Journal of Machine Learning Research 16 (2015) 3637-3686 Submitted 7/13; Revised 11/14; Published 12/15

Supervised Learning via Euler's Elastica Models

Tong Lin LINTONG@PKU.EDU.CN
Hanlin Xue LINLIE312@GMAIL.COM
Ling Wang LING.WANG.NJ@GMAIL.COM
Bo Huang BOHUANG0321@GMAIL.COM
Hongbin Zha ZHA@CIS.PKU.EDU.CN
Key Laboratory of Machine Perception (Ministry of Education)
School of Electronics Engineering and Computer Science
Peking University, Beijing, 100871, China

Editor: Mikhail Belkin

Abstract

This paper investigates the Euler's elastica (EE) model for high-dimensional supervised learning problems in a function approximation framework. In 1744 Euler introduced the elastica energy for a 2D curve on modeling torsion-free thin elastic rods. Together with its degenerate form of total variation (TV), Euler's elastica has been successfully applied to low-dimensional data processing such as image denoising and image inpainting in the last two decades. Our motivation is to apply Euler's elastica to high-dimensional supervised learning problems. To this end, a supervised learning problem is modeled as an energy functional minimization under a new geometric regularization scheme, where the energy is composed of a squared loss and an elastica penalty. The elastica penalty aims at regularizing the approximated function by heavily penalizing large gradients and high curvature values on all level curves. We take a computational PDE approach to minimize the energy functional. By using variational principles, the energy minimization problem is transformed into an Euler-Lagrange PDE. However, this PDE is usually high-dimensional and can not be directly handled by common low-dimensional solvers. To circumvent this difficulty, we use radial basis functions (RBF) to approximate the target function, which reduces the optimization problem to finding the linear coefficients of these basis functions. Some theoretical properties of this new model, including the existence and uniqueness of solutions and universal consistency, are analyzed. Extensive experiments have demonstrated the effectiveness of the proposed model for binary classification, multi-class classification, and regression tasks.

Keywords: supervised learning, Euler's elastica, total variation, geometric regularization, Euler-Lagrange PDE, function approximation, universal consistency

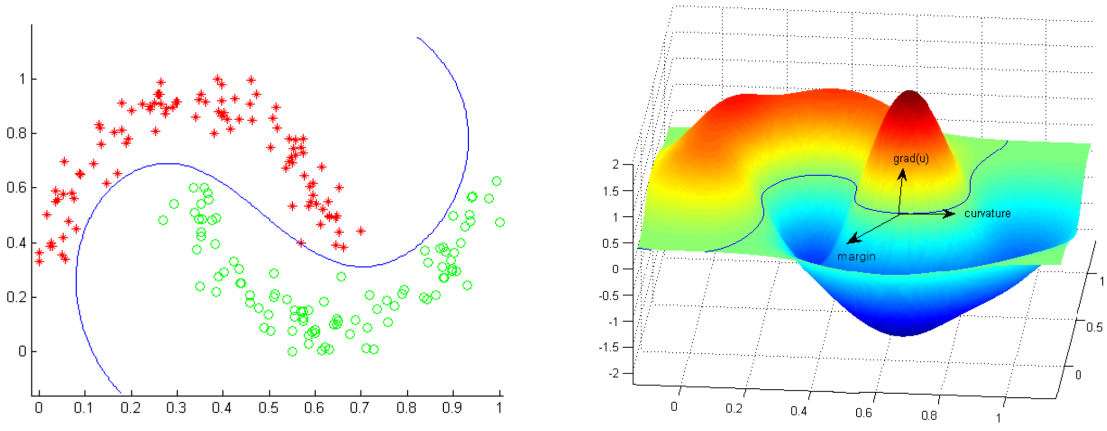
"Read Euler, read Euler, he is our master in everything"
— Pierre-Simon Laplace (1749–1827)

1. Introduction

Supervised learning (Murphy, 2012; Hastie et al., 2009; Bishop, 2006) aims at inferring a function that maps inputs to desired outputs under the guidance of training data. Two main tasks in supervised learning are classification and regression. Numerous supervised learning methods have been developed in several decades; Caruana and Niculescu-Mizil (2006) gave a comprehensive empirical comparison of these methods. A most recent evaluation of classification methods was conducted by Fernández-Delgado et al. (2014): 179 classifiers arising from 17 families were compared on 121 data sets, showing that random forests, support vector machines (SVM), neural networks, and boosting are among the top methods nowadays. Roughly speaking, existing methods can be divided into two main categories: statistics based and function learning based. One advantage of function learning methods is that powerful mathematical theories in functional analysis can be explored rather than doing optimizations on discrete data points.

Most function learning methods can be derived from the energy regularization framework, which minimizes a fitting loss term plus a smoothing penalty. It is arguable that the most successful classification and regression method is the support vector machines (SVM) (Vapnik, 1998; Cristianini and Shawe-Taylor, 2000; Schölkopf and Smola, 2002), whose cost function is composed of a hinge loss and a RKHS norm penalty determined by a kernel. There are several variants of SVM by combining different losses and different penalties (Steinwart, 2005; Bartlett et al., March 2006; Huang et al., 2014). In particular, when replacing the hinge loss by a squared loss, the modified algorithm is called Regularized Least Squares (RLS) method (Rifkin, 2002). Instead of considering a variety of loss terms, manifold regularization (Belkin et al., 2006) introduced a geometric regularizer of squared gradient magnitude on a manifold. Its discrete version corresponds to graph Laplacian regularization (Zhou and Schölkopf, 2005; Nadler et al., 2009). A most recent work is the geometric level set (GLS) classifier (Varshney and Willsky, 2010), with an energy functional composed of a margin-based loss and a geometric regularization term based on the surface area of the decision boundary. The GLS classifier was motivated by the study of minimal surfaces and its applications in image processing. Experiments showed that GLS is competitive with SVM and other state-of-the-art classifiers.

Following the geometric regularization approach, in this paper we propose to use the Euler's elastica for supervised learning problems. The energy functional is composed of a squared loss and an *Euler's elastica* (EE in the sequel) regularizer. Briefly, an elastica regularizer integrates two important geometric factors, gradients and curvatures, in a unified manner. Particularly, its degenerate form is the well-known "total variation" (TV) if only considering gradients and disregarding the influence of curvatures. Since both TV and EE models have achieved great success in image denoising and image inpainting (Chan and Shen, 2005; Aubert and Kornprobst, 2006), a natural question is whether the success of TV and EE models on image processing applications can be transferred to high dimensional data analysis such as supervised learning. This paper investigates the question by extending TV and EE models to supervised learning settings, and evaluating their performance on



(a)

(b)

Figure 1: Results on two moon data by using the EE classifier. (a) Decision boundary (in blue) that separates two classes of points (represented by red stars or green circles); (b) learned target function illustrated as a surface in a 3D space.

benchmark data sets against state-of-the-art methods. Figure 1 shows the classification result and the learned target function on the popular example of two moon dataset by using the EE classifier. Note that three important factors considered in the EE classifier, gradient, curvature, and margin between two classes, are depicted in different directions on one data point of the produced decision boundary in Figure 1(b).

Although some researchers in the machine learning community may think that the supervised learning problems have been widely studied and several leading algorithms like SVM (Vapnik, 1998; Cristianini and Shawe-Taylor, 2000; Schölkopf and Smola, 2002), boosting (Schapire and Freund, 2012), and random forests (Breiman, 2001) have been available to achieve superb classification performance, we argue that this work provides a new perspective on understanding supervised learning problems. Particularly, the contributions of this paper are:

1. A proper balance of three important factors in supervised learning: *margin*, *gradient*, and *curvature*. Here the term *margin* refers to the original geometric meaning used in SVM for binary classification problems, namely, the perpendicular distance from a data point to the decision boundary in the input space. The margin of a SVM classifier $\text{sign}(\mathbf{w}\cdot\mathbf{h}(\mathbf{x}))$ can be written as $y(\mathbf{w}\cdot\mathbf{h}(\mathbf{x}))/(\|\mathbf{w}\|_2\|\mathbf{h}(\mathbf{x})\|_2)$, where $\mathbf{w}$ denotes the coefficients of the separating hyperplane, and $\mathbf{h}(\mathbf{x})$ is the high-dimensional feature vector representation of a data point $\mathbf{x}$. Similarly, the margin in boosting can be defined as $y(\mathbf{w}\cdot\mathbf{h}(\mathbf{x}))/(\|\mathbf{w}\|_1\|\mathbf{h}(\mathbf{x})\|_\infty)$, or simply $yf(\mathbf{x})$ if the combined classifier $f(\mathbf{x})$ has been properly normalized (see Schapire and Freund, 2012, chap. 5). Large margins play a central role in developing several state-of-the-art classifiers. Following the traditions in image processing, in this work the squared loss $(y - f(\mathbf{x}))^2$ is used for easier derivative calculations on both classification and regression tasks. Note that

the squared loss is equivalent to a margin-based loss $(1-yf(\mathbf{x}))^2$, called the quadratic loss (Bartlett et al., March 2006, table 1), since $y \in \{-1,+1\}$ in binary classifications. On the other hand, the term *gradient* is related to the slope of function values in a continuous setting, while the *curvature* measures the degree to which all the level curves (including the decision boundary) is curved. Both gradients and curvatures are geometric measurements that reflect the complexity of the output classifier. The trade-off between the squared loss and the complexity involving gradients and curvatures in this work is new to the machine learning community.

2. Euler-Lagrange PDEs that characterize the optimal solution for supervised learning problems. Historically, PDEs have been used to describe a wide range of physical phenomena such as sound, heat, fluid flow, electrostatics, electrodynamics, or elasticity. Surprisingly, these seemingly distinct physical phenomena can be unified under a PDE framework, which implies that they are essentially governed by same or similar nature's mechanism. A natural question is, can PDEs be applicable to high-dimensional supervised learning problems? To the best of our knowledge, Varshney and Willsky (2010) were the first attempt to propose level set based PDEs for classification. Following this research line, we propose the Euler-Lagrange PDEs derived from Euler's elastica model and its degenerate total-variation model, for classification and regression. These PDEs reveal equilibrium conditions of the desired fitting process for supervised learning.

3. Two numerical algorithms for solving the elastica based supervised learning problem in high dimensions. By using radial basis function approximation, we present two PDE solvers: the gradient descent time marching method and the lagged linear equation iteration method.

The remainder of this paper is organized as follows. In Section 2 we begin with a brief review of TV and EE models used in image processing. The proposed models for supervised learning are described in Section 3, followed by the corresponding numerical solutions presented in Section 4. Some theoretical properties of the proposed models are discussed in Section 5. Section 6 presents the experimental results, and Section 7 concludes the paper.

2. Preliminaries

For better understanding the proposed method, we firstly review the notions of total variation and Euler's elastica from an image processing perspective, and point out some connections with prior work in the machine learning literature.

2.1 Total Variation (TV)

A function is said to have bounded variation (BV functions in the sequel) if its total variation is finite. For simplicity we begin with the classical definition of total variation (TV) for a function of one real variable. The total variation of a real-valued function f defined on an interval $[a,b] \in \mathbb{R}$ is the quantity

$$V_b^a(f) = \sup_P \sum_{i=0}^{n_P - 1} |f(x_{i+1}) - f(x_i)|, \tag{1}$$

where the supremum runs over the set of all partitions P of the given interval $[a,b]$, with n_P being the number of points in a specific partition P. If f is differentiable and its derivative is Riemann-integrable, the total variation can be written as

$$V_b^a(f) = \int_a^b |f'(x)| dx.$$

Intuitively it measures the total distance along the direction of the y-axis, neglecting the contribution of motion along x-axis, traveled by a point moving along the graph. Notice that if $f'(x) > 0$ for all $x \in [a,b]$, it is simply equal to $f(b) - f(a)$ by the fundamental theorem of calculus.

The modern definition is based on the concept of distributional derivatives. Let $\Omega \subset \mathbb{R}$ be a bounded open interval. A function $f \in L^1(\Omega)$ is said to be of *bounded variation* (BV) if

$$\sup_\varphi \Big\{ \int_\Omega f(x)\varphi'(x)dx : \varphi \in C_c^1(\Omega), \|\varphi\|_{L^\infty(\Omega)} < 1 \Big\} < \infty, \tag{2}$$

where $C_c^1(\Omega)$ is the space of continuously differentiable functions with compact support in Ω, and $\|\cdot\|_{L^\infty(\Omega)}$ is the essential supremum norm. Note that this definition may have some variants, e.g. imposing the test function that satisfies $\varphi \in C_c^\infty(\Omega)$ and $\|\varphi\|_{C^0(\Omega)} < 1$ (Golubov and Vitushkin, 2001). An equivalent definition is that BV functions are functions whose distributional derivative is a finite Radon measure. Also the two definitions (1) and (2) are consistent. It is natural to generalize the definition (2) for functions of several variables. For an open $\Omega \subset R^d$, the total variation of $f \in L^1(\Omega)$ is given by

$$\sup_\varphi \Big\{ \int_\Omega f \nabla \cdot \varphi \, dx : \varphi = (\varphi_1, \varphi_2, \cdots, \varphi_d) \in C_c^1(\Omega, R^d), \|\varphi\|_{L^\infty(\Omega)} < 1 \Big\} < \infty, \tag{3}$$

where φ is a vector-valued test function, $\nabla \cdot \varphi = \sum \partial \varphi_i / \partial x_i$ is the divergence operator, and all the components of φ has a $L^\infty(\Omega)$-norm less than one. For more details of TV definitions and the BV function space, one can refer to Chan and Shen (2005), Aubert and Kornprobst (2006), Ambrosio et al. (2000), Giusti (1994), and Golubov and Vitushkin (2001).

By penalizing large gradients of the target functions, total variation has been widely used for image processing tasks such as denoising and inpainting. The pioneering work is Rudin, Osher, and Fatemi's image denoising model (Rudin et al., 1992):

$$E[u] = \int_\Omega \Big((u - u_0)^2 + \lambda |\nabla u| \Big) dx,$$

where u_0 is the input image with noise, u is the desired output image, λ is a regulation parameter that balances the two terms, ∇u is the gradient vector $(\partial u/\partial x, \partial u/\partial y)$ for a function $u(x,y)$, $|\nabla u|$ is the l_2-length of the gradient vector, and Ω denotes a $2D$ rectangular image domain. The first fitting term measures the fidelity to the input, while the second is a p-Sobolev regularization term ($p = 1$) where the gradient ∇u is understood in the distributional sense. The main benefit is to preserve significant image edges during the denoising procedure (Chan and Shen, 2005; Aubert and Kornprobst, 2006), as image edges are important features that should be faithfully retained in image processing. The common downside of TV-based methods is that piecewise constant images with $|\nabla u| = 0$ almost everywhere are favored over piecewise smooth images, which is the so-called staircasing effect (Duan et al., 2013). Euler's elastica model is one of high order approaches to overcome this drawback, which is described in the next subsection.

In the machine learning literature, p-Sobolev regularizer can be found in the literature of nonparametric smoothing splines, generalized additive models, and projection pursuit regression models (Hastie et al., 2009). Specifically, Belkin et al. (2006) proposed the manifold regularization term

$$\int_{x \in M} |\nabla_M u|^2 dx,$$

for any smooth function $u(x)$ on a manifold M. On the other hand, discrete graph Laplacian regularization was discussed in Zhou and Schölkopf (2005) as

$$\sum_{v \in V} |\nabla_v u|^p,$$

where v is a vertex from a vertex set V, and p is an arbitrary number. This penalty measures the roughness of the discrete function u over a graph.

2.2 Euler's Elastica (EE)

The elastica energy first appeared in Euler's work in 1744 on modeling torsion-free thin elastic rods (for the history see Levien, 2008; Fraser, 1991). Then Mumford (1994) reintroduced elastica into computer vision for measuring the quality of interpolating curves in disocclusion. Later, elastica based image inpainting methods were developed in Masnou and Morel (1998) and Chan et al. (2002).

A smooth curve γ is said to be Euler's elastica if it is the equilibrium curve of the elasticity energy:

$$E[\gamma] = \int_{\gamma} (a + b\kappa^2) ds, \tag{4}$$

where a and b are two non-negative constant weights, κ denotes the scalar curvature (see Appendix A for its definition), and ds is an infinitesimal arc length element. Euler obtained the energy in studying the steady shape of a thin and torsion-free rod under external forces. The curve implies the lowest elastica energy, thus getting its name. The ratio a/b (if $b \neq 0$) indicates the relative importance of the total length versus total squared curvature (Chan and Shen, 2005, chap. 2.1).

According to Mumford (1994), the key link between the elastica curves and image inpainting relies on the the interpolation capability of elasticas. Elasticas were discovered to comply to the *connectivity principle* (Chan and Shen, 2001; Kanizsa, 1979) in visual perception better than total variation. This principle in vision psychology shows that humans mostly prefer having two disjoint parts occluded by another object connected psychologically, even when they are far apart. Such kinds of "nonlinear splines", like classical polynomial splines, are natural tools for completing the missing or occluded edges. Besides, there is an interesting Bayesian rationale revealed by Mumford (1994) (see also Chan et al., 2002) by considering the *random walk* of a drunk. Suppose the drunk starts from the origin of a 2-D ground and each step is straight. With some distribution assumptions on the step size and the orientation of each step, the maximum likelihood estimation (MLE) of such discrete random walk is approximately equivalent to the minimization of the elastica energy (4) in a continuous fashion. This drunk walking model also sheds light on the choice of "2" for the curvature power in (4). For any $p > 1$, one could consider the general p-elastica energy

$$E_p[\gamma] = \int_\gamma (a + b|\kappa|^p) ds.$$

Notice that the situation of $p = 1$ is less ideal since in this case the total curvature energy permits sudden turns. Chan et al. (2002) pointed out that generic stationary points of the p-elastica energy are forbidden when $p \geq 3$, implying that $p \in (1, 3)$ sounds to be a good choice.

A common approach to bridge the gap between a prior energy model for curves and that for images is using level sets (or called isophotes), pioneered by Osher and Sethian (1988). By "lifting" a curve prior model into a 2D space, one can construct an image prior model imposed on all the level curves of an image (corresponding to a 2D function). Formally, the Euler's elastica of all the level curves of an image u can be expressed as

$$E[u] = \int_{l=0}^{L} \int_{\gamma_l : u=l} (a + b\kappa^2) ds dl, \tag{5}$$

where γ_l is the level curve determined by $u(x) = l$, and the level value l varies in the image range $[0, L]$. Let dt denote an infinitesimal length element along the normal direction $\mathbf{n}$ of the level curve (or along the steepest ascent curve), then we have

$$\frac{dl}{dt} = |\nabla u| \quad \text{or} \quad dl = |\nabla u| dt.$$

Thus by the *co-area formula* (Giusti, 1994), the integrated elastica energy (5) now passes on to u by

$$E[u] = \int_{l=0}^{L} \int_{\gamma_l : u=l} (a + b\kappa^2) |\nabla u| dt ds = \int_\Omega (a + b\kappa^2) |\nabla u| dx,$$

since dt and ds represent a couple of orthogonal length elements. Here Ω denotes the whole rectangular image domain. Now the elastica energy of an image is completely expressed in terms of u, when considering the well known *curvature formula* (Morel and Solimini, 1995) for any level curve $\gamma_l : u(x) = l$

$$\kappa = \nabla \cdot \mathbf{N} = \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right), \tag{6}$$

where $\nabla \cdot$ denotes the divergence operator, defined as

$$\nabla \cdot \mathbf{V} \doteq \frac{\partial A}{\partial x} + \frac{\partial B}{\partial y}$$

for a vector $\mathbf{V} = (A, B)$, and $\mathbf{N}$ is the ascending unit normal field $\nabla u / |\nabla u|$. See Appendix A for a short derivation of (6). Of course this curvature expression makes sense only for a certain class of smooth functions (such as $C^2(\Omega)$) and requires to be relaxed in order to handle more general functions (like BV or L^1 functions).

Given a small image region D to be inpainted in the whole image domain Ω, Chan and Shen (2005) proposed an inpainting model based on Euler's elastica

$$E = \int_{\Omega \backslash D} (u - u_0)^2 dx + \lambda \int_{\Omega} (a + b\kappa^2) |\nabla u| dx, \tag{7}$$

where λ is a trade-off parameter that balances the first fitting term and the second smoothing term. Notice that the second term in (7) is an elastica regularizer that penalizes high elastica energy on all the level curves of $u(x)$, as expressed in (5). By using *calculus of variation* (van Brunt, 2004), its minimization is reduced to a nonlinear Euler-Lagrange equation. Its numerical method can be implemented by a finite difference scheme, and experimental results show that this elastica based inpainting method performs better than TV based approaches.

Note that total variation can be regarded as a degenerate form of Euler's elastica if setting $a = 1$ and $b = 0$ in (7). In fact, elastica is a combination of total variation that suppresses oscillations in the gradient direction, and a curvature regularizer that penalizes non-smooth level set curves (see Figure 1).

3. The Proposed Framework

We first set up the supervised learning problem, and then introduce three models, Laplacian, total variation, and Euler's elastica, in an increasing order of computational complexity.

3.1 Problem Setup

The general supervised learning problem can be described as follows:

- Given a training data set $\{(\mathbf{x}_1, y_1), ...(\mathbf{x}_n, y_n)\}$ where each data point $\mathbf{x}_i \in \Omega \subset \mathbb{R}^d$ is a d-dimensional column vector and y_i is the corresponding target variable, the goal is to estimate an unknown function $u(\mathbf{x})$ for predicting the desired y on a newly coming point $\mathbf{x}$.

The difference between classification and regression lies only in the corresponding target values, with one discrete and the other continuous. For regression, we simply use $u(\mathbf{x})$ to approximate the target values; for binary classification, the decision boundaries are given by the zero level set of $u(\mathbf{x})$, or $\text{sign}(u(\mathbf{x}))$. Most popular multi-class classifiers are based on some types of reductions to binary classifications; we defer the discussion of multi-class problems to the experiments section.

The widely used functional regularization framework for supervised learning can be formulated as:

$$\min_u \lambda S(u) + \sum_{i=1}^{n} L(y_i, u(\mathbf{x}_i)), \tag{8}$$

where $S(u)$ is a smoothing term or called a penalty and $L(\cdot)$ denotes a loss function. The penalty term is used to control the complexity of the learned function, which has proven to be essential in *Statistical Learning Theory* (Vapnik, 1998; Bousquet et al., 2004; Boucheron et al., 2005; von Luxburg and Schölkopf, 2008). The misclassification risk corresponds to the use of 0-1 loss: $L_{0-1}(y, u(\mathbf{x})) = \mathbf{1}[y \neq \text{sign}\, u(\mathbf{x})]$, where $\mathbf{1}[\alpha]$ denotes an indicator function that is 1 if α holds true and 0 otherwise. Or we can slightly misuse the notation to allow a margin based representation: $L_{0-1}(y, u(\mathbf{x})) = \mathbf{1}(yu(\mathbf{x}))$, where $\mathbf{1}(\alpha)$ is 1 if $\alpha \leq 0$ and 0 otherwise. It is well known that directly minimizing the 0-1 loss is computationally intractable for many nontrivial classes of functions, and often some nonnegative convex nondecreasing loss function are considered for computational efficiency. Another advantage of such convex surrogates for 0-1 loss is that it is possible to demonstrate the Bayes-risk consistency and to obtain uniform upper bounds on the generalization risk. See Bartlett et al. (March 2006) and Boucheron et al. (2005, Section 4.2) for more discussions.

In the literature a variety of convex surrogate loss functions $L(.)$ have been proposed for binary classification where $y \in \{-1, +1\}$, such as:

1. hinge loss $L_{hinge}(y, u(\mathbf{x})) = \max\{0, 1 - yu(\mathbf{x})\}$ for SVM;
2. squared loss $L_{squared}(y, u(\mathbf{x})) = (y - u(\mathbf{x}))^2$ for RLS;
3. logistic loss $L_{logistic}(y, u(\mathbf{x})) = \log(1 + \exp(-yu(\mathbf{x})))$ for logistic regression;
4. and exponential loss $L_{exponential}(y, u(\mathbf{x})) = \exp(-yu(\mathbf{x}))$ in boosting.

Except for the squared loss, other above losses are margin-based since the classification margin $yu(\mathbf{x})$ is explicitly used. When restricting the discussion on binary classification where $y \in \{-1, +1\}$, the squared loss is actually equivalent to the quadratic loss $(1 - yu(\mathbf{x}))^2$ which is then margin-based.

Throughout the paper, the squared loss is used in all our models due to several reasons: (1) The derivative of a squared loss is very simple to calculate; (2) It can be applied to both classification and regression, without any modification; (3) For classification, Rifkin (2002) showed that the RLS method based on squared loss can offer comparable or slightly better accuracies than hinge loss based SVM; (4) Using squared loss is consistent to the related work in image processing area, leading to identical or similar PDEs; (5) We have no intention to exhaustively try and compare different loss functions; instead our focus is on the second term which is a new geometric regularization for supervised learning. For more loss functions and penalties, one can refer to Steinwart (2005), Bartlett et al. (March 2006), and Huang et al. (2014).

Our goal is to explore how TV and EE can be applied to classification and regression problems on high dimensional data sets. To this end, we prefer a continuous integral form rather than the discrete summation form in (8). In contrast to discrete methods such as SVM and graph Laplacian, the proposed framework operates in a continuous fashion where powerful mathematical analysis tools can play a role. Specifically, the calculus of variations plays a role in minimizing the energy functional, leading to the Euler-Lagrange PDE. A typical procedure of this computational PDE approach has three steps: (1) Set up the function learning problem under a continuous setting by designing a proper energy functional; (2) Derive the Euler-Lagrange PDE via the calculus of variations; (3) Finally solve the PDE numerically on discrete data points.

3.2 Laplacian Regularization (LR)

A commonly used model with squared loss can be written as

$$\min_u \lambda S(u) + \sum_{i=1}^n \Big(u(\mathbf{x}_i) - y_i\Big)^2.$$

If the RKHS norm is used as the smoothing term $S(u)$, the model is called regularized least squares (RLS) (Rifkin, 2002). Another natural choice is the squared L_2-norm of the gradient: $S(u) = |\nabla u|^2$, as proposed in Belkin et al. (2006). We need to move from the discrete cost function to a continuous functional to leverage powerful mathematical tools. Suppose $\Omega \in \mathbb{R}^d$ is a regular region that contains all the given data points. Under a continuous setting, we have the following Laplacian regularization (LR) model:

$$E_{LR}[u] = \int_\Omega \Big(\lambda|\nabla u|^2 + (u-y)^2\Big) d\mathbf{x}. \tag{9}$$

This LR model has been widely used in the image processing literatures. By calculus of variations (see Appendix B), the minimization is reduced to the following Euler-Lagrange PDE with a *natural boundary condition* over the boundary $\partial\Omega$:

$$\left\{\begin{array}{l} -\lambda\Delta u + (u-y) = 0, \\ \frac{\partial u}{\partial \mathbf{n}}|_{\partial\Omega} = 0, \end{array}\right. \tag{10}$$

where Δu is the Laplacian operator of u defined as

$$\Delta u \doteq \nabla^2 u = \nabla \cdot \nabla u = \sum_{i=1}^d \frac{\partial^2 u}{\partial (x^{(i)})^2},$$

and $\mathbf{n}$ denotes the outer normal of $\partial\Omega$. This PDE (10) is relatively simple and can be easily solved using common methods in two and three dimensions. The next section provides a function approximation method for solving the PDE in high dimensions.

One can observe that the PDE (10) is very similar to the Poisson's equation $-\Delta u = f$ in mathematical physics, where f is a given function. Hence its behavior shares certain degrees of similarity with Poisson's equation. Particularly, if u fits y perfectly (satisfying $u-y=0$) in a small neighborhood of a particular point $\mathbf{x}$, then by (10) we have $\Delta u = 0$ and further by $u - y = 0$ we also have $\Delta y = 0$ in this neighborhood. On the contrary, if $\Delta y \neq 0$ (implying that $y(\mathbf{x})$ is not a harmonic function), then we can not obtain $u-y=0$; otherwise by (10) we have $\Delta u = 0$ and $\Delta y = 0$, which is contradictive to our assumption $\Delta y \neq 0$. Therefore, the smoothness of the target variable $y(\mathbf{x})$ determines the fitting degree for supervised learning. The regularization parameter λ controls the strength of this connection.

Throughout the paper, the *natural boundary condition* is adopted for easier treatments. It is well known that boundary conditions can play a significant role in traditional low-dimensional PDE areas, where the shape of the domain boundary is explicitly determined. In these situations, boundary conditions are given by the underlying real problems and their physical meanings are clear. However, in our case of high dimensional spaces for supervised learning, there is no need to specify the exact domain boundary as long as this domain contains all the data points. Often the input data is preprocessed by scaling each attribute into the range $[-1, +1]$ or $[0, 1]$, and hence in practice we define the domain of our TV/EE models as a d-dimensional hypercube. Scaling has been a very important step for using neural networks and SVM, with some advantages discussed in Hsu et al. (2007). Most of these considerations also apply to our algorithms. Recall that our focus is to learn the target function $u(\mathbf{x})$ on an "active" region that contains both the given training data and the future test data, whereas this active region is usually far away from the boundary of the hypercube domain in our settings. Hence boundary values in our high dimensional models are not so important as in low dimensional spaces, and we use the natural boundary condition purely from a computational aspect, just like the related work in image processing. Note that in the GLS classifier (Varshney and Willsky, 2010), the issue of PDE boundary conditions was treated in a similar way.

3.3 Total Variation (TV)

Similar to image denoising, the total variation (TV) model for supervised learning can be formulated as

$$E_{TV}[u] = \int_\Omega \Big(\lambda |\nabla u| + \frac{1}{2}(u-y)^2\Big) d\mathbf{x}. \tag{11}$$

The only difference between LR and TV is just on the p-Sobolev regularizer with $p = 2$ for LR and $p = 1$ for TV, respectively. Intuitively, LR penalizes gradients on edges too much due to the squared gradient magnitude, while TV is rather milder to permit sharper edges near the decision boundaries between two classes. Similarly, by calculus of variations (see Appendix B) we get the following PDE, which is the exactly same to that in image denoising area:

$$-\lambda \nabla \cdot \left(\frac{\nabla u}{|\nabla u|}\right) + (u-y) = 0. \tag{12}$$

Note that by the same curvature notation (6) of the associated level hypersurfaces, (12) can be compactly written as

$$-\lambda\kappa + (u-y) = 0. \tag{13}$$

See Appendix A for this curvature notation in R^d, which amounts to the mean curvature up to a constant factor $1/(d-1)$. The PDE (13) implies that the mean curvature κ of all level hypersurfaces with respect to the approximation function $u(\mathbf{x})$ imposes an equilibrium condition on the fitting process of $u - y = 0$.

3.4 Euler's Elastica (EE)

The more complicated elastica model for supervised learning can be formulated as

$$E_{EE}[u] = \int_\Omega \Big(\lambda(a + b\kappa^2)|\nabla u| + \frac{1}{2}(u-y)^2\Big) d\mathbf{x}, \tag{14}$$

where κ is given by (6). Due to the elastica regularizer, the final decision boundary and all level sets of $u(\mathbf{x})$ should have a low elastica energy. If setting $a = 1$ and $b = 0$, this model degenerates to the total variance model. Therefore, a unified solution can be implemented for both TV and EE models, as described in the next section.

Using calculus of variations, we obtain the following PDE for the elastica model:

$$-\lambda \nabla \cdot \mathbf{V}(u) + (u - y) = 0, \tag{15}$$

where the vector field $\mathbf{V}(u)$ is called the *flux* of the elastica energy related to $u(\mathbf{x})$ and can be expressed as a decomposition in a natural orthogonal frame $(\mathbf{N}, \mathbf{T})$:

$$\begin{aligned} \mathbf{V}(u) &\doteq f(\kappa)\mathbf{N} - \frac{\mathbf{T}}{|\nabla u|}\frac{\partial(f'(\kappa)|\nabla u|)}{\partial \mathbf{T}} \\ &= f(\kappa)\mathbf{N} - \frac{1}{|\nabla u|}\Big\{\nabla(f'(\kappa)|\nabla u|) - \mathbf{N}\langle \mathbf{N}, \nabla(f'(\kappa)|\nabla u|)\rangle\Big\} \\ &= f(\kappa)\mathbf{N} - \frac{1}{|\nabla u|}\nabla(f'(\kappa)|\nabla u|) + \frac{1}{|\nabla u|^3}\nabla u\langle \nabla u, \nabla(f'(\kappa)|\nabla u|)\rangle. \end{aligned} \tag{16}$$

Here $f(\kappa) \doteq 1 + b\kappa^2$ by fixing $a = 1$ for simplicity, and $\mathbf{N}$, $\mathbf{T}$ are the normal and tangent vectors given by:

$$\mathbf{N} = \frac{\nabla u}{|\nabla u|}, \quad \mathbf{T} = \mathbf{N}^{\perp}.$$

The directional derivative along $\mathbf{T}$ for a function u is defined as the inner product of ∇u and $\mathbf{T}$:

$$\partial u / \partial \mathbf{T} \doteq \nabla u \cdot \mathbf{T} = \langle \nabla u, \mathbf{T} \rangle.$$

See Appendix B for the detailed derivations from (14) to (15), which originates from Chan et al. (2002). When $b = 0$, (15) degenerates to (12) as $f'(\kappa) = 0$ and $\kappa = \nabla \cdot \mathbf{N}$. Again, the PDE (15) indicates that the divergence of the flux vector field, namely the first term $\nabla \cdot \mathbf{V}(u)$, imposes an equilibrium condition on the fitting process of $u - y = 0$.

4. Numerical Algorithms

Due to the nonlinearity of the regularizer in TV and EE models, the corresponding PDEs in (12) and (15) are too complicated to be efficiently solved in high dimensional space. Even though the PDE in (10) associated with the LR model can be solved by Finite Difference Method (FDM) or Finite Element Method (FEM) in 2-D or 3-D spaces, currently we have no PDE tools to deal with such high dimensional problems. Therefore we take a function approximation idea by using radial basis functions (RBF), similar to the treatment in GLS (Varshney and Willsky, 2010). Then the computational PDE problems can be reduced to finding the expanding coefficients.

In the literature of image denoising and inpainting, dynamic programming was firstly employed to solve elastica related image processing problems in Masnou and Morel (1998). The most widely used method is the computational PDE approach (Chan and Shen, 2005; Aubert and Kornprobst, 2006), partially due to the following reasons:

1. The theory of PDEs is well established;

2. Many variational problems or their regularized approximations can often be effectively computed from their Euler-Lagrange equations;

3. As in classical mathematical physics, PDEs are powerful tools to describe, model, and simulate many dynamic as well as equilibrium phenomena.

Later in Bae et al. (2011) and Komodakis and Paragios (2009), graph-cuts methods are applied to elastica models. Several numerical solutions (Tai et al., 2011; Hahn et al., 2011; Duan et al., 2013) are based on the operator splitting technique and the augmented Lagrangian method (ALM), which decomposes the original problem into a series of subproblems. All subproblems are either linear which can be solved efficiently by iterative solvers, or having closed-form solutions. Recently, Bredies et al. (2013) proposed a convex, lower semi-continuous approximation of Euler's elastica energy on image processing tasks via functional lifting, which can be expressed as a linear program. However, it is still unclear whether these newly developed numerical methods are applicable to high dimensional elastica problems.

4.1 Approximation by Radial Basis Functions

The function approximation idea relies on the fact that a function $u(\mathbf{x})$ can be expressed as a sum of weighted basis function $\{\phi_i(\mathbf{x})\}$. For instance, a Taylor expansion represents a function by using polynomials as basis functions. The Ritz method is a direct method for solving problems in variational calculus by means of a linear combination of known basis functions. In the literature of machine learning, the most widely used are the Gaussian radial basis function (RBF) kernels, which are simple in expressions but have powerful fitting ability. Hence we assume that the function $u(\mathbf{x})$ to be learned has the following representation

$$u(\mathbf{x}) = \sum_{i=1}^{n} w_i \phi_i(\mathbf{x}), \tag{17}$$

where $\{\phi_i(\mathbf{x})\}$ are a set of Gaussian RBF kernels

$$\phi_i(\mathbf{x}) \doteq \exp(-\frac{1}{2}c||\mathbf{x} - \mathbf{x}_i||^2).$$

Here $\{\mathbf{x}_i\}$ are the training samples in supervised learning, and c is a tunable parameter. Note that the granularity of this representation is well-matched to the data size, as the number of RBFs is equal to the number of training samples. By using the RBF approximation, the problem is reduced to finding the coefficients $\{w_i\}$. Hence our approach is similar to kernel machines with the Gaussian RBF kernels since the decision function is formulated as a linear combination of RBFs. The main difference is that our approach is based on the Euler's elastica regularization term, while kernel methods in the literature employs a squared norm of reproducing kernel Hilbert space for regularization.

Though there are numerous basis functions (also known as kernels) being proposed by researchers, four basic types are often considered in the SVM literature and related books: linear, polynomial, sigmoid, and Gaussian RBFs. In Hsu et al. (2007) the Gaussian RBF kernel is suggested to be a reasonable first choice for training SVMs due to several reasons. Most of these considerations also apply to our algorithms, such as the number of hyperparameters, and the difficulties in numerical computations. In addition, one might consider other types of RBFs instead of Gaussians, like compactly supported RBFs used in scattered data interpolation (Wendland, 1995; Floater and Iske, 1996). The main purpose of compactly supported RBFs is for reducing computational complexity. However, the usage of compactly supported RBFs might lead to numerical difficulties in the following derivative calculations in our algorithms.

Let $\mathbf{H}(u)$ denote the Hessian matrix of u, and $\mathbf{I}$ be an identity matrix with a proper size. For short notations we also use ϕ_i for $\phi_i(\mathbf{x})$. Based on the RBF approximation (17), the following are some analytical expressions and handy notations that will be frequently used later. See Appendix C for some derivations of these expressions. Note that d is the dimension of the feature space.

$$\begin{aligned}
\nabla\phi_i &= -c(\mathbf{x}-\mathbf{x}_i)\phi_i, \\
\Delta\phi_i &= c(c|\mathbf{x}-\mathbf{x}_i|^2-d)\phi_i, && (18)\\
\mathbf{H}(\phi_i) &= -c\phi_i\mathbf{I}+c^2(\mathbf{x}-\mathbf{x}_i)(\mathbf{x}-\mathbf{x}_i)^T\phi_i, && (19)\\
\nabla u &= \sum_i w_i\nabla\phi_i = -c\sum_i w_i(\mathbf{x}-\mathbf{x}_i)\phi_i = -c\mathbf{g}, \\
\mathbf{g} &\doteq \sum_i w_i(\mathbf{x}-\mathbf{x}_i)\phi_i, && (20)\\
\Delta u &= \sum_i w_i\Delta\phi_i = c\sum_i w_i(c|\mathbf{x}-\mathbf{x}_i|^2-d)\phi_i, \\
\mathbf{H}(u) &= -c\Big(\sum_i w_i\phi_i\Big)\mathbf{I}+c^2\Phi, && (21)\\
\Phi &\doteq \sum_i w_i(\mathbf{x}-\mathbf{x}_i)(\mathbf{x}-\mathbf{x}_i)^T\phi_i, \\
\mathbf{N} &\doteq \frac{\nabla u}{|\nabla u|} = -\frac{\mathbf{g}}{|\mathbf{g}|}, \\
\kappa &\doteq \nabla\cdot\frac{\nabla u}{|\nabla u|} && (22)\\
&= \frac{1}{|\nabla u|}\left(\Delta u-\frac{\nabla u^T H(u)\nabla u}{\nabla u^T\nabla u}\right) \\
&= \frac{1}{|\mathbf{g}|}\Big\{\sum_i w_i(c|\mathbf{x}-\mathbf{x}_i|^2-d+1)\phi_i-c\frac{\mathbf{g}^T\Phi\mathbf{g}}{\mathbf{g}^T\mathbf{g}}\Big\}. && (23)
\end{aligned}$$

4.2 Algorithm for LR

First, let us consider how to deal with the simplest LR model by solving the linear elliptic PDE (10): $-\lambda\Delta u+(u-y)=0$. By using the RBF approximation (17) and the linearity of the Laplacian operator, the goal is reduced to finding a set of weights $\{w_i\}$:

$$\sum_i w_i(\phi_i-\lambda\Delta\phi_i)=y.$$

Let $\mathbf{w}\doteq(w_1,w_2,...,w_n)^T$ and $\mathbf{y}\doteq(y_1,y_2,...,y_n)^T$, where n is the number of training samples. Then $\mathbf{w}$ can be solved by the system of linear equations:

$$\mathbf{A}\mathbf{w}=\mathbf{y},\;\; \mathbf{A}_{ij}=\phi_j(\mathbf{x}_i)-\lambda\Delta\phi_j(\mathbf{x}_i).$$

Numerically, the following regularized least squares solution is adopted in practice to avoid ill-posed problems:

$$\min_{\mathbf{w}} |\mathbf{A}\mathbf{w} - \mathbf{y}|^2 + \eta|\mathbf{w}|^2.$$

The closed-form solution is simply given by $\mathbf{w} = (\mathbf{A}^T\mathbf{A} + \eta\mathbf{I})^{-1}\mathbf{A}^T\mathbf{y}$ with fast computational speed. It is interesting to see that both classification and regression problems can be solved by fitting a set of linear equations. Naturally, the LR method can be regarded as a generalization of linear regression $\mathbf{X}\mathbf{w} = \mathbf{y}$ or ridge regression $\min_{\mathbf{w}} |\mathbf{X}\mathbf{w} - \mathbf{y}|^2 + \eta|\mathbf{w}|^2$ (Hastie et al., 2009, chap. 3), where the original data matrix $\mathbf{X}$ is replaced by a "new" data matrix $\mathbf{A}(\mathbf{X})$ in the LR model.

4.3 Algorithm for TV and EE Models

As the TV model is one degenerate case of the EE model, we describe solutions for the more complicated EE model in this section. Here two algorithms are developed to tackle the nonlinearity in (15): (1) gradient descent time marching, and (2) lagged linear equation iteration.

4.3.1 Gradient Descent Time Marching

A standard solution is the steepest gradient descent marching with an artificial time t:

$$\frac{\partial u(\mathbf{x}, t)}{\partial t} = -\frac{\partial E_{TV}}{\partial u} = \lambda\nabla \cdot \left(\frac{\nabla u}{|\nabla u|}\right) - (u - y) \tag{24}$$

for the total variation PDE (12) and

$$\frac{\partial u(\mathbf{x}, t)}{\partial t} = -\frac{\partial E_{EE}}{\partial u} = \lambda\nabla \cdot \mathbf{V} - (u - y) \tag{25}$$

for the elastica PDE (15). Note that by setting $u_t = -E_u$, the energy functional E should decrease in the gradient direction as time marching. Here the partial derivative E_u can be obtained from the first variation of E (see Appendix A).

For image processing tasks, these gradient descent flows can be processed on a natural regular grid of the image domain. For high dimensional data space, such computational process is prohibitive. With the function approximation (17), a more practical way is handling the gradient descent flow about the weight vector $\mathbf{w}$. Consider a matrix form of the function approximation (17) on all training data points:

$$\mathbf{u} \doteq \begin{pmatrix} u(\mathbf{x}_1) \\ \vdots \\ u(\mathbf{x}_n) \end{pmatrix} = \Psi\mathbf{w},\ \Psi_{ij} \doteq \phi_j(\mathbf{x}_i).$$

Thus we have the gradient descent flow about $\mathbf{w}$:

$$\frac{\partial \mathbf{w}}{\partial t} = \Psi^{-1}\frac{\partial \mathbf{u}}{\partial t} = \Psi^{-1}\begin{pmatrix} \frac{\partial u}{\partial t}|_{\mathbf{x}=\mathbf{x}_1} \\ \vdots \\ \frac{\partial u}{\partial t}|_{\mathbf{x}=\mathbf{x}_n} \end{pmatrix}.$$

Then in each iteration the weight vector $\mathbf{w}$ can be updated by

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} + \tau \Psi^{-1} \begin{pmatrix} \frac{\partial u^{(k)}}{\partial t}|_{\mathbf{x}=\mathbf{x}_1} \\ \vdots \\ \frac{\partial u^{(k)}}{\partial t}|_{\mathbf{x}=\mathbf{x}_n} \end{pmatrix},$$

where τ is a small time step. We first initialize the weight vector $\mathbf{w}$ as $\mathbf{w}^{(0)} = (\Psi^T\Psi + \eta\mathbf{I})^{-1}\Psi^T\mathbf{y}$ by solving the regularized least squares problem $\Psi\mathbf{w} = \mathbf{y}$, with η a regularization parameter. Then we get $u^{(0)} = \Psi\mathbf{w}^{(0)}$, and run the iteration by computing $\mathbf{w}^{(k+1)}$ and $u^{(k+1)}$ alternately.

Here we give some details about the computation of the partial derivatives. Clearly the partial u_t in (24) can be obtained by (23). By omitting the third and higher order terms, $\nabla \cdot \mathbf{V}$ can be expanded into the following expression (see Appendix D):

$$\nabla \cdot \mathbf{V} = \kappa + b\kappa^3 - \frac{2b(\Delta u)^2}{|\nabla u|^5}\alpha + 6b\Big(\frac{\Delta u}{|\nabla u|^7} - \frac{\kappa}{|\nabla u|^6}\Big)\alpha^2 + \frac{6b}{|\nabla u|^7}\alpha\beta + \frac{2b}{|\nabla u|^5}\gamma, \tag{26}$$

where

$$\alpha \doteq \nabla u^T \mathbf{H}(u)\nabla u, \quad \beta \doteq \nabla u^T \mathbf{H}(u)^2 \nabla u, \quad \gamma \doteq \nabla u^T \mathbf{H}(u)^3 \nabla u.$$

We can see that if by setting $b = 0$, the expression of $\nabla \cdot \mathbf{V}$ is degenerated to $\kappa = \nabla \cdot (\nabla u / |\nabla u|)$, which is exactly the same expression of the TV model.

The time complexity in each iteration is $O(n^2 d)$, where n is the number of data points and d is the dimension. There are 3 parameters in the algorithm: the RBF parameter c, the regularization parameter λ, and the elastica weight parameter b. Note that we always set $a = 1$ since a can be absorbed into λ.

4.3.2 Lagged Linear Equation Iteration

Following the spirit of the lagged diffusivity fixed-point iteration method (Chan and Shen, 2005), we develop the following lagged linear equation iteration method. Empirically, the original lagged diffusivity fixed-point iteration often yields poor performance due to its brute-force linearization on the nonlinear PDE.

For the simpler TV model, by expanding the curvature term with (23) we have

$$-\frac{\lambda}{|\nabla u|}\Big(\Delta u - \frac{\nabla u^T H(u) \nabla u}{\nabla u^T \nabla u}\Big) + (u - y) = 0,$$

or equivalently by the RBF approximation

$$-\lambda\Big\{\sum_i w_i(1 - d + c|\mathbf{x} - \mathbf{x}_i|^2)\phi_i - c\frac{\mathbf{g}^T\Phi\mathbf{g}}{\mathbf{g}^T\mathbf{g}}\Big\} + |\mathbf{g}|\Big\{\Big(\sum_i w_i\phi_i\Big) - y\Big\} = 0.$$

The above nonlinear equation about $\mathbf{w}$ is rather complex as $\mathbf{g}$ and Φ contain the unknown $\mathbf{w}$. To simplify this equation, we use an iteration method that computes $\mathbf{w}$ or $\mathbf{g}$ alternately by fixing the other variables. First, $\mathbf{w}$ is initialized as a random vector. Then $\mathbf{g}$ can be computed according to (20). Now assuming that $\mathbf{g}$ is fixed, we have

$$\begin{aligned} \frac{\mathbf{g}^T\Phi\mathbf{g}}{\mathbf{g}^T\mathbf{g}} &= \frac{\mathbf{g}^T[\sum_i w_i(\mathbf{x}-\mathbf{x}_i)(\mathbf{x}-\mathbf{x}_i)^T\phi_i]\mathbf{g}}{\mathbf{g}^T\mathbf{g}} \\ &= \sum_i w_i\phi_i\Big(\frac{\mathbf{g}^T(\mathbf{x}-\mathbf{x}_i)(\mathbf{x}-\mathbf{x}_i)^T\mathbf{g}}{\mathbf{g}^T\mathbf{g}}\Big). \end{aligned}$$

Thus the original nonlinear equation about $\mathbf{w}$ becomes a linear equation

$$\sum_i w_i\Big(\frac{|\mathbf{g}|}{\lambda}-h\Big)\phi_i = \frac{|\mathbf{g}|}{\lambda}y,$$

where

$$h \doteq 1-d+c|\mathbf{x}-\mathbf{x}_i|^2 - c\frac{\mathbf{g}^T(\mathbf{x}-\mathbf{x}_i)(\mathbf{x}-\mathbf{x}_i)^T\mathbf{g}}{\mathbf{g}^T\mathbf{g}}.$$

Using the lagged idea, we obtain the method of lagged linear equation iteration: (1) By fixing $\mathbf{g}$, solve the system of linear equations with respect to $\mathbf{w}$ to get a new $\mathbf{w}$; (2) Compute $\mathbf{g}$ with the updated $\mathbf{w}$; (3) Iterate until convergence or reaching maximal iteration number.

For the more complicated EE model, we have to simplify the corresponding PDE greatly. Following the lagged idea again, we first assume the term about curvature $K \doteq a+b\kappa^2$ being fixed. Then K can be absorbed into λ, leading to the following linear equation in a similar way:

$$\sum_i w_i\Big(\frac{|\mathbf{g}|}{\lambda K}-f\Big)\phi_i = \frac{|\mathbf{g}|}{\lambda K}y.$$

Similarly, a two-step lagged iteration procedure can be developed for the EE model: (1) By fixing $\mathbf{g}$ and K, solve the linear system with respect to $\mathbf{w}$; (2) Compute $\mathbf{g}$ and K with the updated $\mathbf{w}$; (3) Iterate until convergence or reaching maximal iteration number. There are three parameters: c, λ, and the regularization parameter η (empirically chosen in experiments) in the least squares problems.

5. Theoretical Properties

In this section, we explore some theoretical analysis for elastica based supervised learning algorithms under the framework of statistical learning theory (SLT) (Vapnik, 1998; Bousquet et al., 2004; Boucheron et al., 2005; von Luxburg and Schölkopf, 2008). First we present the existence and uniqueness analysis of our TV/EE solutions. Then we prove that elastica based classifiers are universally consistent, mainly based on the pioneering work of Steinwart (2005) for SVM and other regularized kernel classifiers.

5.1 Existence and Uniqueness of TV

We first consider the TV model (11), which is a special yet useful case of the elastica model (14). It is well-known that one can carry out the existence and uniqueness analysis for TV model in image processing tasks. Thanks to the fact that most properties of a BV function are independent of the data dimension, the following proof in $\mathbb{R}^d$ is a trivial but detailed extension of the overly simplified proof for the TV-based image denoising model in (Chan and Shen, 2005, Theorem 4.14 in chap. 4).

Before giving the theorem on existence and uniqueness, we first review several major properties of BV functions (Chan and Shen, 2005, Section 2.2.2) (Aubert and Kornprobst, 2006, Section 2.2.3) that are frequently used in the following proofs.

Theorem 1 *(1)* (Completeness) $\mathrm{BV}(\Omega) \subset L^1(\Omega)$ *is a Banach space under the BV norm*

$$\|u\|_{BV} \doteq \int_\Omega (|u| + |\nabla u|) d\mathbf{x}.$$

(2) (Weak Compactness) *Let* $\{u_n\}$ *be a bounded sequence in* $\mathrm{BV}(\Omega)$ *where* Ω *is a Lipschitz domain. There must exist a subsequence which converges in* $L^1(\Omega)$*.*
(3) (L^1-Lower Semicontinuity) *Suppose a sequence* $\{u_n\}$ *converges to* u *in* $L^1(\Omega)$*. Then*

$$\int_\Omega |\nabla u| d\mathbf{x} \leq \liminf_n \int_\Omega |\nabla u_n| d\mathbf{x}.$$

In particular if $\{u_n\}$ *is a bounded sequence in* $\mathrm{BV}(\Omega)$*, then* u *belongs to* $\mathrm{BV}(\Omega)$ *as well.*

Theorem 2 (Existence and Uniqueness of TV) *Under the assumption that the given target function* $y(\mathbf{x}) \in L^2(\Omega)$ *with* $\mathbf{x} \in R^d$*, the minimization problem*

$$E_{TV}[u] = \int_\Omega \Big(\frac{1}{2}(u-y)^2 + \lambda|\nabla u|\Big) d\mathbf{x}$$

admits a unique solution $\hat{u}(\mathbf{x}) \in \mathrm{BV}(\Omega)$*.*

Proof We first show the existence. E_{TV} is finite for at least one BV function $\bar{u}(\mathbf{x}) \equiv \int_\Omega y(\mathbf{x}) d\mathbf{x}$, which is a constant function over Ω with $|\nabla \bar{u}| = 0$. Thus there exist some BV functions having finite E_{TV} values. Clearly 0 is a lower bound of these E_{TV} values. Hence this nonempty number set of all E_{TV} values with 0 as a lower bound must have an infimum denoted as $E_0 (\geq 0)$. Since E_0 is an infimum, we can select a sequence of BV functions $\{u_i\}$ with bounded E_{TV} values such that their E_{TV} values converges to E_0. Note that such sequence of $\{u_i\}$ must be bounded as well as in $\mathrm{BV}(\Omega)$ in terms of the BV norm, since the TV seminorm $\int_\Omega |\nabla u| d\mathbf{x}$ is contained in E_{TV} and $\mathrm{BV}(\Omega) \subset L^1(\Omega)$. According to the weak compactness of the BV space, for the bounded sequence $\{u_i\}$ in $\mathrm{BV}(\Omega)$, there must exist a subsequence indexed by $i(k), k = 1, 2, \ldots$, which converges in $L^1(\Omega)$. Due to the completeness of $L^1(\Omega)$, let $\hat{u} \in L^1(\Omega)$ be its limit. By the L^1-lower semicontinuity of the TV seminorm, we have

$$\int_\Omega |\nabla \hat{u}| d\mathbf{x} \leq \liminf_k \int_\Omega |\nabla u_{i_k}| d\mathbf{x}$$

and also $\hat{u} \in \mathrm{BV}(\Omega)$ since $\{u_i\}$ is a bounded sequence in $\mathrm{BV}(\Omega)$. Observe that E_{TV} is lower semicontinuous with respect to the $L^1(\Omega)$ topology because both of its components, the L^2 norm (the squared loss in E_{TV}) and the TV seminorm, are lower semicontinuous. That is,

$$E_{TV}[\hat{u}] \leq \liminf_k E_{TV}[u_{i_k}] = \inf_{u \in \mathrm{BV}(\Omega)} E_{TV}[u] = E_0,$$

indicating that there exists $\hat{u} \in \mathrm{BV}(\Omega)$ achieving the minimum point of E_{TV}.

The uniqueness follows directly from the strict convexity of E_{TV}. Thanks to the Minkowski inequality $\|f+g\|_{L^p} \leq \|f\|_{L^p} + \|g\|_{L^p}$, the TV seminorm is convex (but not strictly convex) given by

$$\int_\Omega |\nabla(\alpha u + (1-\alpha)v)| = \int_\Omega |\alpha \nabla u + (1-\alpha)\nabla v| \leq \alpha \int_\Omega |\nabla u| + (1-\alpha)\int_\Omega |\nabla v|,$$

where $\alpha \in [0,1]$. Apparently the L^2 norm $\int_\Omega (u-y)^2$ is strictly convex. Hence combining two components together, we have that E_{TV} is strictly convex. Therefore as the minimum point of E_{TV}, $\hat{u} \in \mathrm{BV}(\Omega)$ is unique. ■

In the image processing literature, there are some variants of the existence and uniqueness analysis for different TV models. Chan et al. (2002) discussed the existence of TV inpainting models in the cases of noise free and having noise, but the uniqueness is neglected. Aubert and Kornprobst (2006) present the existence and uniqueness analysis for the TV-based image restoration problem

$$\min E_{TV}[u] = \int_\Omega \Big(\frac{1}{2}(Ru-y)^2 + \lambda\phi(|\nabla u|)\Big)d\mathbf{x},$$

where R is a linear blurring operator and ϕ is a strictly convex and nondecreasing cost function.

5.2 Existence of EE

We now consider the more complicated elastica model. In Ambrosio and Masnou (2003), the authors proved that a relaxed version of elastica-based image inpainting has at least one solution in $\mathrm{BV}(\Omega)$. Here we give the existence proof of a discrete elastica model for binary classification, which is adapted from the elegant proof in Steinwart (2005) for SVMs and other regularized kernel classifiers. The existence is the first step to fulfill the consistency proof in the next subsection. But the solution to elastica model can be non-unique, due to the lack of convexity for this energy functional.

We begin with some preliminary notations. In the following, let $\overline{\mathbb{R}} = [-\infty, +\infty]$, $\mathbb{R}^+ = [0, +\infty)$, and $\overline{\mathbb{R}}^+ = [0, +\infty]$. A binary classifier is a rule that assigns to every *training set* $T = \{(x_1, y_1), \ldots, (x_n, y_n)\} \in (X \times Y)^n$ ($Y = \{-1, +1\}$ for binary problems) a measurable function $f : X \to \mathbb{R}$ with the final decision given by $\mathrm{sign} f(x)$. Similar to the gray scale constraint in image processing tasks, we assume that f takes values in a bounded interval (e.g. $[-2, 2]$) since f should approximate $y \in \{-1, +1\}$ and the classification decision is only rated with the sign of f. Sometimes we use a looser condition that $f \in L_\infty(X)$. For a given loss function $L(y, f(x))$, write a **cost function** $C(\alpha, t) \doteq \alpha L(1,t) + (1-\alpha)L(-1,t)$ for $\alpha \doteq P(Y=1|X=x) \in [0,1]$ and $t \in \overline{\mathbb{R}}$. For a fixed α, define $M(\alpha)$ and the corresponding t_α such that $M(\alpha) \doteq C(\alpha, t_\alpha) \doteq \min_t C(\alpha, t)$. We then give the basic condition on the loss function L in order to guarantee that the solution t_α minimizing $C(\alpha, t)$ tends to have the same sign as the Bayes decision rule.

Definition 3 *A continuous function $L(y, f(x))$ is called an* ***admissible*** *loss function if for every $\alpha \in [0,1]$ and $t_\alpha \in \overline{\mathbb{R}}$ we have $t_\alpha < 0$ if $\alpha < 1/2$ and $t_\alpha > 0$ if $\alpha > 1/2$.*

A similar concept called **classification-calibrated** can be found in Bartlett et al. (March 2006), requiring that an incorrect sign of t_α always leads to a strictly larger $M(\alpha)$. The classification-calibrated condition generalizes the requirement of an admissible loss that the minimizer of $C(\alpha, t)$ (if it exists) has the correct sign. The admissibility of L is necessary in order to develop universally consistent classifiers (Steinwart, 2005). In particular, the quadratic loss $L(y, f(x)) = (1 - yf(x))^2$ used in our classification models is admissible and classification-calibrated; other examples can be found in Steinwart (2005) and Bartlett et al. (March 2006). In the following we always assume that $L(y, f(x))$ is a margin-based admissible loss function which is continuous with respect to the margin $yf(x)$.

Definition 4 *Let $S(\lambda, t) : \mathbb{R}^+ \times \overline{\mathbb{R}}^+ \to \overline{\mathbb{R}}^+$ be an increasing function with respect to λ and t, which is continuous in 0 with respect to λ and unbounded with respect to t. Moreover, for all $\lambda > 0$ there exists a $t > 0$ such that $S(\lambda, t) < \infty$. We call $S(\lambda, t)$ a **regularization function** if for all $\lambda > 0$ and $s \in \mathbb{R}^+$ we have $S(\lambda, 0) = S(0, s) = 0$, and if for all $\lambda > 0$, $t \in \overline{\mathbb{R}}^+$, and for all sequences $\{t_n\} \subset \overline{\mathbb{R}}^+$ with $t_n \to t$ and $S(\lambda, t_n) < \infty$, we have $S(\lambda, t_n) \to S(\lambda, t)$.*

In our TV/EE models, $S(\lambda, t) = \lambda t^2$ clearly satisfies the requirements of a regularization function. This regularization function is a typical setting in several variants of SVMs (Steinwart, 2005), leaving the differences of these variants mainly on the loss functions.

Definition 5 *The **(0-1) risk** of a measurable function $f : X \to \mathbb{R}$ is defined by*

$$\begin{aligned} R_P(f) &\doteq P(\{(x, y) : \mathrm{sign} f(x) \neq y\}) \\ &= \mathbb{E}_{(x,y)\sim P} 1(y\, f(x)). \end{aligned}$$

The smallest achievable risk

$$R_P \doteq \inf\{R_P(f) : f : X \to \mathbb{R} \text{ measurable}\}$$

*is called the **Bayes risk** of P.*

Definition 6 *Given an admissible loss function L and a probability measure P, the **L-risk** of a measurable function $f : X \to \mathbb{R}$ is defined by*

$$\begin{aligned} R_{L,P}(f) &\doteq \mathbb{E}_{(x,y)\sim P} L(y, f(x)) \\ &= \int_{(x,y)\sim P} L(y, f(x)) P_X(dx) P_Y(dy) \\ &= \int_X C(P(Y = 1|X = x), f(x)) P_X(dx). \end{aligned}$$

*The smallest possible L-risk is denoted by $R_{L,P}$. Furthermore, given a regularization function S, the **regularized L-risk** is defined by*

$$R^{reg}_{L,P,\lambda}(f) \doteq S(\lambda, \|f\|_{EE}) + R_{L,P}(f)$$

for all $\lambda > 0$. Here $\|f\|^2_{EE} \doteq \int_X (1 + b\kappa^2)|\nabla f| dx$ is the Euler's elastica regularizer with a misused norm notation, and $\kappa = \nabla \cdot \left(\frac{\nabla f}{|\nabla f|}\right)$. If overlooking the curvature term, it degenerates to the TV seminorm $\|f\|^2_{TV} \doteq \int_X |\nabla f| dx$. If P is an empirical measure with respect to $T \in (X \times Y)^n$, we write $R_{L,T}(f)$ and $R^{reg}_{L,T,\lambda}(f)$, respectively.

Theorem 7 (Existence of EE) *For all Borel probability measures P on $X \times Y$ and all $\lambda > 0$, there always exists a function $f_{P,\lambda} \in \mathrm{BV}(X)$ minimizing the regularized L-risk $R^{reg}_{L,P,\lambda}(f)$. Moreover, for all such $f_{P,\lambda} \in \mathrm{BV}(X)$ we have $\|f_{P,\lambda}\|_{EE} \le \delta_\lambda$ where*

$$\delta_\lambda \doteq \sup\{t : S(\lambda, t) \le 2[L(1,0) + L(-1,0)]\}.$$

Proof The following proof is adapted from Steinwart (2005, Lemma 3.1), and the difference lies on $R^{reg}_{L,P,\lambda}(f)$ where the original RKHS norm $\|f\|_H$ for SVM is replaced by the pseudo-norm $\|f\|_{EE}$ for EE. The proof consists of the following five steps.

A. Clearly $R^{reg}_{L,P,\lambda}(f)$ is finite for the BV function $\bar{f}(\mathbf{x}) \equiv \mathbb{E}_{(x,y)\sim P}\, y$ or $\bar{f}(\mathbf{x}) \equiv 0$, which is a constant function over X with $|\nabla \bar{f}| = 0$. Thus there exist some BV functions having finite $R^{reg}_{L,P,\lambda}(f)$ values. For all $\varepsilon \in (0, L(1,0) + L(-1,0)]$, by the definition of an infimum we can select an function $f_\varepsilon \in L^1(X)$ with

$$R^{reg}_{L,P,\lambda}(f_\varepsilon) \le \inf_{f \in L^1(X)} R^{reg}_{L,P,\lambda}(f) + \varepsilon.$$

Now we have

$$\begin{aligned} \inf_{f \in L^1(X)} R^{reg}_{L,P,\lambda}(f) &\le R^{reg}_{L,P,\lambda}(f \equiv 0) \\ &= S(\lambda, \|f \equiv 0\|_{EE}) + R_{L,P}(f \equiv 0) \\ &= 0 + \mathbb{E}_{(x,y)\sim P} L(y, f(x) \equiv 0) \\ &= P(y = 1|x)L(1,0) + P(y = -1|x)L(-1,0) \\ &\le L(1,0) + L(-1,0), \end{aligned}$$

where S satisfies the condition $S(\lambda, 0) = 0$ in the second equality. Furthermore,

$$\begin{aligned} S(\lambda, \|f_\varepsilon\|_{EE}) &\le S(\lambda, \|f_\varepsilon\|_{EE}) + R_{L,P}(f_\varepsilon) = R^{reg}_{L,P,\lambda}(f_\varepsilon) \\ &\le \inf_{f \in L^1(X)} R^{reg}_{L,P,\lambda}(f) + \varepsilon \le 2[L(1,0) + L(-1,0)]. \end{aligned}$$

As $S(\lambda, t)$ is an increasing function with respect to t, we obtain the boundedness of $\|f_\varepsilon\|^2_{EE}$. Since $\|f\|^2_{TV} \le \|f\|^2_{EE}$, we also have the boundedness of $\|f_\varepsilon\|^2_{TV}$ and $f_\varepsilon \in \mathrm{BV}(X)$.

B. The *Bolzano-Weierstrass theorem* states that each bounded sequence in $\mathbb{R}^n$ has a convergent subsequence. In functional analysis, the *Eberlein-Smulian theorem* (Conway, 1990, Theorem 13.1 in chap. 5) states that three different kinds of weak compactness are equivalent in a Banach space. Particularly, we will use the sequential compactness property of a subset A in a Banach space: *Every sequence from A has a convergent subsequence whose limit is in A in the weak sense.* Recall that $\mathrm{BV}(X)$ is a Banach space. By the two theorems, there exist $f_{P,\lambda} \in \mathrm{BV}(X)$, a sequence $\{f_{\varepsilon_n}\} \in \mathrm{BV}(X)$, and two finite number $c_1, c_2 \in \mathbb{R}^+$ such that $\|f_{\varepsilon_n}\|_{EE} \to c_1$, $\|f_{\varepsilon_n}\|_{TV} \to c_2$, and $f_{\varepsilon_n} \to f_{P,\lambda}$ weakly. Note that the weak convergence implies that $f_{P,\lambda}$ is uniquely determined, $\|f_{P,\lambda}\|_{BV} \le \liminf_n \|f_{\varepsilon_n}\|_{BV}$, and $\|f_{P,\lambda}\|_{L^1} \le \liminf_n \|f_{\varepsilon_n}\|_{L^1}$ since $\mathrm{BV}(X) \subset L^1(X)$ (Yosida, 1999, Theorem 5 and 9 in Chapter V.1). In particular, by the weak compactness of the BV space, we further have that $\{f_{\varepsilon_n}\}$ converges to $f_{P,\lambda}$ in $L^1(X)$. Thus $yf_{\varepsilon_n}(x) \to yf_{P,\lambda}(x)$ since the margin is a linear functional of f. As L is continuous with respect the margin, we obtain $L(y, f_{\varepsilon_n}(x)) \to$

$L(y, f_{P,\lambda}(x))$ for all $(x,y) \in X \times Y$. Recall that $|L(y, f_{\varepsilon_n}(x))|$ is uniformly bounded by the boundedness assumption of $|f|$ and the continuity of L. Therefore, the *bounded convergence theorem* (as a special case of *Lebesgue dominated convergence theorem*) implies

$$\begin{aligned}
R_{L,P}(f_{\varepsilon_n}(x)) &= \int_{(x,y)\sim P} L(y, f_{\varepsilon_n}(x))P_X(dx)P_Y(dy) \\
&\to \int_{(x,y)\sim P} L(y, f_{P,\lambda}(x))P_X(dx)P_Y(dy) \\
&= R_{L,P}(f_{P,\lambda}(x)).
\end{aligned}$$

C. By $R_{L,P}(f_{\varepsilon_n}) \to R_{L,P}(f_{P,\lambda})$, for a fixed $\rho > 0$, there exists an index n_0 such that for all $n \geq n_0$ we have both $\varepsilon_n \leq \rho$ and $R_{L,P}(f_{P,\lambda}) - R_{L,P}(f_{\varepsilon_n}) \leq \rho$. In other words, we obtain the following inequalities

$$\begin{aligned}
S(\lambda, \|f_{\varepsilon_n}\|_{EE}) + R_{L,P}(f_{P,\lambda}) - \rho &\leq S(\lambda, \|f_{\varepsilon_n}\|_{EE}) + R_{L,P}(f_{\varepsilon_n}) = R^{reg}_{L,P,\lambda}(f_{\varepsilon_n}) \\
&\leq \inf_{f\in L^1(X)} R^{reg}_{L,P,\lambda}(f) + \varepsilon_n \\
&\leq R^{reg}_{L,P,\lambda}(f_{P,\lambda}) + \varepsilon_n \\
&= S(\lambda, \|f_{P,\lambda}\|_{EE}) + R_{L,P}(f_{P,\lambda}) + \varepsilon_n,
\end{aligned}$$

where the second inequality is based on the definition of f_{ε_n}. It implies that

$$S(\lambda, \|f_{\varepsilon_n}\|_{EE}) \leq S(\lambda, \|f_{P,\lambda}\|_{EE}) + \varepsilon_n + \rho \leq S(\lambda, \|f_{P,\lambda}\|_{EE}) + 2\rho.$$

On the other hand, we need to consider another inequality in the opposite direction. By the weak convergence we already have $\|f_{P,\lambda}\|_{BV} \leq \liminf_n \|f_{\varepsilon_n}\|_{BV}$ and $\|f_{P,\lambda}\|_{L^1} \leq \liminf_n \|f_{\varepsilon_n}\|_{L^1}$. However these two inequalities have nothing to do with $\|f\|_{EE}$. Thanks to the lower semicontinuity of the mean curvature's L^p norm, Leonardi and Masnou (2009, Theorem 4.4) proved that

$$\mathcal{F}_p(f) = \int_X |\nabla f|(1 + |\nabla \cdot \left(\frac{\nabla f}{|\nabla f|}\right)|^p)dx$$

is lower semicontinuous in the class of $C^2(\mathbb{R}^d)$ functions whenever $p \geq 1$ for $d = 2$ or $p \geq 2$ for $d \geq 3$. An earlier result (Ambrosio and Masnou, 2003, Theorem 6) required $p > d-1$ for $d \geq 2$. Of course the definition of $\mathcal{F}_p(f)$ is valid only for a certain class of smooth functions and we use the following relaxed functional (Ambrosio and Masnou, 2003; Leonardi and Masnou, 2009)

$$\overline{\mathcal{F}}_p(f) = \inf\{\liminf_{h\to\infty} \mathcal{F}_p(f_h) : f_h \to f \in L^1\}$$

to extend to the whole space $L^1(\mathbb{R}^d)$ (including $BV(X)$). We also have lower semicontinuity of $\overline{\mathcal{F}}_p(f)$ (Ambrosio and Masnou, 2003, Theorem 5) and $\overline{\mathcal{F}}_p(f) = \mathcal{F}_p(f)$ whenever $f \in C^2(X)$ (Leonardi and Masnou, 2009, Theorem 4.4). Immediately we obtain

$$\|f_{P,\lambda}\|_{EE} \leq \liminf_n \|f_{\varepsilon_n}\|_{EE}$$

and thus by the increasing property of $S(\lambda, t)$,

$$S(\lambda, \|f_{P,\lambda}\|_{EE}) \leq \lim_{n\to\infty} S(\lambda, \|f_{\varepsilon_n}\|_{EE}).$$

Combining the inequalities in two directions together yields

$$\lim_{n\to\infty} S(\lambda, \|f_{\varepsilon_n}\|_{EE}) = S(\lambda, \|f_{P,\lambda}\|_{EE}).$$

D. Combining $R_{L,P}(f_{\varepsilon_n}) \to R_{L,P}(f_{P,\lambda})$ with $S(\lambda, \|f_{\varepsilon_n}\|_{EE}) \to S(\lambda, \|f_{P,\lambda}\|_{EE})$, we have

$$R^{reg}_{L,P,\lambda}(f_{\varepsilon_n}) \to R^{reg}_{L,P,\lambda}(f_{P,\lambda}).$$

Because the definition of $\{f_{\varepsilon_n}\}$ indicates

$$R^{reg}_{L,P,\lambda}(f_{\varepsilon_n}) \to \inf_{f\in L^1(X)} R^{reg}_{L,P,\lambda}(f),$$

we have found a $f_{P,\lambda} \in BV(X) \subset L^1(X)$ such that

$$R^{reg}_{L,P,\lambda}(f_{P,\lambda}) = \inf_{f\in L^1(X)} R^{reg}_{L,P,\lambda}(f).$$

E. The second assertion $\|f_{P,\lambda}\|_{EE} \leq \delta_\lambda$ is obtained by the boundedness of f_ε in the first step. ■

5.3 Binary Classification Consistency

In classical statistics, a statistic $\hat{\theta}_n$ is a consistent estimator of a parameter θ based on a sample of size n if and only if for any $\varepsilon > 0$, $\lim_{n\to\infty} P(|\hat{\theta}_n - \theta| > \varepsilon) = 0$. In the same spirit, it is natural to request that a learning algorithm should eventually "converge" to an optimal solution when more and more training examples are presented. In the literature of machine learning, there exists two different types of consistency depending on the optimal solution that belongs to some particular function space or the space of all functions (von Luxburg and Schölkopf, 2008). The latter is often called *Bayes consistency* if the risk of a learned classifier converges to the risk of the Bayes optimal decision rule. It is well accepted that a good learning algorithm should satisfy this asymptotic property of consistency when the data size is sufficiently large.

The literature on the consistency analysis of learning algorithms can be roughly classified into following categories: (1) binary classification (Zhang, 2004a; Bartlett et al., March 2006), in particular for SVM (Steinwart, 2005), for Boosting (Bartlett and Traskin, 2007), and for random forests (Biau et al., 2008); (2) multi-class classification (Zhang, 2004b; Tewari and Bartlett, 2007; Glasmachers, 2010); (3) regression (Zakai and Ritov, 2009); (4) learning to rank (Cossock and Zhang, 2008; Xia et al., 2008; Duchi et al., 2010); (5) multi-label learning (Gao and Zhou, 2013). The work by Biau et al. (2008) showed that some popular classifiers, including Breiman's random forest classifier, are not consistent.

We first formalize the definitions of several kinds of consistency used in this section, following von Luxburg and Schölkopf (2008) and Steinwart (2005).

Definition 8 *A classifier f_n is said to be (Bayes)* ***consistent*** *with respect to a given probability measure P if the risk $R(f_n)$ converges in probability to the Bayes risk, that is for all $\varepsilon > 0$,*

$$P(R(f_n) - R(f^*) > \varepsilon) \to 0 \ \ \textit{as} \ \ n \to \infty$$

where $R(f) \doteq P(\{(x,y) : \mathrm{sign} f(x) \neq y\})$ is the risk of a classifier f and f^* denotes the Bayes classifier. Furthermore, f_n is said to be ***universally consistent*** *if it is consistent for all distributions P on $X \times Y$. It is called* ***strongly universally consistent*** *if such limiting property even holds almost surely (a.s.), that is*

$$P(\lim_{n\to\infty} R(f_n) = R(f^*)) = 1.$$

Note that the Bayes risk is the minimum that we can achieve in the space of all measurable functions, so we always have $R(f_n) \geq R(f^*)$ and there is no need to use the absolute value as in classical statistics.

We also need the notion of simple functions to approximate any function from $L^p(X)$.

Definition 9 *A* ***simple function*** *is a function $\psi : X \to \mathbb{R}$ of the form*

$$\psi(\mathbf{x}) = \sum_{i=1}^{n} c_i \chi_{A_i}(\mathbf{x})$$

where χ_A is the indicator function of the set A and $\{c_i\} \subset \mathbb{R}$. Another description of a simple function is a function that takes on finitely many values in its range.

Proposition 10 (From Regularized to Unregularized) *For every Borel probability measure P on $X \times Y$, we have*

$$\lim_{\lambda\to 0} R^{reg}_{L,P,\lambda}(f_{P,\lambda}) = R_{L,P}$$

where $f_{P,\lambda} \in \mathrm{BV}(X)$ minimizes the regularized L-risk $R^{reg}_{L,P,\lambda}(f)$, and $R_{L,P}$ is the smallest possible L-risk $R_{L,P}(f)$ achieved by any measurable function $f : X \to \mathbb{R}$.

Proof First by the definition of $f_{P,\lambda}$ we have

$$\begin{aligned}
\lim_{\lambda\to 0} R^{reg}_{L,P,\lambda}(f_{P,\lambda}) &= \lim_{\lambda\to 0} \inf_{f\in \mathrm{BV}(X)} R^{reg}_{L,P,\lambda}(f) \\
&= \lim_{\lambda\to 0} \inf_{f\in \mathrm{BV}(X)} \{S(\lambda, \|f\|_{EE}) + R_{L,P}(f)\} \\
&= \inf_{f\in \mathrm{BV}(X)} \{\lim_{\lambda\to 0} S(\lambda, \|f\|_{EE}) + R_{L,P}(f)\} \\
&= \inf_{f\in \mathrm{BV}(X)} R_{L,P}(f)
\end{aligned}$$

since $S(\lambda, \cdot)$ is continuous in 0 with respect to λ and $S(0,\cdot) = 0$. Next we show that the following identities hold true

$$\inf_{f\in \mathrm{BV}(X)} R_{L,P}(f) = \inf_{f\in L^1(X)} R_{L,P}(f) = R_{L,P}$$

for a sequence of embedding spaces $\mathrm{BV}(X) \subset L^1(X) \subset \{f : X \to \overline{\mathbb{R}} \text{ measurable}\}$, which suffices to prove the assertion.

We first check the first identity. Recall that the simple functions that belong to $L^p(X)$ are *dense* in $L^p(X)$ for $1 \leq p \leq \infty$ (Hunter, 2011, Theorem 7.8). Note that an integrable simple function

$$\psi = \sum_{i=1}^{n} c_i \chi_{A_i}$$

belongs to $L^p(X)$ for $1 \leq p < \infty$ if and only if $\mu(A_i) < \infty$ for each $A_i \subset X$ such that $c_i \neq 0$, meaning that its support has finite μ measure. On the other hand, each simple function belongs to L^∞. We restrict the discussion on bounded functions in $L^p(X)$ since any unbounded $f \in L^p(X)$ can be replaced by a modified bounded $\tilde{f} \in L^p(X)$ to make the loss L smaller. Hence the nice property of density indicates that for every bounded $f \in L^p(X)$ ($1 \leq p \leq \infty$), there exists a sequence of simple functions g_n such that $\|f - g_n\|_{L^p} \to 0$ and $|g_n(x)| \leq |f(x)|$ pointwise. The strong convergence in L^p norm implies the weak convergence in measure

$$P_X(\{x \in X : |f - g_n| \geq \varepsilon\}) \to 0.$$

Since $L(y,t)$ is uniformly continuous with respect to the second variable in the closed interval $[-|f(x)|, |f(x)|]$, for any fixed y we have

$$P_X(\{x \in X : |L(y, f(x)) - L(y, g_n(x))| \geq \varepsilon\}) \to 0.$$

By the previous assumption that $L(y, f(x))$ is a margin-based admissible loss function which is continuous with respect to the margin $yf(x)$, there exists a function $\hat{L}(yf(x)) \in L^1(X)$ such that

$$|L(y, g_n(x))| \leq \hat{L}(yf(x)).$$

By the *Lebesgue's dominated convergence theorem*, the expectation in $R_{L,P}(f)$ and the limit can change order:

$$\begin{aligned}
\lim_{n\to\infty} \int_{(x,y)\sim P} L(y, g_n(x)) P_X(dx) P_Y(dy) &= \int_{(x,y)\sim P} L(y, f(x)) P_X(dx) P_Y(dy) \\
&= \mathbb{E}_{(x,y)\sim P} L(y, f(x)) \\
&= R_{L,P}(f).
\end{aligned}$$

Thus by fixing $p = 1$ we have

$$\inf\{R_{L,P}(f) : f \text{ simple}\} = \inf_{f \in L^1(X)} R_{L,P}(f).$$

Clearly such simple functions belong to $\mathrm{BV}(X)$, and also by the definition of BV functions we have $\mathrm{BV}(X) \subset L^1(X)$. Then the relation of embedding spaces implies that

$$\inf\{R_{L,P}(f) : f \text{ simple}\} \geq \inf_{f \in \mathrm{BV}(X)} R_{L,P}(f) \geq \inf_{f \in L^1(X)} R_{L,P}(f).$$

Together with the previous identity between simple functions and $L^1(X)$ functions, the first identity

$$\inf_{f \in \mathrm{BV}(X)} R_{L,P}(f) = \inf_{f \in L^1(X)} R_{L,P}(f)$$

follows.

The second identity comes from the fact

$$\inf_{f \in L^\infty(X)} R_{L,P}(f) = R_{L,P}$$

with the proof given by Steinwart (2005, Proposition 3.2). On the other hand, the embedding relationship $L^\infty(X) \subset L^1(X) \subset \{f : X \to \overline{\mathbb{R}} \text{ measurable}\}$ leads to

$$\inf_{f \in L^\infty(X)} R_{L,P}(f) \geq \inf_{f \in L^1} R_{L,P}(f) \geq R_{L,P}.$$

Therefore the second identity

$$\inf_{f \in L^1} R_{L,P}(f) = R_{L,P}$$

holds true. ■

Following the framework of consistency proof in Steinwart (2005), we need the final piece of the puzzle by showing that some suitable concentration inequalities hold true for our proposed algorithms. These concentration inequalities bridge the gap between the expected L-risk of $f_{P,\lambda}$ and the empirical L-risk of $f_{P,\lambda}$. Steinwart's framework is somehow modular: each tuple of concentration inequality, loss function, and function space gives a condition on $\{\lambda_n\}$ ensuring $|R_{L,P}(f_{P,\lambda}) - R_{L,T}(f_{P,\lambda})| \to 0$, and each different combination of this tuple leads to new consistency results. There exist several concentration inequalities in Steinwart (2005) based on covering numbers, localized covering numbers, and algorithmic stability. Among these three concentration inequalities, the algorithmic stability (Bousquet and Elisseeff, 2002; Kutin and Niyogi, 2002; Poggio et al., 2004) is an elegant approach that does not depend on any complexity measure of the underlying hypothesis space, but rather depend on how the learning algorithm searches this space. However, stability based concentration inequalities (Bousquet and Elisseeff, 2002) heavily rely on the reproducing property of the RKHS space and often require that the regularization term is convex, while these conditions do not hold for our elastica based learning algorithm. In the following we give a concentration inequality based on covering numbers.

For a metric space (M,d) we define its *covering number* $\mathcal{N}((M,d),\varepsilon)$ to be the minimal l such that there exist l disks in M with radius ε covering M:

$$\mathcal{N}((M,d),\varepsilon) \doteq \min\Big\{\, l \in \mathbb{N} \;:\; \{x_1,\ldots,x_l\} \subset M,\ M \subset \bigcup_{i=1}^{l} B(x_i,\varepsilon)\Big\},$$

where $B(x,\varepsilon)$ denotes the closed ball with center x and radius $\varepsilon \geq 0$. We also have to measure the continuity of a given loss function L. The *modulus of continuity* of L is defined by

$$\omega(L,\delta) \doteq \sup\{|L(y,t) - L(y,t')| \;:\; y \in Y,\ t,t' \in \mathbb{R},\ |t-t'| \leq \delta\}.$$

In addition we define the *inverted modulus of continuity* as

$$\omega^{-1}(L,\varepsilon) \doteq \sup\{\delta > 0 :\ \omega(L,\delta) \leq \varepsilon\}.$$

Moreover, since only $f_{P,\lambda} \in \mathrm{BV}(X)$ and $f_{T,\lambda} \in \mathrm{BV}(X)$ are our focus considered in the consistency results, we define the *restricted loss function*:

$$L_\lambda(\cdot,\cdot) \doteq L(y,f(x)) :\ y \in Y,\ f \in \mathrm{BV}(X) \cap L^\infty(X),\ \|f\|_{TV} \leq \delta_\lambda\,,$$

where δ_λ given in Theorem 7 is a simple upper bound on the TV semi-norm of the solutions of $R^{reg}_{L,P,\lambda}(f)$.

Lemma 11 (Concentration) *For all Borel probability measures P on $X \times Y$, all $\varepsilon > 0$, $\lambda > 0$, and all $n \geq 1$ we have*

$$P(|R_{L,T}(f_{T,\lambda}) - R_{L,P}(f_{T,\lambda})| \geq \varepsilon) \;\leq\; 2\mathcal{N}\left(\delta_\lambda I,\, \omega^{-1}(L_\lambda, \varepsilon/3)\right) \exp\Big(- \frac{2n\varepsilon^2}{9\|L_\lambda\|_\infty^2}\Big),$$

where $\delta_\lambda I \doteq \{f \in \mathrm{BV}(X) \cap L^\infty(X) : \|f\|_{TV} \leq \delta_\lambda\}$ is a metric space equipped with the $\|\cdot\|_\infty$ norm.

Proof Write the loss class as $\mathcal{F} \doteq \{L(\cdot, f(\cdot)) : f \in \mathrm{BV}(X) \cap L^\infty(X),\ \|f\|_{TV} \leq \delta_\lambda\}$. Note that $\mathcal{F}$ is a subset of $C(X \times Y)$ of nonnegative functions that are bounded by $\|L_\lambda\|_\infty$. Let $l = \mathcal{N}(\mathcal{F}, \varepsilon/3)$ and consider $f_1, \ldots, f_l$ such that the disks D_j centered at f_j and with radius $\varepsilon/3$ cover F. Recall that *Hoeffding's inequality* (Bousquet et al., 2004, Theorem 1) (see also the book by Boucheron et al., 2013), perhaps the most elegant quantitative version of the law of large numbers, states that for all $\varepsilon > 0$,

$$P\Big(\Big|\frac{1}{n}\sum_{i=1}^{n} f(Z_i) - \mathbb{E}[f(Z)]\Big| > \varepsilon\Big) \leq 2\exp\Big(- \frac{2n\varepsilon^2}{(b-a)^2}\Big),$$

where $Z_1, \ldots, Z_n$ be n i.i.d. random variables with $f(Z) \in [a, b]$. For each fixed f_j, applying Hoeffding's inequality yields

$$P(|R_{L,T}(f_j) - R_{L,P}(f_j)| \leq \varepsilon/3) \geq 1 - 2\exp\Big(- \frac{2n(\varepsilon/3)^2}{\|L_\lambda\|_\infty^2}\Big),$$

with $R_{L,P}(f_j) = \mathbb{E}_{(x,y)\sim P} L(y, f_j(x))$ and $L(y, f_j(x)) \in [0, \|L_\lambda\|_\infty]$. As the disks D_j are $\varepsilon/3$ cover of F, the following inequalities hold true

$$\begin{aligned} & \sup_{f \in Dj} |R_{L,T}(f) - R_{L,P}(f)| \\ = \;& \sup_{f \in Dj} |R_{L,T}(f) - R_{L,T}(f_j) + R_{L,T}(f_j) - R_{L,P}(f_j) + R_{L,P}(f_j) - R_{L,P}(f)| \\ \leq \;& \varepsilon/3 + |R_{L,T}(f_j) - R_{L,P}(f_j)| + \varepsilon/3 \\ \leq \;& \varepsilon, \end{aligned}$$

with probability at least $1 - 2\exp\big(- \frac{2n\varepsilon^2}{9\|L_\lambda\|_\infty^2}\big)$ over the random choice of the training set T. Since $\|f\|_{EE} \leq \delta_\lambda$ implies $\|f\|_{TV} \leq \delta_\lambda$, using the union bound we get

$$P(\sup_{\|f\|_{EE} \leq \delta_\lambda} |R_{L,T}(f) - R_{L,P}(f)| \geq \varepsilon) \leq 2\mathcal{N}(\mathcal{F}, \varepsilon/3) \exp\big(- \frac{2n\varepsilon^2}{9\|L_\lambda\|_\infty^2}\big).$$

By the definition of the modulus of continuity, every ε cover $f_1, \ldots, f_l$ with $\|f_j\|_{TV} \leq \delta_\lambda$ defines an $\omega(L_\lambda, \varepsilon)$ cover $L(\cdot, f_1(\cdot)), \ldots, L(\cdot, f_l(\cdot))$ of $\mathcal{F}$ with respect to the supremum norm. Thus we have

$$\mathcal{N}(\mathcal{F}, \varepsilon/3) \leq \mathcal{N}(\delta_\lambda I,\, \omega^{-1}(L_\lambda, \varepsilon/3)),$$

which immediately yields

$$P\Big(\sup_{f \in \delta_\lambda I} |R_{L,T}(f) - R_{L,P}(f)| \geq \varepsilon\Big) \leq 2\,\mathcal{N}(\delta_\lambda I, \omega^{-1}(L_\lambda, \varepsilon/3)) \exp\Big(- \frac{2n\varepsilon^2}{9\|L_\lambda\|_\infty^2}\Big).$$

Since Lemma 7 guarantees that $\|f_{P,\lambda}\|_{TV} \leq \delta_\lambda$ or $\|f_{T,\lambda}\|_{TV} \leq \delta_\lambda$, the assertion follows. ■

Theorem 12 (Universal Consistency) *The classifier $f_{T,\lambda_n} \in \mathrm{BV}(X)$ minimizing the regularized empirical L-risk $R^{reg}_{L,T,\lambda_n}(f)$ is universally consistent for a positive sequence $\{\lambda_n\}$ with $\lambda_n \to 0$ and*

$$\frac{1}{n}\|L_{\lambda_n}\|^2_\infty \ln \mathcal{N}(\delta_{\lambda_n} I, \omega^{-1}(L_{\lambda_n}, \varepsilon)) \to 0$$

for all $\varepsilon > 0$.

Proof The Proposition 3.3 of Steinwart (2005) states that for any Borel probability measure P on $X \times Y$ and for all $\varepsilon > 0$, there exists a $\delta > 0$ such that for all measurable $f : X \to \overline{\mathbb{R}}$ with $R_{L,P}(f) \leq R_{L,P} + \delta$ we have $R_P(f) \leq R_P + \varepsilon$. Here L in $R_{L,P}(f)$ requires to be an admissible loss function. Therefore, in order to prove the 0-1 risk $R_P(f_{T,\lambda_n}) \leq R_P + \varepsilon$, it suffices to show the L-risk $R_{L,P}(f_{T,\lambda_n}) \leq R_{L,P} + \delta$.

The outline is given as follows:

$$\begin{aligned}
R_{L,P}(f_{T,\lambda_n}) &\leq S(\lambda_n, \|f_{T,\lambda_n}\|_{EE}) + R_{L,P}(f_{T,\lambda_n}) \\
&\leq S(\lambda_n, \|f_{T,\lambda_n}\|_{EE}) + R_{L,T}(f_{T,\lambda_n}) + \delta/3 && (27)\\
&\leq S(\lambda_n, \|f_{P,\lambda_n}\|_{EE}) + R_{L,T}(f_{P,\lambda_n}) + \delta/3 && (28)\\
&\leq S(\lambda_n, \|f_{P,\lambda_n}\|_{EE}) + R_{L,P}(f_{P,\lambda_n}) + 2\delta/3 && (29)\\
&= R^{reg}_{L,P,\lambda_n}(f_{P,\lambda_n}) + 2\delta/3 \\
&\leq R_{L,P} + \delta. && (30)
\end{aligned}$$

Among the above inequalities, (27) and (29) hold true by the empirical concentration inequality in Lemma 11 with probability at least

$$1 - 2\mathcal{N}\left(\delta_\lambda I,\ \omega^{-1}(L_\lambda, \varepsilon/3)\right) \exp\left(- \frac{2n\varepsilon^2}{9\|L_\lambda\|^2_\infty}\right)$$

over the random choice of the training set T, while (28) is obtained by the fact that f_{T,λ_n} minimizes the regularized empirical L-risk $R^{reg}_{L,T,\lambda_n}(f)$. Proposition 10 with respect to $\lambda_n \to 0$ immediately implies (30): there exists an integer $n_0 \geq 1$ such that for all $n \geq n_0$ we have

$$|R^{reg}_{L,P,\lambda_n}(f_{P,\lambda_n}) - R_{L,P}| \leq \delta/3.$$

Note that the condition

$$\frac{1}{n}\|L_{\lambda_n}\|^2_\infty \ln \mathcal{N}(\delta_{\lambda_n} I, \omega^{-1}(L_{\lambda_n}, \varepsilon)) \to 0$$

assures that $R_{L,P}(f_{T,\lambda_n}) \leq R_{L,P} + \delta$ holds true with probability 1 nearly as $n \to \infty$. Then the universal consistency follows by $P(R_P(f_{T,\lambda_n}) - R_P \leq \varepsilon) \to 1$ for all distributions P on $X \times Y$. ■

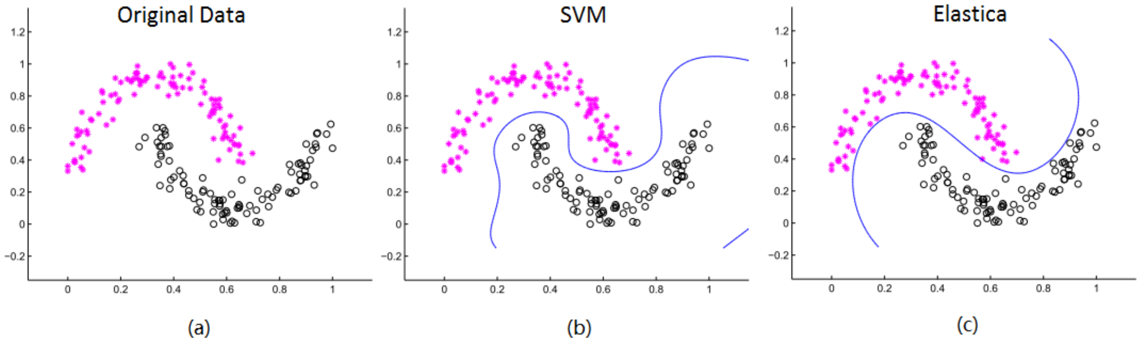


Figure 2: Decision boundaries produced by SVM and EE with common parameters on two moon data.

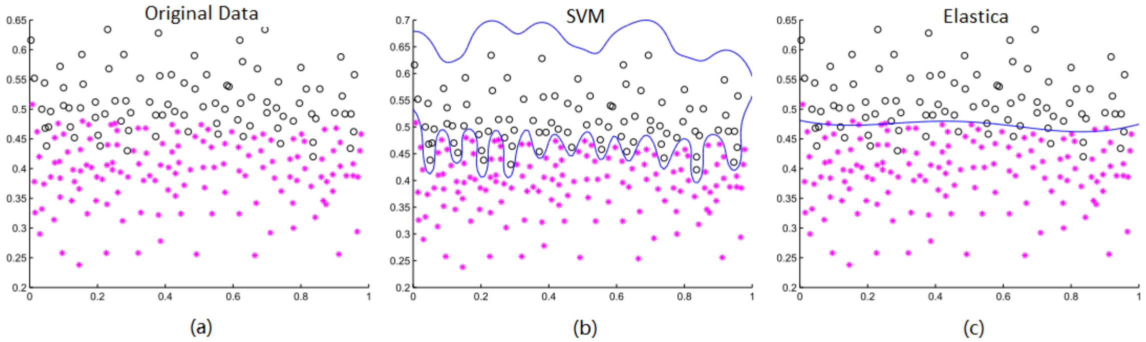
6. Experimental Results

The proposed two models (TV and EE) are compared with LR, SVM with RBF kernels using the LIBSVM implementation (Chang and Lin, 2011), and Back-Propagation Neural Networks (BPNN) in the Matlab neural network toolbox. Two implementations of our methods are also compared: Gradient Descent method (GD) and Lagged Linear Equation method (LAG). The maximum number of iterations in GD and LAG is empirically setting as 40. Binary classification, multi-class classification, and regression tasks are tested on synthetic and real-world data sets. We collected real data sets from the libsvm website (Chang and Lin, 2011) and the UCI machine learning repository (Asuncion and Newman, 2013). Some attributes have been removed due to missing entries. Some data sets have a huge number of instances, hence we use only 1000 instances in our experiments. All data sets are scaled into [0,1] before training and testing.

6.1 Synthetic Data

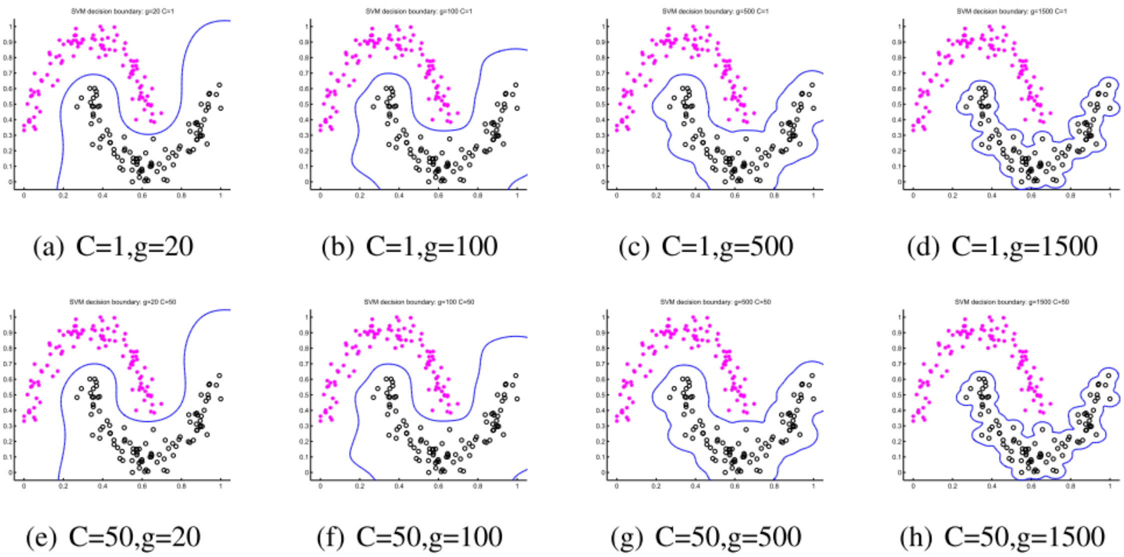
We first compare our EE model and SVM for binary classification on two synthetic data sets: the two moon data and one data set made by ourselves. Fig. 2 and Fig. 3 show the decision boundaries produced by SVM and EE with common parameters. We can see that SVM tends to yield curved or even wiggly decision boundaries to pursue low training errors. In contrast, smooth or even straight decision boundaries with low curvature are favored by EE, hence reducing the risk of overfitting.

One may argue that SVM can produce smooth and low curvature decision boundaries by tuning the parameters. Fig. 4 shows the results of SVM with different combinations of kernel parameter g and slack parameter C. For comparison, Fig. 5 displays the results of EE with different combinations of regularization parameter λ and kernel parameter c. We can see that most decision boundaries produced by EE have lower curvature values and are smoother than the results by SVM. Actually the elastica term in EE may be interpreted as the accumulated bending energy of all level lines, including the level line on the decision boundary.



(a) (b) (c)

Figure 3: Decision boundaries produced by SVM and EE with common parameters on our synthetic data.



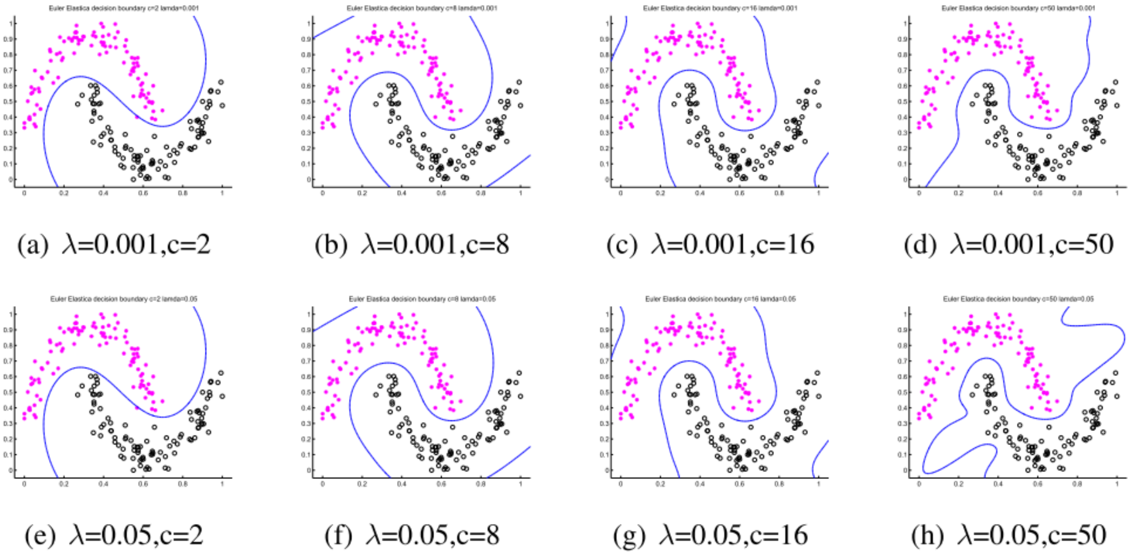
(a) C=1,g=20 (b) C=1,g=100 (c) C=1,g=500 (d) C=1,g=1500

(e) C=50,g=20 (f) C=50,g=100 (g) C=50,g=500 (h) C=50,g=1500

Figure 4: Decision boundaries produced by SVM with different parameter combinations on two moon data.

6.2 Binary Classification

We use eleven data sets for binary classification. The optimal parameters for each algorithm are selected by grid search using 5-fold cross-validation. To make the grid search more practical, only two common parameters are searched for all methods except BPNN: (C, g) for SVM, while (c, λ) for LR, TV, and EE. Empirically, the parameter η is set as 1 for LR, and the parameter b is fixed as 0.01 for EE. Then excluding BPNN, the two common parameters are searched from $-10 : 10$ in logarithm with step 2. For each data set, we randomly run the 5-fold cross validation ten times to reduce the influence of data partitions.



(a) λ=0.001,c=2 (b) λ=0.001,c=8 (c) λ=0.001,c=16 (d) λ=0.001,c=50

(e) λ=0.05,c=2 (f) λ=0.05,c=8 (g) λ=0.05,c=16 (h) λ=0.05,c=50

Figure 5: Decision boundaries produced by EE with different parameter combinations on two moon data.

Table 1 gives the average classification accuracies (with standard deviations) for the five methods. The results indicate that BPNN performs the worst, while the LAG version of EE achieves the best accuracies on six data sets. LR and other implementations of TV and EE are comparable with SVM. When comparing EE-LAG and SVM in a pairwise fashion, we can see that EE-LAG achieves improvements over SVM on 10 datasets (though not much statistically significant as the differences on two averaged accuracies is often less than one standard deviation).

6.3 Multi-Class Classification

For multi-class tasks, we collected twelve data sets. For the 256-dimensional USPS data, PCA is used as a preprocessing step to reduce the dimension to 30 and we randomly select 1000 samples for experiments. Same as the settings for binary problems, we use ten runs 5-fold cross-validation to choose the optimal parameters for each method. All methods except for BPNN have two common parameters which are searched from $-10 : 10$ in logarithm with step 1.

Aside from BPNN that has a built-in ability for multi-class tasks, almost all function learning approaches are originally designed for binary classification. In order to handle multi-class situations, usually “one versus all” (OVA) or “one versus one” (OVO) strategies can be adopted. If using OVA, one needs to learn M scoring functions to fulfill the multi-class task, where M is the number of classes. The final decision is the label whose scoring function achieves the largest value or confidence score. However, these scoring functions are learned independently, often suffering to the so-called *calibration problem* (Mohri et al., 2012, chap. 8). LIBSVM uses the OVO strategy, with some reasons and detailed comparisons given in (Hsu and Lin, 2002). See also Mohri et al. (2012, chap. 8) for dis-

Data	Dim	Num	SVM	BPNN	LR	TV		EE	
						GD	LAG	GD	LAG
Australian	14	690	85.94	85.34	87.06	87.11	87.01	86.54	**87.25**
			±2.70	±1.97	±2.45	±2.06	±2.46	±2.31	±2.01
Blood	4	748	79.01	79.08	79.32	79.55	79.42	**79.73**	**79.73**
transfusion			±3.01	±3.36	±3.74	±2.38	±2.60	±2.18	±2.03
Breast-	10	683	97.36	96.40	97.60	97.36	97.72	97.13	**97.83**
cancer			±1.59	±1.14	±1.27	±1.28	±1.43	±1.37	±1.29
Diabetes	8	768	77.73	76.85	77.96	77.83	77.81	**78.23**	78.10
			±3.03	±4.22	±3.50	±3.19	±2.73	±2.54	±2.63
German.	24	1000	77.10	76.37	77.10	76.19	77.10	76.50	**77.22**
number			±1.61	±1.61	±1.36	±1.47	±1.29	±1.59	±1.30
Haberman's	3	306	74.51	74.52	**75.77**	75.30	75.28	75.65	75.34
survival			±4.31	±3.53	±3.00	±3.31	±3.78	±3.42	±3.32
Heart	13	270	83.70	81.76	84.26	84.45	84.58	84.78	**84.96**
			±2.72	±3.16	±2.22	±2.82	±2.73	±2.69	±2.79
Liver-	6	345	73.62	71.52	73.20	**74.81**	73.62	74.32	73.91
disorders			±5.72	±4.44	±2.95	±2.49	±2.65	±2.29	±2.83
Planning	12	182	**73.63**	67.62	72.22	71.67	71.67	72.22	71.67
relax			±4.41	±4.93	±4.46	±4.93	±4.08	±4.25	±4.79
Sonar	60	208	89.90	88.99	**90.88**	90.30	90.27	90.07	90.50
			±4.41	±4.79	±3.83	±4.47	±4.72	±3.27	±3.37
Vertebral	6	310	85.81	85.16	84.52	84.55	84.75	85.83	**85.92**
column			±4.26	±3.12	±3.90	±4.14	±4.37	±3.38	±3.68

Table 1: Average accuracies (%) for binary classification with 5-fold cross-validation.

cussions between OVA and OVO. Recently in Varshney and Willsky (2010), an efficient binary encoding strategy was proposed to represent the decision boundary by using only $m = \lceil log_2 M \rceil$ functions. Empirically we compared the $log_2 M$ strategy and the OVA strategy for LR, TV and EE, and found that the in most cases the $log_2 M$ strategy performs slightly better. As the codewords for making decisions are represented as 0-1 bits of length m, the $log_2 M$ strategy may somehow "favor" those methods with good function approximation ability. In multi-class experiments, the $log_2 M$ strategy is used for LR, TV and EE, while LIBSVM runs with the OVO strategy.

The multi-class results of classification accuracies are shown in Table 2. The accuracy results demonstrate that both SVM and EE-GD offer the best accuracies on four (different) data sets, and both EE-LAG and TV-GD take the first place on two (different) data sets. If we compare SVM and EE-GD in a pairwise fashion by excluding other competing methods, the results show that SVM wins on only five data sets while EE-GD performs better on the other seven data sets. Therefore on multi-class tasks, Table 2 implies that our EE-GD version can offer competitive results, or can perform slightly better than SVM.

6.4 Regression

We use ten regression data sets to validate the proposed TV/EE methods compared with SVM, BPNN, and LR. All data sets are scaled into [0,1]. The same experimental settings

Data	Cls	Dim	Num	SVM	BPNN	LR	TV		EE	
							GD	LAG	GD	LAG
Balance scale	3	4	625	**98.40** ±1.13	92.48 ±1.77	89.44 ±1.90	90.88 ±1.25	89.92 ±1.36	90.40 ±1.80	91.36 ±1.35
Flags	8	29	194	52.06 ±7.57	46.90 ±7.25	53.13 ±7.34	51.50 ±7.29	52.10 ±7.10	**53.55** ±6.39	52.10 ±7.22
Glass	6	9	214	73.83 ±9.22	63.99 ±11.83	73.81 ±7.34	75.59 ±8.73	**76.19** ±8.62	75.82 ±9.07	75.71 ±9.15
Hayes-rath	3	5	132	**81.82** ±4.12	74.26 ±4.62	73.63 ±4.31	77.87 ±4.59	77.08 ±4.67	78.90 ±4.29	78.15 ±4.31
Iris	3	4	150	**96.67** ±3.65	96.00 ±3.37	95.33 ±3.42	96.00 ±3.50	96.00 ±3.27	96.00 ±3.33	96.00 ±3.10
Statlog imageseg	7	19	2310	97.27 ±0.91	96.74 ±1.12	97.31 ±0.95	97.31 ±0.93	97.21 ±0.88	**97.45** ±0.81	97.44 ±0.83
Seeds	3	7	210	94.76 ±1.78	**95.71** ±1.56	92.38 ±1.85	92.86 ±1.74	92.65 ±1.62	92.86 ±1.93	92.75 ±1.87
Teaching assist	3	5	151	60.93 ±20.97	56.63 ±19.44	63.47 ±17.28	65.18 ±14.26	66.00 ±15.37	65.41 ±16.23	**67.33** ±17.41
USPS	10	30	1000	94.10 ±1.39	89.72 ±2.79	94.90 ±1.28	94.40 ±1.32	94.80 ±1.73	94.40 ±1.54	**95.00** ±1.27
Vehicle	4	18	846	84.40 ±0.70	79.18 ±1.41	82.75 ±1.33	**85.00** ±0.82	84.25 ±0.93	**85.00** ±0.78	84.84 ±0.90
Wine	3	13	178	98.88 ±1.27	97.78 ±1.43	**99.44** ±0.83	**99.44** ±0.83	99.43 ±0.85	**99.44** ±0.83	98.86 ±1.31
Yeast	10	8	1484	**60.78** ±3.26	54.49 ±4.57	58.22 ±3.79	57.95 ±3.34	57.91 ±3.27	57.95 ±3.64	57.97 ±3.52

Table 2: Average accuracies (%) for multi-class classification with 5-fold cross-validation.

are repeated by running ten times of 5-fold cross-validation for each data set. Table 3 shows the regression results in mean square errors (MSE) with standard deviations.

Clearly, we can see that TV-LAG and two versions of EE achieve the best regression results, with each winning three times on overall ten data sets. BPNN yields the lowest errors on two data sets. Surprisingly SVM takes the first place on only one data set. If we select SVM and LR in a pairwise fashion by excluding other methods, we find that LR offers lower errors on seven data sets while SVM performs better on only other three data sets. If we compare SVM and TV-GD separately by neglecting other methods, TV-GD performs better on nine data sets. Note that TV-GD performs the worst among all versions of TV/EE. These results demonstrate that compared with other competing methods, the performance of SVM on regression tasks is rather unsatisfactory. The reason might be that the original purpose of SVM is designed for classification, not for regression. In contrast, our TV/EE methods exhibit excellent regression ability on these data sets.

6.5 Running Times

To compare the real performance in computational burdens, in Table 4 we list the running times of the competing methods on five data sets for binary classification. The running times are obtained for five-fold cross-validation in one single round, averaged by ten rounds. The experiments are conducted on a PC Sever with two Intel Xeon 5620 cores and 8GB RAM.

Data	Dim	Num	SVM	BPNN	LR	TV		EE	
						GD	LAG	GD	LAG
Auto MPG	7	392	7.11	5.63	6.07	5.67	5.62	**5.47**	5.69
			±0.56	±0.57	±0.55	±0.56	±0.53	±0.51	±0.56
Concrete	8	1030	6.42	**4.88**	6.02	5.98	5.43	5.83	5.24
comp. str.			±0.62	±0.56	±0.64	±0.83	±0.60	±0.78	±0.61
Concrete	9	103	4.48	14.30	5.01	1.86	**1.61**	1.76	**1.61**
slump test			±2.00	±7.14	±1.70	±0.81	±0.70	±0.71	±0.70
Forest	12	517	5.95	6.20	3.41	3.43	3.41	**3.37**	3.41
fires			±3.62	±3.73	±3.69	±3.61	±3.69	±3.54	±3.69
Housing	13	506	5.88	7.54	5.13	4.92	**4.90**	5.14	4.95
			±2.28	±2.41	±2.36	±2.47	±2.29	±2.39	±2.33
Machine	6	209	3.32	5.18	1.78	2.37	1.91	**1.75**	**1.75**
CPU			±2.81	±3.23	±1.70	±1.80	±1.78	±1.72	±1.72
Pyrim	27	74	9.32	23.06	6.59	**5.81**	5.89	5.90	5.93
			±9.75	±9.97	±6.22	±5.17	±5.85	±6.09	±6.12
Servo	4	167	9.93	**5.62**	7.29	8.81	8.34	8.87	7.86
			±5.09	±5.24	±5.80	±5.49	±5.63	±5.29	±5.83
Triazines	60	186	**19.24**	41.90	20.73	19.67	20.32	19.63	19.95
			±6.61	±8.71	±4.46	±3.93	±4.08	±2.50	±2.79
Yacht	6	308	4.52	3.70	7.75	2.33	**1.45**	2.07	**1.45**
hydrodynamics			±0.31	±0.29	±1.91	±0.47	±0.32	±0.43	±0.32

Table 3: Regression errors measured by MSE (10^{-3}) with 5-fold cross-validation.

We can see that the computational burdens of TV/EE algorithms is similar to that of BPNN in Matlab toolbox, but much slower than LIBSVM. The computational PDE approach of our TV/EE models is implemented by gradient descent or lagged iteration, which often requires a long time for assuring that the iterations converge. In each iteration, all the data points participate in the computations of our methods within the current implementations. In contrast, the solutions of SVM is essentially sparse, and recent several improvements show that carefully selecting a small representative subset of the training data can further greatly speed-up the optimization process of SVM (Nandan et al., 2014; Wang et al., 2014).

Our intention in this paper is not to develop a fully-fledged and highly optimized algorithm for supervised learning problems. Instead, this work only serves as a starting point for applying Euler's elastica to classification and regression tasks. The above experiments have demonstrated the excellent accuracies of our elastica based algorithms, though the numerical solutions are rather slow. Hence there exists an opportunity to dramatically improve the computational efficiency by considering the following techniques: (1) Some first order numerical methods, like the augmented Lagrangian method (ALM). The operator splitting method and ALM have been successfully implemented to solve Euler's elastica model for image applications (Tai et al., 2011; Hahn et al., 2011; Duan et al., 2013). The speed-up is spectacular compared with prior approaches. Interestingly the ALM has been also applied to optimize the primal SVM problem with linear computational cost (Nie et al., 2014). (2) Imposing the sparsity constraint on the coefficients **w**. The sparsity property may enhance the efficiency in each iteration. (3) Selecting a small representative subset of the training data in a similar way proposed by Nandan et al. (2014) and Wang et al. (2014).

Data	Dim	Num	SVM	BPNN	LR	TV		EE	
						GD	LAG	GD	LAG
Australian	14	690	0.859	30.673	4.734	24.734	32.453	25.734	33.197
Blood transfusion	4	748	0.297	22.247	5.938	28.467	35.746	27.481	36.497
Breast-cancer	10	683	0.453	20.318	4.609	18.953	19.447	19.732	20.278
Diabetes	8	768	0.547	23.142	6.453	20.120	20.981	22.145	21.519
German.number	24	1000	1.266	31.452	14.156	29.266	32.145	34.497	31.876

Table 4: Running times (in seconds) for binary classification with 5-fold cross-validation in one single round.

7. Conclusion

Regularization framework and function learning approaches have become very popular in the recent machine learning literature. Due to the great success of total variation and Euler's elastica models in image processing area, we extend these two models for supervised classification and regression on high dimensional data sets. The TV regularizer permits steeper edges near the decision boundaries, while the elastica smoothing term penalizes non-smooth level set hypersurfaces of the target function. Compared with SVM and BPNN, our proposed methods have demonstrated the competitive performance on commonly used benchmark data sets. Specifically, TV and EE offer superb results on binary classification and regression tasks, and performs slightly better than SVM on multiclass problems. In comparison, SVM often yields excellent accuracies for multi-class classification, but offer poor results on regression problems.

Our future work is to explore other possibilities in using different basis functions and to speedup the training time. Recently, several fast Augmented Lagrangian Methods (ALM) (Tai et al., 2011; Duan et al., 2013) have been applied to solve Euler's elastica models in image denoising, inpainting, and zooming applications. Particularly in Duan et al. (2013), the Euler's elastica functional is reformulated as a serial of subproblems, which can be efficiently solved by either closed-form solution or fast iteration method. Whether these methods can be extended to high dimensional problems needs further investigations. Another interesting direction is to extend the work of Zakai and Ritov (2009) on regression consistency to the TV and EE models.

Acknowledgments

The authors thank the anonymous reviewers for valuable comments and insightful suggestions. This work was supported by the National Basic Research Program of China (973 Program) 2011CB302202, the Natural Science Foundation of China (NSFC Grants 61075119 and 61375051), and the Seeding Grant for Medicine and Information Sciences of Peking University (2014-MI-21).

Appendix A: Curvature

The following material comes from Aubert and Kornprobst (2006, chap. 2.4) with slightly different notations. Readers are also referred to the classical geometry book of do Carmo (1976).

Let $\mathbf{c}(p) = (x(p), y(p))$ be a regular planar oriented curve on R^2 with parameter $p \in [0, 1]$. Then $\mathbf{T}(p) = \mathbf{c}'(p) = (x'(p), y'(p))$ is the tangent vector, $\mathbf{N}(p) = (-y'(p), x'(p))$ is the normal vector, and

$$s(p) = \int_0^p |\mathbf{c}'(q)| dq = \int_0^p \sqrt{(x'(p))^2 + (y'(p))^2} dq$$

is the arc length. Due to the regularity condition $\mathbf{c}'(p) \neq 0$, the arc length s is a differentiable function of p and $ds/dp = |\mathbf{c}'(p)|$. If we parametrize the regular curve $\mathbf{c}$ by s, then $\mathbf{T}(s) = d\mathbf{c}(s)/ds$ is the unit tangent vector satisfying $|\mathbf{T}(s)| = 1$. The number $\kappa(s) \doteq |d\mathbf{T}(s)/ds|$ is called the *curvature* at s, measuring the change rate of the angle which neighboring tangents make. Since $|\mathbf{T}(s)| = 1$, we have $\mathbf{T}(s) \cdot d\mathbf{T}(s)/ds = 0$, indicating $d\mathbf{T}(s)/ds$ is collinear to the unit normal vector $\mathbf{N}(s)$. That is, under the arc length parametrization, $d\mathbf{T}(s)/ds = \kappa(s)\mathbf{N}(s)$, or $\kappa(s) = |\mathbf{T} \times d\mathbf{T}/ds| = |\mathbf{c}'(s) \times \mathbf{c}''(s)|$ where × is the exterior product. Back to the general parametrization $\mathbf{c}(p)$, we have

$$\kappa(p) = \frac{|\mathbf{c}'(p) \times \mathbf{c}''(p)|}{|\mathbf{c}'(p)|^3} = \frac{x'y'' - x''y'}{((x')^2 + (y')^2)^{3/2}}. \tag{31}$$

Now we derive the divergence expression (6) of the curvature on a level curve. Consider the case where $\mathbf{c}(s)$ is the l-level curve of a function $u : R^2 \to R$, denoted by

$$\mathbf{c}(s) = \{(x(s), y(s)) : u(x(s), y(s)) = l\}.$$

By differentiating the equality $u(x(s), y(s)) = l$ with respect to s, we obtain

$$u_x x'(s) + u_y y'(s) = 0. \tag{32}$$

Hence the vectors $(x'(s), y'(s))$ and $(-u_y, u_x)$ are collinear, or equivalently for some λ we have

$$\begin{cases} x'(s) = -\lambda u_y, \\ y'(s) = \lambda u_x. \end{cases} \tag{33}$$

Note that since $|\mathbf{c}'(s)| = 1$, from (33) we get $\lambda = 1/|\nabla u|$ (supposing $|\nabla u| \neq 0$). If differentiating again (32) with respect to s we obtain

$$u_{xx}(x'(s))^2 + u_{yy}(y'(s))^2 + 2u_{xy}x'(s)y'(s) + u_x x''(s) + u_y y''(s) = 0.$$

Plugging (33) into the above equality leads to

$$\lambda^2[u_{xx}(u_y)^2 + u_{yy}(u_x)^2 - 2u_{xy}u_y u_x] + \frac{1}{\lambda}[y'(s)x''(s) - x'(s)y''(s)] = 0.$$

By (31) we can deduce the curvature expression as

$$\kappa(s) = \frac{|\mathbf{c}'(s) \times \mathbf{c}''(s)|}{|\mathbf{c}'(s)|^3} = x'(s)y''(s) - x''(s)y'(s) = \frac{u_{xx}(u_y)^2 + u_{yy}(u_x)^2 - 2u_{xy}u_y u_x}{|\nabla u|^3}.$$

Denoting $f \doteq |\nabla u| = \sqrt{(u_x)^2 + (u_y)^2}$, we have

$$\begin{aligned}
\nabla \cdot \left(\frac{\nabla u}{|\nabla u|}\right) &= \frac{\partial}{\partial x}(\frac{1}{f}u_x) + \frac{\partial}{\partial y}(\frac{1}{f}u_y) \\
&= \frac{\partial}{\partial x}(\frac{1}{f})u_x + \frac{1}{f}u_{xx} + \frac{\partial}{\partial y}(\frac{1}{f})u_y + \frac{1}{f}u_{yy} \\
&= -\frac{1}{f^2}f_x u_x - \frac{1}{f^2}f_y u_y + \frac{1}{f}(u_{xx} + u_{yy}) \\
&= -\frac{1}{f^2}\Big[\frac{1}{f}(u_x u_{xx} + u_y u_{yx})\Big]u_x - \frac{1}{f^2}\Big[\frac{1}{f}(u_x u_{xy} + u_y u_{yy})\Big]u_y + \frac{1}{f}(u_{xx} + u_{yy}) \\
&= -\frac{1}{f^3}\Big\{(u_x)^2 u_{xx} + (u_y)^2 u_{yy} + 2u_x u_y u_{xy} - \Big[(u_x)^2 + (u_y)^2\Big](u_{xx} + u_{yy})\Big\} \\
&= \frac{u_{xx}(u_y)^2 + u_{yy}(u_x)^2 - 2u_{xy}u_y u_x}{|\nabla u|^3} \\
&= \kappa(s),
\end{aligned}$$

thus getting the curvature expression (6).

Since the above derivations only consider the case of level curves for a 2D function $u(x, y)$, here we give some remarks on the curvature expression (6) in high dimensional spaces. For a level surface defined in 3D space, the curvature expression (6) at point p amounts to the mean curvature of this surface:

$$H = \frac{1}{2}\nabla \cdot \mathbf{N},$$

where $\mathbf{N}$ is a unit normal of the surface (see Chan and Shen, 2005, chap. 2.1.2). Formally, the mean curvature is defined as the average of the principal curvatures (Spivak, 1999, vol. 3, chap. 2): $H = (\kappa_1 + \kappa_2)/2$, where κ_1 and κ_2 are two principal curvatures. In this case, the Gaussian curvature is given by $K = \kappa_1 \cdot \kappa_2$. More generally (Spivak, 1999, vol. 4, chap. 7), for a $(d-1)$-dimensional level hypersurface embedded in R^d the mean curvature is given as $H = (\kappa_1 + \cdots + \kappa_{d-1})/(d-1)$ in terms of principal curvatures. More abstractly, the mean curvature is the trace of the second fundamental form divided by $d-1$ (or equivalently the shape operator or Weingarten map). The shape operator s (Lee, 1997, chap. 8) is an extrinsic curvature, and the Gaussian curvature is given by the determinant of s. Mean curvature is closely related to the first variation of surface area, in particular a minimal surface such as a soap film, has mean curvature zero and a soap bubble has constant mean curvature. Unlike Gauss curvature, the mean curvature is extrinsic and depends on the embedding, for instance, a cylinder and a plane are locally isometric but the mean curvature of a plane is zero while that of a cylinder is nonzero (see http://en.wikipedia.org/wiki/Curvature). One can also refer to Ambrosio and Masnou (2003) for the description of this high dimensional representation.

Appendix B: PDEs Derived by Calculus of Variations

We present the following derivations of the Euler-Lagrange PDEs by calculus of variations (van Brunt, 2004). Note that the variation operator δ acts much like a differentiation operator. First we list some expressions about δ which are useful in the following derivations (where $\mathbf{F}$ is a d-dimensional differentiable vector field):

$$\begin{aligned}
\delta(\nabla u) &= \nabla(\delta u),\\
\delta(\nabla\cdot\mathbf{F}) &= \delta\Big(\sum_{i=1}^{d}\frac{\partial F^{(i)}}{\partial x^{(i)}}\Big)=\sum_{i=1}^{d}\delta\Big(\frac{\partial F^{(i)}}{\partial x^{(i)}}\Big)=\sum_{i=1}^{d}\frac{\partial(\delta F)^{(i)}}{\partial x^{(i)}}=\nabla\cdot\delta\mathbf{F},\\
\delta(|\nabla u|^2) &= \delta\Big[\sum_{i=1}^{d}\Big(\frac{\partial u}{\partial x^{(i)}}\Big)^2\Big]=\sum_{i=1}^{d}2\frac{\partial u}{\partial x^{(i)}}\delta\Big(\frac{\partial u}{\partial x^{(i)}}\Big)=2\langle\nabla u,\delta\nabla u\rangle=2\langle\nabla u,\nabla\delta u\rangle,\\
\delta(|\nabla u|) &= \delta\Big\{\Big[\sum_{i=1}^{d}\Big(\frac{\partial u}{\partial x^{(i)}}\Big)^2\Big]^{1/2}\Big\}=\frac{1}{2}\Big[\sum_{i=1}^{d}\Big(\frac{\partial u}{\partial x^{(i)}}\Big)^2\Big]^{-1/2}\delta\Big[\sum_{i=1}^{d}\Big(\frac{\partial u}{\partial x^{(i)}}\Big)^2\Big]\\
&= \frac{1}{2}\frac{1}{|\nabla u|}2\langle\nabla u,\nabla\delta u\rangle=\Big\langle\frac{\nabla u}{|\nabla u|},\nabla\delta u\Big\rangle,\\
\delta\Big(\frac{1}{|\nabla u|}\Big) &= \delta\Big\{\Big[\sum_{i=1}^{d}\Big(\frac{\partial u}{\partial x^{(i)}}\Big)^2\Big]^{-1/2}\Big\}=-\frac{1}{2}\Big[\sum_{i=1}^{d}\Big(\frac{\partial u}{\partial x^{(i)}}\Big)^2\Big]^{-3/2}\delta\Big[\sum_{i=1}^{d}\Big(\frac{\partial u}{\partial x^{(i)}}\Big)^2\Big]\\
&= -\frac{1}{2}\frac{1}{|\nabla u|^3}2\langle\nabla u,\nabla\delta u\rangle=-\frac{1}{|\nabla u|^3}\langle\nabla u,\nabla\delta u\rangle.
\end{aligned}$$

Proof of (9)⇒(10). Suppose $u : R^d \to R$ is a differentiable function. The first variation, $E_{LR} \to E_{LR} + \delta E_{LR}$, under $u \to u + \delta u$ is given by

$$\begin{aligned}
\delta E_{LR} &= \delta\Big\{\int_\Omega\Big[(u-y)^2+\lambda|\nabla u|^2\Big]d\mathbf{x}\Big\}\\
&= \int_\Omega\Big\{\delta\Big[(u-y)^2\Big]+\lambda\delta\Big(|\nabla u|^2\Big)\Big\}d\mathbf{x}\\
&= \int_\Omega\Big[2(u-y)\delta u+2\lambda\langle\nabla u,\nabla\delta u\rangle\Big]d\mathbf{x}\\
&= 2\Big[\int_\Omega(u-y)\delta u d\mathbf{x}+\int_{\partial\Omega}\lambda\nabla u\delta u\cdot\mathbf{n}dS-\int_\Omega\lambda(\nabla\cdot\nabla u)\delta u d\mathbf{x}\Big] \qquad (34)\\
&= 2\Big[\int_\Omega(u-y)\delta u d\mathbf{x}+\int_{\partial\Omega}\lambda\frac{\partial u}{\partial\mathbf{n}}\delta u dS-\int_\Omega\lambda(\nabla\cdot\nabla u)\delta u d\mathbf{x}\Big] \qquad (35)\\
&= 2\int_\Omega\Big[(u-y)-\lambda\Delta u\Big]\delta u d\mathbf{x}. \qquad (36)
\end{aligned}$$

Here $\nabla\cdot$ is the divergence operator, Δ is the Laplacian operator, and $\mathbf{n}$ denotes the outer normal along the boundary $\partial\Omega$. The equation (34) is obtained based on the Gauss-Green divergence theorem in vector calculus (Spiegel and Lipschutz, 2009) (which is a special case of the more general Stokes' theorem):

$$\int_V(\nabla\cdot\mathbf{F})dV=\int_S(\mathbf{F}\cdot\mathbf{n})dS,$$

where V is a subset of R^d (in the case of $d = 3$, V represents a volume in 3D space) which is compact and has a piecewise smooth boundary S (also indicated with $\partial V = S$), $\mathbf{F}$ is a continuously differentiable vector field, and $\mathbf{n}$ is the outward pointing unit normal field of the boundary ∂V. In fact, we use the following corollary of the divergence theorem when applied to the product of a scalar function g (that is δu in our context) and a vector field $\mathbf{F}$ (∇u in our context):

$$\int_V [\mathbf{F} \cdot (\nabla g) + g(\nabla \cdot \mathbf{F})] dV = \int_S (g\mathbf{F} \cdot \mathbf{n}) dS.$$

Then integration by parts implies (34). The equation (35) is written with the directional derivative notation $\partial u / \partial \mathbf{n} \doteq \nabla u \cdot \mathbf{n} = \langle \nabla u, \mathbf{n} \rangle$. The last equation (36) is due to the assumption of *natural boundary conditions*

$$\frac{\partial u}{\partial \mathbf{n}} |_{\partial \Omega} = 0.$$

According to the *fundamental lemma of calculus of variations*, the integrand part in parentheses is equal to zero because δu is an arbitrary function. Hence we obtain (10).

Proof of (11)⇒(12). The first variation, $E_{TV} \to E_{TV} + \delta E_{TV}$, under $u \to u + \delta u$ is given by

$$\begin{aligned}
\delta E_{TV} &= \delta \Big\{ \int_\Omega \Big[\frac{1}{2}(u-y)^2 + \lambda |\nabla u| \Big] d\mathbf{x} \Big\} \\
&= \int_\Omega \Big\{ (u-y)\delta u + \lambda \delta(|\nabla u|) \Big\} d\mathbf{x} \\
&= \int_\Omega \Big[(u-y)\delta u + \lambda \Big\langle \frac{\nabla u}{|\nabla u|}, \nabla \delta u \Big\rangle \Big] d\mathbf{x} \\
&= \int_\Omega (u-y)\delta u d\mathbf{x} + \int_{\partial\Omega} \lambda \frac{1}{|\nabla u|} \frac{\partial u}{\partial \mathbf{n}} \delta u dS - \int_\Omega \lambda \Big(\nabla \cdot \frac{\nabla u}{|\nabla u|} \Big) \delta u d\mathbf{x} \qquad (37) \\
&= \int_\Omega \Big[(u-y) - \lambda \nabla \cdot \frac{\nabla u}{|\nabla u|} \Big] \delta u d\mathbf{x}. \qquad (38)
\end{aligned}$$

Again the integration term over the boundary $\partial\Omega$ in (37) can be removed by the natural boundary conditions. By the fundamental lemma of calculus of variations, the integrand part in parentheses of (38) must equal to zero. Thus we get (12).

Proof of (14)⇒(15). The original derivation comes from Chan et al. (2002). Let $f(\kappa) = a + b\kappa^2$ and the elastica regularization term be

$$R(u) = \int_\Omega f(\kappa) |\nabla u| d\mathbf{x}.$$

We need to prove that the first variation, $R(u) \to R(u) + \delta R(u)$, under $u \to u + \delta u$ is given by

$$\delta R(u) = \int_\Omega -\nabla \cdot \mathbf{V}(u) \delta u d\mathbf{x},$$

where $\mathbf{V}(u)$ is a flux field defined as

$$\mathbf{V}(u) = f(\kappa)\mathbf{N} - \frac{\mathbf{T}}{|\nabla u|} \frac{\partial (f'(\kappa)|\nabla u|)}{\partial \mathbf{T}}.$$

Here $\mathbf{N}$ is the ascending normal field $\nabla u/|\nabla u|$, and $\mathbf{T}$ is the tangent field defined as $\mathbf{T} = \mathbf{N}^{\perp}$. Note that the exact orientation of $\mathbf{T}$ does not matter due to the coupling of $\mathbf{T}$ and $\partial/\partial\mathbf{T}$ in the expression. Since the curvature κ is a function of u, by variational rules we have

$$\begin{aligned}
\delta R(u) &= \delta\Big\{ \int_\Omega f(\kappa)|\nabla u| d\mathbf{x}\Big\} \\
&= \int_\Omega \Big\{|\nabla u|\delta\Big[f(\kappa)\Big] + f(\kappa)\delta(|\nabla u|)\Big\}d\mathbf{x} \\
&= \int_\Omega \Big\{|\nabla u| f'(\kappa)\delta\kappa + f(\kappa)\Big\langle\frac{\nabla u}{|\nabla u|}, \nabla\delta u\Big\rangle\Big\}d\mathbf{x} \\
&= \int_\Omega |\nabla u| f'(\kappa)\delta\kappa d\mathbf{x} + \int_{\partial\Omega} \frac{f(\kappa)}{|\nabla u|}\frac{\partial u}{\partial\mathbf{n}}\delta u dS - \int_\Omega f(\kappa)\Big(\nabla\cdot\frac{\nabla u}{|\nabla u|}\Big)\delta u d\mathbf{x} \qquad (39) \\
&= \int_\Omega \Big\{|\nabla u| f'(\kappa)\delta\kappa - f(\kappa)\Big(\nabla\cdot\frac{\nabla u}{|\nabla u|}\Big)\delta u\Big\}d\mathbf{x} \\
&= \int_\Omega \Big\{|\nabla u| f'(\kappa)\delta\kappa - \Big[\nabla\cdot(f(\kappa)\mathbf{N})\Big]\delta u\Big\}d\mathbf{x}.
\end{aligned}$$

Here the integration term over the boundary $\partial\Omega$ in (39) can be removed by the natural boundary conditions. The variation of curvature $\kappa = \nabla\cdot\mathbf{N}$ is a function of δu, which can be further written as

$$\begin{aligned}
\delta\kappa &= \delta(\nabla\cdot\mathbf{N}) \\
&= \nabla\cdot\delta\mathbf{N} \\
&= \nabla\cdot\delta\Big(\frac{\nabla u}{|\nabla u|}\Big) \\
&= \nabla\cdot\Big[\frac{1}{|\nabla u|}\delta(\nabla u) + \nabla u\delta\Big(\frac{1}{|\nabla u|}\Big)\Big] \\
&= \nabla\cdot\Big[\frac{1}{|\nabla u|}\nabla(\delta u) - \nabla u\Big(\frac{1}{|\nabla u|^3}\langle\nabla u, \nabla(\delta u)\rangle\Big)\Big] \\
&= \nabla\cdot\Big[\frac{1}{|\nabla u|}\nabla(\delta u) - \frac{1}{|\nabla u|}\mathbf{N}\langle\mathbf{N}, \nabla(\delta u)\rangle\Big] \\
&= \nabla\cdot\Big[\frac{1}{|\nabla u|}(\mathbf{I} - \mathbf{N}\otimes\mathbf{N})\nabla(\delta u)\Big] \\
&= \nabla\cdot\Big[\frac{1}{|\nabla u|}P_{\mathbf{T}}(\nabla(\delta u))\Big].
\end{aligned}$$

Here $\mathbf{I}$ denotes the identity transform, $P_{\mathbf{N}} \doteq \mathbf{N}\otimes\mathbf{N}$ is the orthogonal projection onto the ascending normal direction of u, and $P_{\mathbf{T}} \doteq \mathbf{I} - \mathbf{N}\otimes\mathbf{N} = \mathbf{T}\otimes\mathbf{T}$ is the orthogonal projection onto the tangent direction of u. Therefore by the Gauss-Green divergence theorem we have

$$\begin{aligned}
&\int_\Omega |\nabla u| f'(\kappa)\delta\kappa d\mathbf{x} \\
= &\int_\Omega f'(\kappa)|\nabla u|\Big\{\nabla\cdot\Big[\frac{1}{|\nabla u|}P_{\mathbf{T}}(\nabla(\delta u))\Big]\Big\}d\mathbf{x} \\
= &\int_{\partial\Omega} f'(\kappa)P_{\mathbf{T}}(\nabla(\delta u))\cdot\mathbf{n}dS - \int_\Omega\Big\langle\nabla\Big[f'(\kappa)|\nabla u|\Big], \frac{1}{|\nabla u|}P_{\mathbf{T}}(\nabla(\delta u))\Big\rangle d\mathbf{x}
\end{aligned}$$

$$
\begin{align*}
&= -\int_{\Omega}\Big\langle \nabla\Big[f'(\kappa)|\nabla u|\Big], \frac{1}{|\nabla u|}P_{\mathbf{T}}(\nabla(\delta u))\Big\rangle d\mathbf{x} \\
&= -\int_{\Omega}\Big\langle \frac{1}{|\nabla u|}P_{\mathbf{T}}\Big\{\nabla\Big[f'(\kappa)|\nabla u|\Big]\Big\}, \nabla(\delta u)\Big\rangle d\mathbf{x} \tag{40} \\
&= -\int_{\partial\Omega}\frac{1}{|\nabla u|}P_{\mathbf{T}}\Big\{\nabla\Big[f'(\kappa)|\nabla u|\Big]\Big\}\delta u\cdot\mathbf{n}dS + \int_{\Omega}\nabla\cdot\frac{1}{|\nabla u|}P_{\mathbf{T}}\Big\{\nabla\Big[f'(\kappa)|\nabla u|\Big]\Big\}\delta u d\mathbf{x} \\
&= \int_{\Omega}\nabla\cdot\frac{1}{|\nabla u|}P_{\mathbf{T}}\Big\{\nabla\Big[f'(\kappa)|\nabla u|\Big]\Big\}\delta u d\mathbf{x},
\end{align*}
$$

where proper natural boundary conditions are imposed to remove the integrations over the boundary $\partial\Omega$, and the equation (40) is given by the symmetry property of the projection operator $P_{\mathbf{T}}$ in an inner product. Finally, using the definition of directional derivative $P_{\mathbf{T}}(\nabla f) = \mathbf{T}(\partial f/\partial\mathbf{T})$, we complete the derivations of (14)⇒(15) by

$$
\begin{aligned}
\delta R(u) &= \int_{\Omega}\Big\{|\nabla u|f'(\kappa)\delta\kappa - \Big[\nabla\cdot(f(\kappa)\mathbf{N})\Big]\delta u\Big\}d\mathbf{x} \\
&= \int_{\Omega}\Big\{\nabla\cdot\frac{1}{|\nabla u|}P_{\mathbf{T}}\Big\{\nabla\Big[f'(\kappa)|\nabla u|\Big]\Big\} - \nabla\cdot(f(\kappa)\mathbf{N})\Big\}\delta u d\mathbf{x} \\
&= -\int_{\Omega}\nabla\cdot\Big\{f(\kappa)\mathbf{N} - \frac{1}{|\nabla u|}P_{\mathbf{T}}\Big\{\nabla\Big[f'(\kappa)|\nabla u|\Big]\Big\}\delta u d\mathbf{x} \\
&= -\int_{\Omega}\nabla\cdot\Big\{f(\kappa)\mathbf{N} - \frac{1}{|\nabla u|}\frac{\partial(f'(\kappa)|\nabla u|)}{\partial\mathbf{T}}\mathbf{T}\Big\}\delta u d\mathbf{x} \\
&= -\int_{\Omega}\nabla\cdot\mathbf{V}\delta u d\mathbf{x}.
\end{aligned}
$$

Appendix C: Expressions in Terms of RBF Approximations

The following gives some useful expressions about Laplacian, Hessian, and curvature of $u(\mathbf{x})$ in terms of RBF approximations $u(\mathbf{x}) = \sum_{i=1}^{n} w_i\phi_i(\mathbf{x})$, where $\phi_i(\mathbf{x}) = \exp(-c|\mathbf{x}-\mathbf{x_i}|^2/2)$.

Proof of (18) for Laplacian:

$$
\begin{aligned}
\frac{\partial^2\phi_k}{\partial x^{(i)}\partial x^{(i)}} &= \frac{\partial}{\partial x^{(i)}}\Big(\frac{\partial\phi_k}{\partial x^{(i)}}\Big) \\
&= \frac{\partial}{\partial x^{(i)}}\Big[-c(x^{(i)}-x_k^{(i)})\phi_k\Big] \\
&= -c(x^{(i)}-x_k^{(i)})\frac{\partial\phi_k}{\partial x^{(i)}} - c\phi_k\frac{\partial(x^{(i)}-x_k^{(i)})}{\partial x^{(i)}} \\
&= -c(x^{(i)}-x_k^{(i)})[-c(x^{(i)}-x_k^{(i)})\phi_k] - c\phi_k \\
&= c[c(x^{(i)}-x_k^{(i)})^2-1]\phi_k. \\
\Delta\phi_k &= \sum_{i=1}^{d}\frac{\partial^2\phi_k}{\partial x^{(i)}\partial x^{(i)}} = c(c|x-x_k|^2-d)\phi_k.
\end{aligned}
$$

Proof of (19) for Hessian:

$$
(\text{for } i\neq j)\quad \frac{\partial^2\phi_k}{\partial x^{(i)}\partial x^{(j)}} = \frac{\partial}{\partial x^{(j)}}[-c(x^{(i)}-x_k^{(i)})\phi_k]
$$

$$\begin{aligned} &= -c(x^{(i)} - x_k^{(i)})[-c(x^{(j)} - x_k^{(j)})\phi_k] \\ &= c^2(x^{(i)} - x_k^{(i)})(x^{(j)} - x_k^{(j)})\phi_k. \\ \text{(for } i = j\text{)} \quad \frac{\partial^2 \phi_k}{\partial x^{(i)} \partial x^{(j)}} &= c[c(x^{(i)} - x_k^{(i)})^2 - 1]\phi_k. \\ \mathbf{H}(\phi_k) &= c^2(\mathbf{x} - \mathbf{x}_k)(\mathbf{x} - \mathbf{x}_k)^T \phi_k - c\phi_k \mathbf{I}. \end{aligned}$$

Proof of (21) for Hessian in terms of notation $\Phi \doteq \sum_{i=1}^n w_i(\mathbf{x} - \mathbf{x}_i)(\mathbf{x} - \mathbf{x}_i)^T \phi_i$:

$$\begin{aligned} \mathbf{H}(u) &= \sum_{i=1}^n w_i \mathbf{H}(\phi_i) \\ &= \sum_{i=1}^n w_i[-c\phi_i \mathbf{I} + c^2(\mathbf{x} - \mathbf{x}_i)(\mathbf{x} - \mathbf{x}_i)^T \phi_i] \\ &= -c\sum_{i=1}^n w_i \phi_i \mathbf{I} + c^2 \sum_{i=1}^n w_i(\mathbf{x} - \mathbf{x}_i)(\mathbf{x} - \mathbf{x}_i)^T \phi_i \\ &= -c\Big(\sum_{i=1}^n w_i \phi_i\Big)\mathbf{I} + c^2\Phi. \end{aligned}$$

To prove (23) and others derivations involving gradients in Appendix D, here we list some useful expressions (notice that we do not distinguish $\mathbf{H}(u)$ from $\mathbf{H}(u)^T$ due to symmetry):

$$\begin{aligned} \nabla(|\nabla u|^2) &= \nabla\Big[\sum_{i=1}^d \Big(\frac{\partial u}{\partial x^{(i)}}\Big)^2\Big] = \sum_{i=1}^d 2\frac{\partial u}{\partial x^{(i)}} \nabla\Big(\frac{\partial u}{\partial x^{(i)}}\Big) \\ &= \sum_{i=1}^d 2\frac{\partial u}{\partial x^{(i)}} \begin{pmatrix} \frac{\partial^2 u}{\partial x^{(i)} \partial x^{(1)}} \\ \cdots \\ \frac{\partial^2 u}{\partial x^{(i)} \partial x^{(d)}} \end{pmatrix} = 2\mathbf{H}(u)\nabla u, \\ \nabla(|\nabla u|) &= \nabla\Big\{\Big[\sum_{i=1}^d \Big(\frac{\partial u}{\partial x^{(i)}}\Big)^2\Big]^{1/2}\Big\} = \frac{1}{2}\Big[\sum_{i=1}^d \Big(\frac{\partial u}{\partial x^{(i)}}\Big)^2\Big]^{-1/2} \nabla\Big[\sum_{i=1}^d \Big(\frac{\partial u}{\partial x^{(i)}}\Big)^2\Big] \\ &= \frac{1}{2|\nabla u|} 2\mathbf{H}(u)\nabla u = \frac{1}{|\nabla u|}\mathbf{H}(u)\nabla u, \\ \nabla\Big(\frac{1}{|\nabla u|}\Big) &= \nabla\Big\{\Big[\sum_{i=1}^d \Big(\frac{\partial u}{\partial x^{(i)}}\Big)^2\Big]^{-1/2}\Big\} = -\frac{1}{2}\Big[\sum_{i=1}^d \Big(\frac{\partial u}{\partial x^{(i)}}\Big)^2\Big]^{-3/2} \nabla\Big[\sum_{i=1}^d \Big(\frac{\partial u}{\partial x^{(i)}}\Big)^2\Big] \\ &= -\frac{1}{2|\nabla u|^3} 2\mathbf{H}(u)\nabla u = -\frac{1}{|\nabla u|^3}\mathbf{H}(u)\nabla u, \\ \nabla\Big(\frac{1}{|\nabla u|^3}\Big) &= \nabla\Big\{\Big[\sum_{i=1}^d \Big(\frac{\partial u}{\partial x^{(i)}}\Big)^2\Big]^{-3/2}\Big\} = -\frac{3}{2}\Big[\sum_{i=1}^d \Big(\frac{\partial u}{\partial x^{(i)}}\Big)^2\Big]^{-5/2} \nabla\Big[\sum_{i=1}^d \Big(\frac{\partial u}{\partial x^{(i)}}\Big)^2\Big] \\ &= -\frac{3}{2|\nabla u|^5} 2\mathbf{H}(u)\nabla u = -\frac{3}{|\nabla u|^5}\mathbf{H}(u)\nabla u. \end{aligned}$$

Proof of (23) for curvature:

$$\kappa \doteq \nabla \cdot \Big(\frac{\nabla u}{|\nabla u|}\Big)$$

$$
\begin{aligned}
&= \frac{1}{|\nabla u|}\nabla \cdot \nabla u + \nabla\Big(\frac{1}{|\nabla u|}\Big) \cdot \nabla u \\
&= \frac{1}{|\nabla u|}\Delta u - \frac{1}{|\nabla u|^3}\mathbf{H}(u)\nabla u \cdot \nabla u \\
&= \frac{1}{|\nabla u|}\Big(\Delta u - \frac{\nabla u^T \mathbf{H}(u)\nabla u}{\nabla u^T \nabla u}\Big) \\
&= \frac{1}{|-c\mathbf{g}|}\left\{ c\sum_i w_i(c|\mathbf{x}-\mathbf{x}_i|^2 - d)\phi_i - \frac{(-c\mathbf{g})^T(c^2\Phi - c(\sum_i w_i\phi_i)\mathbf{I})(-c\mathbf{g})}{(-c\mathbf{g})^T(-c\mathbf{g})} \right\} \\
&= \frac{1}{|\mathbf{g}|}\left\{ \sum_i w_i(c|\mathbf{x}-\mathbf{x}_i|^2 - d)\phi_i - \frac{\mathbf{g}^T(c\Phi - (\sum_i w_i\phi_i)\mathbf{I})\mathbf{g}}{\mathbf{g}^T\mathbf{g}} \right\} \\
&= \frac{1}{|\mathbf{g}|}\left\{ \sum_i w_i(c|\mathbf{x}-\mathbf{x}_i|^2 - d + 1)\phi_i - c\frac{\mathbf{g}^T\Phi\mathbf{g}}{\mathbf{g}^T\mathbf{g}} \right\}.
\end{aligned}
$$

Appendix D: Expansion of $\nabla \cdot \mathbf{V}$ in Gradient Descent Time Marching

Here we give the derivations of the expansion (26) of $\nabla \cdot \mathbf{V}$ in Gradient Descent Time Marching. For simpler notations we define

$$
\alpha \doteq \nabla u^T \mathbf{H}(u)\nabla u, \quad \beta \doteq \nabla u^T \mathbf{H}(u)^2 \nabla u, \quad \gamma \doteq \nabla u^T \mathbf{H}(u)^3 \nabla u.
$$

By definition (16)

$$
\begin{aligned}
\mathbf{V}(u) &\doteq f(\kappa)\mathbf{N} - \frac{\mathbf{T}}{|\nabla u|}\frac{\partial(f'(\kappa)|\nabla u|)}{\partial \mathbf{T}} \\
&= f(\kappa)\mathbf{N} - \frac{1}{|\nabla u|}\nabla(f'(\kappa)|\nabla u|) + \frac{1}{|\nabla u|^3}\nabla u\langle \nabla u, \nabla\left(f'(\kappa)|\nabla u|\right)\rangle \\
&= (1+b\kappa^2)\mathbf{N} - \frac{1}{|\nabla u|}\nabla(2b\kappa|\nabla u|) + \frac{1}{|\nabla u|^3}\nabla u\langle \nabla u, \nabla\left(2b\kappa|\nabla u|\right)\rangle,
\end{aligned}
$$

we have

$$
\nabla \cdot \mathbf{V} = \nabla \cdot [(1+b\kappa^2)\mathbf{N}] - 2b\nabla \cdot \Big[\frac{1}{|\nabla u|}\nabla(\kappa|\nabla u|)\Big] + 2b\nabla \cdot \Big\{\frac{1}{|\nabla u|^3}\nabla u\Big[\nabla u^T \nabla(\kappa|\nabla u|)\Big]\Big\}. \quad (41)
$$

Then we show the following derivations for the three parts on the right side of (41).

Part 1: The first term can be expanded as

$$
\nabla \cdot \mathbf{N} + b\nabla \cdot (\kappa^2 \mathbf{N}) = \kappa + b[\nabla(\kappa^2)\cdot \mathbf{N} + \kappa^2 \nabla \cdot \mathbf{N}] = \kappa + b(2\kappa\nabla\kappa \cdot \mathbf{N} + \kappa^3),
$$

where $\nabla\kappa$ can be further written as

$$
\begin{aligned}
\nabla\kappa &= \nabla\Big[\nabla \cdot \Big(\frac{\nabla u}{|\nabla u|}\Big)\Big] \\
&= \nabla\Big[\nabla\Big(\frac{1}{|\nabla u|}\Big)\cdot \nabla u + \frac{1}{|\nabla u|}\Delta u\Big]
\end{aligned}
$$

$$
\begin{aligned}
&= \cancel{\nabla\Big[\nabla\Big(\frac{1}{|\nabla u|}\Big)\Big]}\cdot\nabla u + \nabla(\nabla u)\cdot\nabla\Big(\frac{1}{|\nabla u|}\Big) + \nabla\Big(\frac{1}{|\nabla u|}\Big)\Delta u + \cancel{\frac{1}{|\nabla u|}\nabla(\Delta u)}\\
&\approx \mathbf{H}(u)\nabla\Big(\frac{1}{|\nabla u|}\Big) + \Delta u\nabla\Big(\frac{1}{|\nabla u|}\Big)\\
&= -\frac{1}{|\nabla u|^3}[\mathbf{H}(u)^2\nabla u + \Delta u\mathbf{H}(u)\nabla u].
\end{aligned}
$$

Here the third equality is obtained by the formula for the gradient of a dot product

$$
\begin{aligned}
\nabla(\mathbf{a}\cdot\mathbf{b}) &= (\nabla\mathbf{a})\cdot\mathbf{b} + (\nabla\mathbf{b})\cdot\mathbf{a}\\
&= \begin{pmatrix} \frac{\partial a_1}{\partial x_1} & \cdots & \frac{\partial a_d}{\partial x_1}\\ \cdots & \cdots & \cdots\\ \frac{\partial a_1}{\partial x_d} & \cdots & \frac{\partial a_d}{\partial x_d}\end{pmatrix}\mathbf{b} + \begin{pmatrix} \frac{\partial b_1}{\partial x_1} & \cdots & \frac{\partial b_d}{\partial x_1}\\ \cdots & \cdots & \cdots\\ \frac{\partial b_1}{\partial x_d} & \cdots & \frac{\partial b_d}{\partial x_d}\end{pmatrix}\mathbf{a},
\end{aligned}
$$

and we omit the third order derivatives by notation $\diagup$ for easier calculations. Therefore, the first term on the right side of (41) can be written as

$$
\begin{aligned}
\nabla\cdot((1+b\kappa^2)\mathbf{N}) &= \kappa + b\kappa^3 - \frac{2b\kappa}{|\nabla u|^4}[\nabla u^T\mathbf{H}(u)^2\nabla u + \Delta u\nabla u^T\mathbf{H}(u)\nabla u]\\
&= \kappa + b\kappa^3 - \frac{2b\kappa}{|\nabla u|^4}(\alpha\Delta u + \beta).
\end{aligned}
$$

Part 2: The second term on the right side of (41) can be expanded as

$$
\begin{aligned}
&\nabla\cdot\Big[\frac{1}{|\nabla u|}\nabla(\kappa|\nabla u|)\Big]\\
={}& \nabla\Big(\frac{1}{|\nabla u|}\Big)\cdot\nabla(\kappa|\nabla u|) + \frac{1}{|\nabla u|}\nabla\cdot\Big[\nabla(\kappa|\nabla u|)\Big]\\
={}& -\frac{1}{|\nabla u|^3}\mathbf{H}(u)\nabla u\cdot\Big[|\nabla u|\nabla\kappa + \kappa\nabla(|\nabla u|)\Big] + \frac{1}{|\nabla u|}\Big\{\nabla\cdot\Big[|\nabla u|\nabla\kappa + \kappa\nabla(|\nabla u|)\Big]\Big\}\\
={}& -\frac{1}{|\nabla u|^2}\nabla u^T\mathbf{H}(u)\nabla\kappa - \frac{\kappa}{|\nabla u|^3}\nabla u^T\mathbf{H}(u)\nabla(|\nabla u|)\\
&+\frac{1}{|\nabla u|}\Big[\nabla(|\nabla u|)\cdot\nabla\kappa + \cancel{|\nabla u|\nabla\cdot\nabla\kappa} + \nabla\kappa\cdot\nabla(|\nabla u|) + \cancel{\kappa\nabla\cdot\nabla(|\nabla u|)}\Big]\\
\approx{}& -\frac{1}{|\nabla u|^2}\nabla u^T\mathbf{H}(u)\nabla\kappa - \frac{\kappa}{|\nabla u|^3}\nabla u^T\mathbf{H}(u)\Big[\frac{1}{|\nabla u|}\mathbf{H}(u)\nabla u\Big] + \frac{2}{|\nabla u|^2}\nabla u^T\mathbf{H}(u)\nabla\kappa\\
={}& \frac{1}{|\nabla u|^2}\nabla u^T\mathbf{H}(u)\nabla\kappa - \frac{\kappa}{|\nabla u|^4}\nabla u^T\mathbf{H}(u)^2\nabla u\\
={}& \frac{1}{|\nabla u|^2}\nabla u^T\mathbf{H}(u)\Big\{-\frac{1}{|\nabla u|^3}[\mathbf{H}(u)^2\nabla u + \Delta u\mathbf{H}(u)\nabla u]\Big\} - \frac{\kappa}{|\nabla u|^4}\beta\\
={}& -\Big(\frac{\Delta u}{|\nabla u|^5} + \frac{\kappa}{|\nabla u|^4}\Big)\beta - \frac{1}{|\nabla u|^5}\gamma.
\end{aligned}
$$

Part 3: Finally we consider the third term on the right side of (41). With notation $\mathbf{v} \doteq \nabla(\kappa|\nabla u|)$, we have

$$
\nabla\cdot\Big\{\frac{1}{|\nabla u|^3}\nabla u\Big[\nabla u^T\nabla(\kappa|\nabla u|)\Big]\Big\}
$$

$$
\begin{aligned}
&= \nabla \cdot \Big[\frac{1}{|\nabla u|^3}\nabla u(\nabla u \cdot \mathbf{v})\Big] \\
&= \nabla\Big(\frac{\nabla u \cdot \mathbf{v}}{|\nabla u|^3}\Big) \cdot \nabla u + \Big(\frac{\nabla u \cdot \mathbf{v}}{|\nabla u|^3}\Big)\Delta u \\
&= \Big[\nabla\Big(\frac{1}{|\nabla u|^3}\Big)(\nabla u \cdot \mathbf{v}) + \cancel{\frac{1}{|\nabla u|^3}\nabla(\nabla u \cdot \mathbf{v})}\Big] \cdot \nabla u + \Big(\frac{\nabla u \cdot \mathbf{v}}{|\nabla u|^3}\Big)\Delta u \\
&\approx \Big[\nabla\Big(\frac{1}{|\nabla u|^3}\Big) \cdot \nabla u + \frac{\Delta u}{|\nabla u|^3}\Big](\nabla u \cdot \mathbf{v}) \\
&= \Big(\frac{\Delta u}{|\nabla u|^3} - \frac{3}{|\nabla u|^5}\alpha\Big)(\nabla u \cdot \mathbf{v}).
\end{aligned}
$$

Because

$$
\begin{aligned}
\nabla u \cdot \mathbf{v} &= \nabla u \cdot [\nabla(\kappa|\nabla u|)] \\
&= \nabla u \cdot (|\nabla u|\nabla\kappa + \kappa\nabla(|\nabla u|)) \\
&= |\nabla u|\nabla u \cdot \nabla\kappa + \kappa\nabla u \cdot \nabla(|\nabla u|) \\
&= |\nabla u|\nabla u \cdot \Big\{-\frac{1}{|\nabla u|^3}[\mathbf{H}(u)^2\nabla u + \Delta u\mathbf{H}(u)\nabla u]\Big\} + \kappa\nabla u \cdot \Big(\frac{1}{|\nabla u|}\mathbf{H}(u)\nabla u\Big) \\
&= \Big(\frac{\kappa}{|\nabla u|} - \frac{\Delta u}{|\nabla u|^2}\Big)\alpha - \frac{1}{|\nabla u|^2}\beta,
\end{aligned}
$$

we obtain the expansion of the third term on the right side of (41):

$$
\begin{aligned}
&\nabla \cdot \Big\{\frac{1}{|\nabla u|^3}\nabla u\Big[\nabla u^T\nabla(\kappa|\nabla u|)\Big]\Big\} \\
&= \Big(\frac{\Delta u}{|\nabla u|^3} - \frac{3}{|\nabla u|^5}\alpha\Big)(\nabla u \cdot \mathbf{v}) \\
&= \Big(\frac{\kappa\Delta u}{|\nabla u|^4} - \frac{(\Delta u)^2}{|\nabla u|^5}\Big)\alpha + \Big(\frac{3\Delta u}{|\nabla u|^7} - \frac{3\kappa}{|\nabla u|^6}\Big)\alpha^2 - \frac{\Delta u}{|\nabla u|^5}\beta + \frac{3}{|\nabla u|^7}\alpha\beta
\end{aligned}
$$

Putting all three parts together, we have the expansion of $\nabla \cdot \mathbf{V}$ as

$$
\begin{aligned}
\nabla \cdot \mathbf{V} &= \kappa + b\kappa^3 - \frac{2b\kappa}{|\nabla u|^4}(\alpha\Delta u + \beta) + 2b\Big\{\Big(\frac{\Delta u}{|\nabla u|^5} + \frac{\kappa}{|\nabla u|^4}\Big)\beta + \frac{1}{|\nabla u|^5}\gamma\Big\} \\
&\quad +2b\Big\{\Big(\frac{\kappa\Delta u}{|\nabla u|^4} - \frac{(\Delta u)^2}{|\nabla u|^5}\Big)\alpha + \Big(\frac{3\Delta u}{|\nabla u|^7} - \frac{3\kappa}{|\nabla u|^6}\Big)\alpha^2 - \frac{\Delta u}{|\nabla u|^5}\beta + \frac{3}{|\nabla u|^7}\alpha\beta\Big\} \\
&= \kappa + b\kappa^3 - \frac{2b(\Delta u)^2}{|\nabla u|^5}\alpha + 6b\Big(\frac{\Delta u}{|\nabla u|^7} - \frac{\kappa}{|\nabla u|^6}\Big)\alpha^2 + \frac{6b}{|\nabla u|^7}\alpha\beta + \frac{2b}{|\nabla u|^5}\gamma.
\end{aligned}
$$

References

L. Ambrosio and S. Masnou. A direct variational approach to a problem arising in image reconstruction. *Interface and Free Boundaries*, 5:63–81, 2003.

L. Ambrosio, N. Fusco, and D. Pallara. *Functions of Bounded Variation and Free Discontinuity Problems.* Oxford University Press, 2000.

A. Asuncion and D.J. Newman. *UCI Machine Learning Repository.* 2013. URL `http://archive.ics.uci.edu/ml/`.

G. Aubert and P. Kornprobst. *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations.* Springer-Verlag, 2nd edition, 2006.

E. Bae, J. Shi, and X.C. Tai. Graph cuts for curvature based image denoising. *IEEE Transaction on Image Processing*, 20(5):1199–1210, 2011.

P.L. Bartlett and M. Traskin. Adaboost is consistent. *Journal of Machine Learning Research*, 8:2347–2368, 2007.

P.L. Bartlett, M.I. Jordan, and J.D. McAuliffe. Convexity, classification, and risk bounds. *Journal of American Statistical Association*, 101(473):138–156, March 2006.

M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7(48):2399–2434, 2006.

G. Biau, L. Devroye, and G. Lugosi. Consistency of random forests and other averaging classifiers. *Journal of Machine Learning Research*, 9:2015–2033, 2008.

C.M. Bishop. *Pattern Recognition and Machine Learning.* Springer-Verlag, 2006.

S. Boucheron, O. Bousquet, and G. Lugosi. Theory of classification: A survey of recent advances. *ESAIM: Probability and Statistics*, 9:323–375, 2005.

S. Boucheron, G. Lugosi, and P. Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence.* Oxford University Press, 2013.

O. Bousquet and A. Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, 2002.

O. Bousquet, S. Boucheron, and G. Lugosi. Introduction to statistical learning theory. In *Advanced Lectures in Machine Learning.* Springer, 2004.

K. Bredies, T. Bock, and B. Wirth. A convex, lower semi-continuous approximation of euler's elastica energy. Preprint, 2013.

L. Breiman. Random forest. *Machine Learning*, 45(1):5–32, 2001.

R. Caruana and A. Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 161–168, Pittsburgh, Pennsylvania, 2006.

T.F. Chan and J. Shen. Nontexture inpainting by curvature driven diffusions (CDD). *Journal of Visual Communication and Image Representation*, 12:436–449, 2001.

T.F. Chan and J. Shen. *Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods.* SIAM, 2005.

T.F. Chan, S.H. Kang, and J. Shen. Euler's elastica and curvature-based inpaintings. *SIAM Journal on Applied Mathematics*, 63:564–592, 2002.

C.C. Chang and C.J. Lin. Libsvm – a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:1–27, 2011. URL `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`.

J.B. Conway. *A Course in Functional Analysis*. Springer-Verlag, 2nd edition, 1990.

D. Cossock and T. Zhang. Statistical analysis of Bayes optimal subset ranking. *IEEE Transaction on Information Theory*, 54(11):5140–5154, 2008.

N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.

M. do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, 1976.

Y. Duan, Y. Wang, and J. Hahn. A fast augmented lagrangian algorithm for euler's elastica models. *Numerical Mathematics: Theory, Methods and Applications*, 6(1):47–71, 2013.

J.C. Duchi, L.W. Mackey, and M.I. Jordan. On the consistency of ranking algorithm. In *Proceedings of the 27th International Conference on Machine Learning*, pages 327–334, Haifa, Israel, 2010.

M. Fernández-Delgado, E. Cernadas, and S. Barro. Do we need hundreds of classifiers to solve real world classification problems? *Journal of Machine Learning Research*, 15: 3133–3181, 2014.

M.S. Floater and A. Iske. Multistep scattered data interpolation using compactly supported radial basis functions. *Journal of Computational and Applied Mathematics*, 73(1–2):65–78, 1996.

C.G. Fraser. Mathematical technique and physical conception in Euler's investigation of the elastica. *Centaurus*, 34(3):211–246, 1991.

W. Gao and Z.H. Zhou. On the consistency of multi-label learning. *Artificial Intelligence*, 199–200:22–44, 2013.

E. Giusti. *Minimal Surfaces and Functions of Bounded Variation*. Birkhäuser, Boston, 1994.

T. Glasmachers. Universal consistency of multi-class support vector classication. In *Proceedings of the 24th Annual Conference on Neural Information Processing Systems*, Vancouver, Canada, 2010.

B.I. Golubov and A.G. Vitushkin. Variation of a function. In M. Hazewinkel, editor, *Encyclopedia of Mathematics*. Springer, 2001. URL `http://www.encyclopediaofmath.org/index.php/Function_of_bounded_variation`, updated in 2013.

J. Hahn, G.J. Chung, Y. Wang, and X.C. Tai. Fast algorithms for p-elastica energy with the application to image inpaiting and curve reconstruction. In *Proceedings of International Conference on Scale Space and Variational Methods in Computer Vision*, pages 169–182. Springer, 2011.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer-Verlag, 2009.

C.-W. Hsu and C.-J. Lin. A comparison of methods for multi-class support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, 2002.

C.-W. Hsu, C.-C. Chang, and C.-J. Lin. *A practical guide to support vector classification.* 2007. URL `http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf`.

X. Huang, L. Shi, and J.A.K. Suykens. Ramp loss linear programming support vector machine. *Journal of Machine Learing Research*, 15(6):2185–2211, 2014.

J.K. Hunter. Chapter 7: L^p spaces. *Course Notes of Measure Theory*, 2011. URL `http://www.math.ucdavis.edu/~hunter/m206/ch6_measure_notes.pdf`.

G. Kanizsa. *Organization in Vision.* Praeger, New York, 1979.

N. Komodakis and N. Paragios. Beyond pairwise energies: efficient optimization for higher-order MRFs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2985–2992, 2009.

S. Kutin and P. Niyogi. Almost everywhere algorithmic stability and generalization error. In *Proceedings of the Eighteenth Conference Conference on Uncertainty in Artificial Intelligence*, pages 275–282, Alberta,Canada, 2002.

J.M. Lee. *Riemannian Manifolds: An Introduction to Curvature.* Springer-Verlag, 1997.

G.P. Leonardi and S. Masnou. Locality of the mean curvature of rectifiable varifolds. *Advances in Calculus of Variations*, 2(1):17–42, 2009.

R. Levien. The elastica: a mathematical history. Technical report, EECS Department, University of California, Berkeley, 2008.

S. Masnou and J.-M. Morel. Level lines based disocclusion. In *Proceedings of the 5th IEEE International Conference on Image Processing*, pages 259–263, Chicago, Illinois, October 1998.

M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of Machine Learning.* MIT Press, 2012.

J.M. Morel and S. Solimini. Variational methods in image segmentation. In *Progress in Nonlinear Differential Equations and Their Applications.* Birkhäuser, Boston, 1995.

D. Mumford. Elastica and computer vision. In C.L. Bajaj, editor, *Algebraic Geometry and Its Applications*, pages 491–506. Springer-Verlag, New York, 1994.

K.P. Murphy. *Machine Learning: a Probabilistic Perspective.* MIT Press, 2012.

B. Nadler, N. Srebro, and X. Zhou. Semi-supervised learning with the graph laplacian: The limit of infinite unlabelled data. In *Proceedings of the 24th Annual Conference on Neural Information Processing Systems*, pages 1330–1338, Vancouver, B.C., Canada, 2009.

M. Nandan, P.P. Khargonekar, and S.S. Talathi. Fast SVM training using approximate extreme points. *Journal of Machine Learning Research*, 15(1):59–98, 2014.

F. Nie, Y. Huang, and H. Huang. Linear time solver for primal SVM. In *Proceedings of The 31st International Conference on Machine Learning*, pages 505–513, Beijing, 2014.

S. Osher and J.A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, 1988.

T. Poggio, S. Rifkin, S. Mukherjee, and P. Niyogi. General conditions for predictivity in learning theory. *Nature*, 428:419–422, 2004.

R. M. Rifkin. *Everything Old Is New Again : A Fresh Look at Historical Approaches in Machine Learning.* PhD thesis, MIT, 2002.

L.I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60:259–268, 1992.

R.E. Schapire and Y. Freund. *Boosting: Foundations and Algorithms.* MIT Press, 2012.

B. Schölkopf and A. Smola. *Learning with Kernels.* MIT Press, 2002.

M. R. Spiegel and S. Lipschutz. *Vector Analysis.* McGraw-Hill, 2nd edition, 2009.

M. Spivak. *A Comprehensive Introduction to Differential Geometry*, volume 3–4. Publish or Perish Press, 3rd edition, 1999.

I. Steinwart. Consistency of support vector machines and other regularized kernel classifiers. *IEEE Transactions on Information Theory*, 51(1):128–142, 2005.

X.C. Tai, J. Hahn, and G.J. Chung. A fast algorithm for euler's elastica model using augmented lagrangian method. *SIAM Journal on Imaging Sciences*, 4(1):313–344, 2011.

A. Tewari and P.L. Bartlett. On the consistency of multiclass classification methods. *Journal of Machine Learning Research*, 8:1007–1025, 2007.

B. van Brunt. *The Calculus of Variations.* Springer-Verlag, 2004.

V.N. Vapnik. *Statistical Learning Theory.* Wiley-Interscience, 1998.

K. R. Varshney and A. S. Willsky. Classification using geometric level sets. *Journal of Machine Learning Research*, 11(2):491–516, 2010.

U. von Luxburg and B. Schölkopf. Statistical learning theory: Models, concepts, and results. Technical report, arXiv:0810.4752, 2008.

J. Wang, P. Wonka, and J. Ye. Scaling SVM and least absolute deviations via exact data reduction. In *Proceedings of The 31st International Conference on Machine Learning*, pages 523–531, Beijing, 2014.

H. Wendland. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in Computational Mathematics*, 4(1):389–396, 1995.

F. Xia, T.Y. Liu, J. Wang, W. Zhang, and H. Li. Listwise approach to learning to rank: Theory and algorithm. In *Proceedings of the 25th International Conference on Machine Learning*, pages 1192–1199, Helsinki, Finland, 2008.

K. Yosida. *Functional Analysis*. Springer-Verlag, 6th edition, 1999.

A. Zakai and Y. Ritov. Consistency and localizability. *Journal of Machine Learning Research*, 10:827–856, 2009.

T. Zhang. Statistical behavior and consistency of classification methods based on convex risk minimization. *Annals of Statistics*, 32(1):56–85, 2004a.

T. Zhang. Statistical analysis of some multi-category large margin classification methods. *Journal of Machine Learning Research*, 5:1225–1251, 2004b.

D. Zhou and B. Schölkopf. Regularization on discrete spaces. In *Proceedings of the 27th DAGM Symposium Symposium on Pattern Recognition*, pages 361–368, Springer, Berlin, 2005.