{x}^k - \boldsymbol{x}^{k-1}\|}{\|\boldsymbol{x}^{k-1}\|} < 10^{-4} \quad \text{and} \quad k > 1000.$$

We investigate the regression from the following points of view:

1. The prediction accuracy;
2. The sparsity of the optimal solution $\boldsymbol{\lambda}^*$.

We evaluate the prediction accuracy by Root Mean Squared Error (RMSE) defined as

$$E_{\mathrm{RMS}} = \sqrt{\frac{\sum_{i=1}^{M}\left(y^0(\boldsymbol{x}_i) - t_i\right)^2}{M}},$$

where $\{(\boldsymbol{x}_i, t_i)\}$ $(i = 1, \ldots, M)$ is test dataset. We also define the following index sets:

$$\begin{aligned} I_{\text{sparse}} &= \left\{ i \middle| |\lambda_i| \leq 10^{-5} \right\}, \\ I_{\text{non-sparse}} &= \left\{ i \middle| |\lambda_i| > 10^{-5} \right\}. \end{aligned}$$

Then, we evaluate the sparsity of the optimal solution $\boldsymbol{\lambda}^*$ by the following "sparsity score":

$$S_{\text{sparsity}} = \left(\frac{\gamma(I_{\text{sparse}})}{\gamma(I_{\text{sparse}}) + \gamma(I_{\text{non-sparse}})} \right) \times 100, \tag{5.1}$$

where $\gamma(\cdot)$ denotes the number of elements in the set. The sparsity score expresses the percentage of sparse elements of $\boldsymbol{\lambda}^*$

The problem (4.2) has parameters ε, β and C, and we implement experiments in various combinations of parameters. For *pumadyn-8nh*, we select the parameters ε, β and C from the following sets, respectively:

$$\begin{aligned} \Omega_\varepsilon &= \{0.0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0\} \quad (11 \text{ values}), \\ \Omega_\beta &= \{0.0, 0.5, 1.0, 1.5, 2.0, 2.5\} \quad (6 \text{ values}), \\ \Omega_C &= \{1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, \infty\} \quad (10 \text{ values}). \end{aligned}$$

We conduct experiments for all 660 combinations $(\varepsilon, \beta, C) \in \Omega_\varepsilon \times \Omega_\beta \times \Omega_C$ in the dataset *pumadyn-8nh*. For *abalone*, we select the parameters ε, β and C from the following sets, respectively:

$$\begin{aligned} \Omega_\varepsilon &= \{0.0, 0.4, 0.8, 1.2, 1.6, 2.0, 2.4, 2.8, 3.2, 3.6, 4.0\} \quad (11 \text{ values}), \\ \Omega_\beta &= \{0.0, 0.025, 0.05, 0.075, 0.1, 0.15, 0.2, 0.25, 0.5, 1.0\} \quad (10 \text{ values}), \\ \Omega_C &= \{2.0, 4.0, 6.0, 8.0, 10.0, 12.0, 14.0, 16.0, 18.0, \infty\} \quad (10 \text{ values}). \end{aligned}$$

We conduct experiments for all 1100 combinations $(\varepsilon, \beta, C) \in \Omega_\varepsilon \times \Omega_\beta \times \Omega_C$ in the dataset *abalone*.

Figures 5.1 and 5.2 show the results of experiments for the dataset *pumadyn-8nh* and *abalone*, respectively, and the horizontal axis shows the RMSE. The vertical axis of the figure shows the sparsity score. Tables 5.2 and 5.3 show results (RMSE and sparsity score) for some specific parameter settings, that is , GPR, SVR1 (linear ε-insensitive function), SVR2 (quadratic ε-insensitive function), a robust regression (Huber loss function) and the proposed regressions. Note also that "the proposed regression" in the tables is the parameter setting that cannot be expressed by any other regression methods.

We first focus on Table 5.2 for *pumadyn-8nh*. The robust regression (Huber) gives the best (smallest) RMSE. On the other hand, 1. SVR(ε) gives the best (largest) sparsity score. Next, we focus on Table 5.2 for *abalone*. GPR gives the best (smallest) RMSE. On the other hand, 1. SVR (ε) gives the best (largest) sparsity score.

In *pumadyn-8nh* and *abalone*, the proposed regression does not have the best scores from the view of both the RMSE and the sparsity score. However, these settings are superior to any other settings either the RMSE or the sparsity score. In other words, the setting of the proposed regression has

the advantages of both GPR and SVR. Therefore, we can regard the proposed regression as one of practical regressions.

Table 5.2: RMSE and sparsity score in *pumadyn-8nh*

parameter settings	ε	β	C	RMSE	sparsity score (%)
1. SVR(ε)	3.0	0.0	3.0	3.4286	**62.33**
2. SVR(ε^2)	2.5	0.5	∞	3.4696	52.23
3. GPR	0.0	2.0	∞	3.3620	0.00
4. robust regression (Huber)	0.0	1.5	2.0	**3.3527**	0.00
5. Proposed regression 1	1.0	0.5	2.0	3.3547	24.29
6. Proposed regression 2	2.5	0.5	2.0	3.4232	53.55

Table 5.3: RMSE and sparsity score in *abalone*

parameter settings	ε	β	C	RMSE	sparsity score (%)
1. SVR(ε)	3.2	0.0	12.0	2.1993	**87.03**
2. SVR(ε^2)	1.6	0.05	∞	2.0889	57.43
3. GPR	0.0	0.025	∞	**1.9930**	0.00
4. robust regression (Huber)	0.0	0.10	18.0	1.9987	0.00
5. proposed regression 1	1.2	0.025	18.0	2.0028	51.77
6. proposed regression 2	2.0	0.025	10	2.0396	71.87

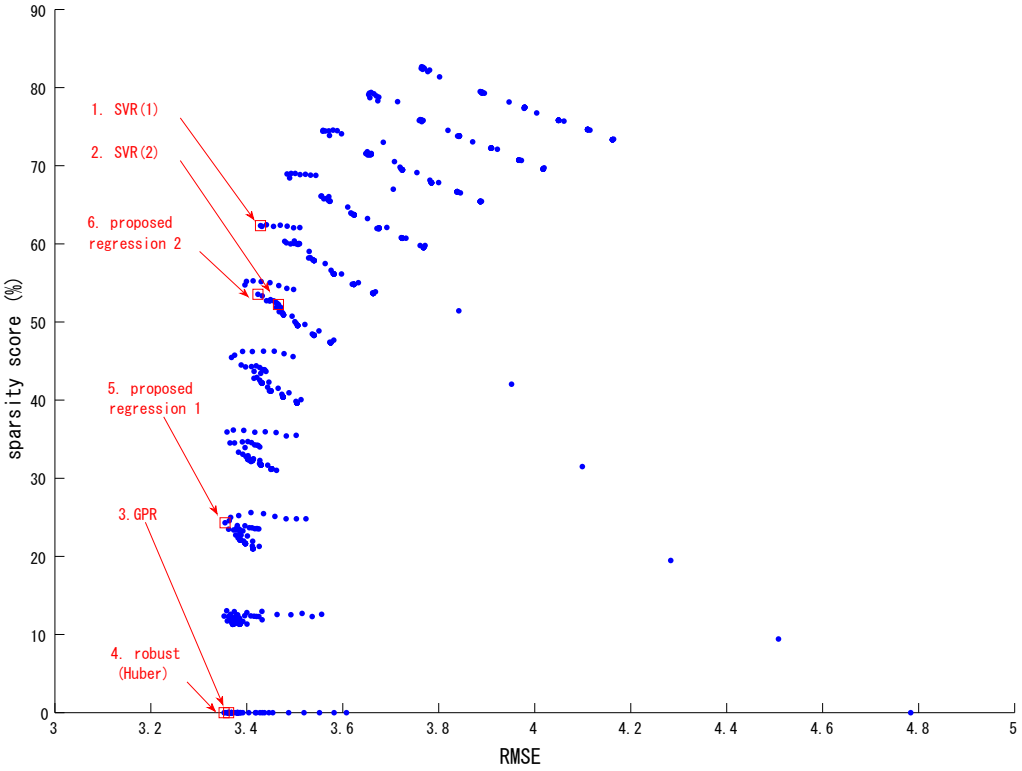

Figure 5.1: RMSE and sparsity score in *pumadyn-8nh*

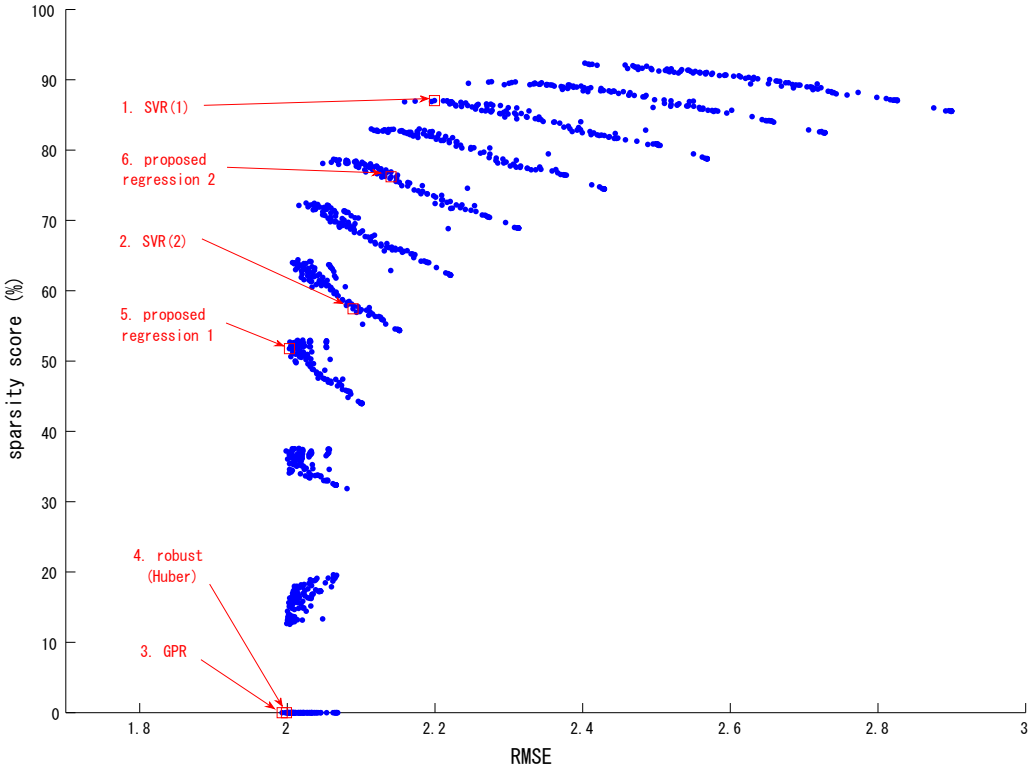

Figure 5.2: RMSE and sparsity score in *abalone*

6 Conclusion

In this paper, we proposed a new general support vector regression with a new general loss function. The proposed regression has the advantages of both GPR and SVR. We considered the Fenchel-type dual problem in order to apply the kernel trick to the proposed general SVR. Moreover, we adapted the accelerated proximal gradient method to solve the Fenchel-type problem, and presented concrete implementation for applying the method effectively, which exploits the special structure of the general SVR. We conducted some experiments, and showed that the proposed regression has the advantages of both (GPR) and (SVR) by setting the parameters suitably.

However, we did not present a method to solve the problem (4.4). This problem has equality constraint $\sum_{i=1}^{N} \lambda_i = 0$, and this make it difficult to solve the problem. Then, to construct an algorithm for solving the problem fast is required for future work.

Acknowledgments

First of all, I would like to express sincere appreciation to Professor Nobuo Yamashita. He always kindly looked after me and gave me plenty of precise advice. Although I sometimes troubled him due to my greenness, he always supported me kindly. It is an honor to have studied under him. I also would like to thank to Assistant Professor Ellen Hidemi Fukuda for her precious advices in the workshops

and seminars. Finally, I would like to thank all members of Yamashita Laboratory, my friends and my family for their encouraging words.

References

[1] A. Beck and M. Teboulle: Fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM J, Imaging Sciences (2008).

[2] C. M. Bishop: Pattern Recognition and Machine Learning, Springer, New York, USA (2006).

[3] C. J. C. Burges: A tutorial on support vector machines for pattern recognition, Knowledge Discovery and Data Mining 2(2), 121–167 (1998).

[4] A. Chambolle, R. A. DeVore and N. Y. Lee, and B. J. Lucier: Nonlinear wavelet image processing: Variational problems, compression, and noise removal through wavelet shrinkage, IEEE Trans. Image Processing, Vol. 7, pp. 319–335 (1998).

[5] N. Cressie: Statistics for Spatial Data, John Wiley, New York (1993).

[6] N. Cristianini and J. Shawe-Taylor: An Introduction to Support Vector Machines, Cambridge University Press, Cambridge, UK (2000).

[7] D. Duvenaud, H. Nickisch and C. E. Rasmussen: Additive Gaussian processes, In Advances in Neural Information Processing Systems (2011).

[8] M. N. Gibbs: Bayesian Gaussian processes for regression and classification, Phd thesis, University of Cambidge (1997).

[9] A. Heinrich: Fenchel duality-based algorithms for convex optimization problems with applications in machine learning and image restoration, Doctoral Thesis, Chemnitz University of Technology (2012).

[10] A. E. Hoerl and R. Kennard: Ridge regression: biased estimation for nonorthogonal problems, Technometrics 12, 55-- 67 (1970).

[11] D. Nguyen-Tuong, M. Seeger and J. Peters: Model learning with local gaussian process regression, Advanced Robotics, Vol. 23, 2015–2034 (2009).

[12] J. C. Platt: Fast training of support vector machines using sequential minimal optimization. In Advances in Kernel Methods - Support Vector Learning, B. Scholkopf, C. J. C. Burges and A. J. Smola, Eds. MIT Press, Cambridge, MA (1998).

[13] R. T. Rockafellar: Convex Analysis, Princeton University Press, Princeton (1970).

[14] R. Tibshirani: Regression shrinkage and selection via the lasso, Journal of the Royal Statistical Society, B 58, 267–288 (1996).

[15] V. Vapnik and A. Chervonenkis: A note on one class of perceptrons, Automation and Remote Control, Vol. 25 (1964).

[16] G. S. Watson: Smooth regression analysis, Sankhya, A 26, 359–372 (1964).

[17] C. K. I. Williams and C. E. Rasmussen: Gaussian processes for regression. In Advances in Neural Information Processing Systems 8. MIT Press (1996).

[18] C. K. I. Williams and M. Seeger (2001): Using the Nystrom method to speed up kernel machines. In T. K. Leen, T. G. Dietterich, and V. Tresp (Eds.), Advances in Neural Information Processing Systems, Vol. 13, 682–688, MIT Press (2001).

[19] F. Yan and Y. Qi: Sparse Gaussian process regression via l1 penalization, In Proceedings of ICML-10, 1183–1190 (2010).