Decoupling Noises and Features via Weighted $\ell_1$-analysis Compressed Sensing


Graphics&Geometric Computing Laboratory
University of Science and Technology of China

Technical Report
September 1, 2012
Decoupling Noises and Features via Weighted $\ell_1$-analysis Compressed Sensing

Ruimin WANG and Zhouwang YANG and Ligang LIU and Jiansong DENG and Falai CHEN
University of Science and Technology of China

Technical Report #GCL-2012-03
http://gcl.ustc.edu.cn

Abstract

Many geometry processing applications are sensitive to noises and sharp features. Although there are a number of works on detecting noises and sharp features in the literature, they are heuristic. On the one hand, traditional denoising methods use filtering operators to remove the noises, however, may blur sharp features and shrink the object. On the other hand, noises make detection of features, which relies on computation of differential properties, unreliable and unstable. Therefore, detecting noises and features on discrete surfaces still remains challenging.

In this paper, we present a novel approach for decoupling noises and features on 3D shapes. Our approach consists of two phases. In the first phase, an estimated base mesh is generated to approximate the true underlying surface of the input noisy mesh by a global Laplacian regularization denoising scheme. The base mesh is guaranteed to asymptotically converge to the underlying surface with probability one as the sample size goes to infinity. In the second phase, an $\ell_1$-analysis compressed sensing optimization is proposed to recover sharp features from the residual between the base mesh and the input mesh. This is based on our discovery that sharp features can be sparsely represented in some coherent dictionary which is constructed by the pseudo-inverse matrix of the Laplacian of the shape. The features are recovered from the residual in a progressive way. Theoretical analysis and experimental results have shown that our approach reliably and robustly removes noises and extracts sharp features on 3D shapes.

Keywords: Smoothing, denoising, sharp feature, $\ell_1$-analysis compressed sensing

1 Introduction

With digital scanning devices becoming widespread, more and more acquired raw data of the sampled 3D models is available. Even with high-resolution scanners, the raw data contains inevitable noise from various sources. Although a large number of mesh denoising schemes already exist [Taubin 1995; Fleishman et al. 2003; Jones et al. 2003; Zheng et al. 2010], removing noise while preserving sharp features still remains a challenge.

The reasons are three-fold. First, sharp features and noise are ambiguous since there are no precise mathematical definitions for them. Features are often hard to distinguish from large amounts of noise even for human beings (see Figures 1 and 7). Second, traditional approaches adopt local filtering operators, which rely on various differential properties, such as normal (or tangent plane) and curvature, to average nearby points to “remove” the noise. On the one hand, computation of the differential properties is a “chicken and egg” problem, since the definition of geometric differential assumes local smoothness, and its computation is not reliable and robust in the presence of noise. On the other hand, the filtering operators would blur sharp features and shrink the object. Third, most of the existing denoising methods require fine tuning of various parameters in order to produce the best results for different inputs, thus making it difficult for users.
In this work, we present a novel approach for decoupling noises and sharp features of 3D shapes. Suppose vertices of the input mesh are sampled from an underlying surface with additive independent and identically distributed (i.i.d.) random noise over the surface. Our idea is to estimate a base mesh to approximate the true underlying surface and then recover the features from the residual between the base mesh and the input mesh.

First, we propose a discrete Laplacian regularization smoothing (DLRS) model to estimate the base mesh. The objective function consists of a data term, which measures how far the smoothed points are from the original data, and a smoothness term, which uses the discrete Laplacian of the points to measure the smoothness of the resulted points. Unlike previous approaches which heuristically select the smoothness parameter to balance the data term and the smoothness term, we present an automatic scheme, based on the generalized cross validation (GCV) scheme, to compute the optimal value of this parameter according to the input data.

If the true underlying surface is $C^2$-smooth, we prove that the computed parameter is asymptotically optimal. That is, with this optimal parameter, the estimated base surface is guaranteed to asymptotically converge to the true underlying surface with probability one when the sample size goes infinity.

In general cases, an underlying surface is a piecewise $C^2$-smooth surface with sharp features. As we mentioned above, it is difficult to identify these features in the presence of noise. We find that features are regarded as large-scale “noise” in the denoising phase and the large “errors” produced will be uniformly distributed to other regions in the base mesh. That is, the presence of sharp features may result in generating artifacts in regions of $C^2$ during the denoising phase.

We thus propose a novel approach for identifying and recovering features from the residual between the estimated base mesh and the input mesh. This is achieved by two observations. First, the residual inevitably mixes the information of features and the noise. We discover that the pseudo-inverse matrix of the Laplacian matrix of the mesh is a coherent dictionary for sparsely representing the feature signal on the shape. Second, we are highly inspired by the promising development of the technique of Compressed Sensing in recent years [Candes and Tao 2005; Donoho 2006; Eldar and Kutyniok 2012].

Therefore, we formulate the identification of sharp features as an $\ell_1$-analysis compressed sensing optimization problem. We propose an iterative procedure to recover the features progressively. In each iteration, we identify a part of features. After marking the identified features, we find more features in the next iteration. This is reasonable as the feature becomes sparser in the latter iterations as the identified features in previous iterations have already been marked and do not contribute to the feature detection.

We demonstrate the performance of our method on a number of diverse inputs, with either synthetic or real noise, and demonstrate its ability to denoise the surfaces and discover their features. We also compare with previous approaches and the results show that our approach achieves much better results than the state-of-the-art methods.

Contributions. Our approach for decoupling noises and features is quite different from previous approaches. To the best of our knowledge, this is the first time noise and features are analyzed and separated in such an elegant manner with theoretical guaranty. The contributions of our approach are summarized in the following:

- Asymptotically optimal surface denoising: Our denoising approach is fully automatic without tuning any parameter. The smoothness parameter is automatically computed. The denoised surface is guaranteed to asymptotically converge to the true underlying surface with probability one if the underlying surface is $C^2$-smooth.
- Faithful feature recovering by $\ell_1$-analysis: We successfully apply the $\ell_1$-analysis compressed sensing technique to identify and recover sharp features from the residual between the estimated base surface and the input surface. This is based on our discovery that the pseudo-inverse matrix of the Laplacian matrix of the mesh is a coherent dictionary for sparsely representing the sharp feature signal on the shape.

2 Related Work

2.1 Feature preserving surface denoising

A wide variety of surface denoising/smoothing algorithms have been proposed during the past two decades [Botsch et al. 2010]. A thorough review on this topic is out of the scope of this paper.

The most common techniques are based on Laplacian operators. Taubin [Taubin 1995] developed a fast and simple iterative local smoothing scheme based on the definition of the Laplacian operator on meshes. This approach was extended to irregular meshes using a geometric flow analogy by Desbrun et al. [Desbrun et al. 1999]. Other methods extended feature-preserving anisotropic diffusion in image processing to anisotropic geometric diffusion on surfaces [Clarenz et al. 2000; Bajaj and Xu 2003].

Fleishman et al. [Fleishman et al. 2003] and Jones et al. [Jones et al. 2003] extended the bilateral filter from image denoising [Tomasi and Manduchi 1998] to mesh denoising, which anisotropically averages the nearby vertices weighted by a monotonously decreasing function in terms of both spatial difference and vertex difference. Duguet et al. [Duguet et al. 2004] proposed a higher order bilateral filter for mesh denoising. The bilateral filter was also applied to the facet normal field defined over the mesh by Zheng et al. [Zheng et al. 2010]. The normal field was first filtered and then the denoised surface was reconstructed from the filtered normal field. This scheme obtains better results than previous normal-based filtering methods [Yagou et al. 2002; Sun et al. 2007]. Fan et al. [Fan et al. 2010] applied an anisotropic filter and second-order bilateral filter to smooth the normal field as well as the curvature tensor field, which can better preserve curvature details and alleviate volume shrinkage during denoising.

A few global, non-iterative mesh smoothing approaches have been developed during the last few years. Nealen et al. [Nealen et al. 2006] and Liu et al. [Liu et al. 2007] proposed similar global smoothing schemes by setting the vertex Laplacians to zero and reconstructing the surface with geometric constraints. Su et al. [Su et al. 2009] adopted a mean filter to smooth the vertex Laplacians and then reconstructed the geometry from the filtered Laplacian.

Unlike previous approaches which simultaneously denoise the data and preserve the features, our approach separate features and noises in two phases, that is, generates a base mesh by denoising the input mesh and then recovers features from the residual.

2.2 Compressed sensing (CS)

In recent years, Compressed Sensing (CS) has attracted considerable attention in areas of applied mathematics, computer science, and signal processing [Candes and Tao 2005; Donoho 2006; Candes and Wakin 2008; Candes et al. 2010; Eldar and Kutyniok 2012]. The central insight of CS is that many signals are sparse, i.e., represented using only a few non-zero coefficients in a suitable ba-
s or dictionary and such signals can be recovered from very few measurements (undersampled data) by a nonlinear optimization. Designing measurement/sensing matrices with favorable properties and constructing suitable dictionaries are the important research topics in CS. We will see in Section 5 that the pseudo-inverse matrix of Laplacian of a mesh is a coherent dictionary for representing the topics in CS. We will see in Section 5 that the pseudo-inverse matrix and constructing suitable dictionaries are the important research measurements (undersampled data) by a nonlinear optimization.

Instead of applying the \( \ell_1 \)-analysis CS optimization. Using our \( \ell_1 \)-analysis approach on the residual between the denoised mesh and the original mesh, we can faithfully identify the locations of sharp features. Then sharp features can eventually be recovered by a modified Laplacian optimization.

The general idea of \( \ell_1 \)-regularization for the purpose of sparse signal reconstruction has been used in the community of geometry processing [Avron et al. 2010; Habbecke and Kobbelt 2012] recently. Instead of applying the \( \ell_1 \)-regularization in the optimization, our approach adopts the \( \ell_1 \)-analysis CS framework to recover sparse features on 3D shapes.

3 Overview

In contrast from most previous approaches, which denoise the input data while simultaneously preserving the features, we perform this in two phases. In the first phase, we obtain an estimated base mesh, which is obtained by denoising the input mesh. In the second phase, we recover sharp features from the residual between the base mesh and the input data. Figure 2 shows the pipeline of our approach.

Phase I: Mesh denoising. A global Laplacian regularization denoising scheme is developed to denoise the input mesh. The denoised mesh is considered as an estimated base mesh for an approximation to the true underlyng surface (see Figure 2(b)). We use the generalized cross validation (GCV) method to automatically choose an optimal parameter which is used to balance the data term and the smoothness term in the objective function. We have proved that by using this parameter the resulting denoised mesh is asymptotically optimal, which means it approximates the true underlying surface with probability one as the sample size goes to infinity if the underlying surface is \( C^2 \)-smooth without any sharp features. See Section 4 for the detail.

Phase II: Feature recovering. However, for non \( C^2 \)-smooth surfaces with sharp features, the residual between the base mesh and the input mesh inevitably mixes the information from features and noises. Then we perform the second phase to recover features from the residual. We discover that the pseudo-inverse matrix of the Laplacian matrix of a mesh is a coherent dictionary for sparsely representing sharp feature signals on the shape. Also we are profoundly inspired by the emerging technique of compressed sensing in recent years. Therefore, we formulate the identification and recovery of sharp features as an \( \ell_1 \)-analysis compressed sensing optimization on the residual. To handle with shapes with many features, we employ an iterative process to recover the features (see Figure 2(c)). The detail is elaborated in Section 5.

4 Mesh Denoising with Asymptotic Optimality

Given a noisy mesh with sharp features, our first step is to seek a base mesh which approximates the original noise-free mesh as much as possible. To this end, we propose a discrete Laplacian regularization smoothing (DLRS) model for estimating the base surface. We also prove the theoretical properties of our DLRS model, such as convergence rate and asymptotic optimality.

4.1 Denoising with DLRS Model

Problem formulation. The problem of surface denoising can be stated as follows: Assume that we are given a mesh \( P = \{p_i\}_{i=1}^n \), where \( p_i \) are sampled possibly with noise from a \( C^2 \)-smooth surface \( S \), i.e.,

\[
p_i = s_i + \varepsilon_i n_i, \quad i = 1, \ldots, n,
\]

where \( n_i \) is the unit normal of surface at \( s_i \in S \) and \( \varepsilon_i \) represents noise. The noise \( \varepsilon_i, i = 1, \ldots, n \), are assumed to be independent and identically distributed (i.i.d.) random variables with zero mean and finite variance \( \sigma^2 \). The goal of denoising is to produce a smooth mesh surface \( \hat{S} = \{\hat{s}_i\} \) to approximate the true underlying surface \( S \) as much as possible.

DLRS denoising model. To find a \( C^2 \)-smooth surface \( \hat{S} \) which is a reasonable estimate of the true underlying surface \( S \), we formulate it as the following variational minimization problem

\[
\hat{S} = \arg\min_S \frac{1}{n} \sum_{i=1}^{n} d^2(p_i, S) + \lambda \int_S (2H)^2, \quad (2)
\]

where \( d(p_i, S) \) is the geometric distance from \( p_i \) to \( S \). \( H \) is the mean curvature of \( S \), and \( \lambda \) is a smoothness parameter. The objective function consists of two terms: the data term, which measures how good the surface approximates the points, and the smoothness (regularization) term, which measures how smooth the surface is. The smoothness parameter \( \lambda \) plays the role of balancing the two terms.

To simplify the computation, we replace the smoothness term by its
discrete approximation

\[ J(S) = \frac{1}{n} \sum_{i=1}^{n} 4H^2(s_i) = \frac{1}{n} \sum_{i=1}^{n} \| \Delta_S s_i \|^2 \approx \frac{1}{n} \sum_{i=1}^{n} \| L_i S \|^2, \quad (3) \]

where \( S = (s_1, \ldots, s_n)^T \in \mathbb{R}^{n \times 3} \), \( \Delta_S \) is the Laplace-Beltrami operator on surface \( S \), and \( L = (L_1^T, \ldots, L_n^T)^T \) is the discrete Laplacian matrix. The second equality in (3) is derived by \( \Delta_S s_i = 2H n_i \). Thus we arrive at the following denoising model:

\[ \hat{S} = \arg \min_S \frac{1}{n} \sum_{i=1}^{n} \| p_i - s_i \|^2 + \lambda \sum_{i=1}^{n} \| L_i S \|^2. \quad (4) \]

We call it the discrete Laplacian regularization smoothing (DLRS) model.

Denote \( P = (p_1, \ldots, p_n)^T \in \mathbb{R}^{n \times 3} \). The DLRS model eventually leads to linear systems

\[ (I_n + \lambda M) \hat{S} = P, \quad (5) \]

where \( M = L^T L = \sum_{i=1}^{n} L_i^T L_i \). Thus, given a specific value of the smoothness parameter \( \lambda \), we have the solution

\[ \hat{S}_n(\lambda) = (I_n + \lambda M)^{-1} P \]

where \( I_n \) is the \( n \times n \) identity matrix, as the estimated base mesh of the true underlying surface \( S \).

Choosing the optimal value of \( \lambda \). Our DLRS model is similar to the global Laplacian optimization approaches [Nealen et al. 2006; Liu et al. 2007]. However, previous works select the smoothness parameter \( \lambda \) in heuristic manners or they allow users to adjust \( \lambda \) to control the smoothness of the denoised results.

We observe that there is no specific value of \( \lambda \), which works for all input data. That is, the parameter \( \lambda \) should be chosen by different values for different input data to gain the best denoised results. Inspired by the research on smoothing splines in statistics [Wahba 1990], we adopt the generalized cross validation (GCV) [Wahba 1990] (Chapter 4) to determine the smoothness parameter \( \lambda \) in our DLRS model.

Specifically, the merit function of GCV is defined as

\[ \text{GCV}_n(\lambda) = \frac{\frac{1}{n} \| P - \hat{S}_n(\lambda) \|_F^2}{(1 - \frac{1}{n} \text{tr}[A_n(\lambda)])^2}, \quad (6) \]

where \( A_n(\lambda) = (I_n + \lambda M)^{-1} \). And the optimal value of \( \lambda \) can be computed by minimizing the above GCV function, i.e.,

\[ \hat{\lambda}_G = \arg \min_{\lambda > 0} \text{GCV}_n(\lambda), \quad (7) \]

which can be easily solved by a line search optimization.

Fast computation. Computation of \( \text{tr}[A_n(\lambda)] \) in (6) is expensive due to the costly computation of the inverse of the matrix \( I_n + \lambda M \) and the spectra (eigenvalues), for various values of \( \lambda \). Here we compute the eigenvalues \( \mu_1, \mu_2, \ldots, \mu_n \) of \( M \). Then \( \text{tr}[A_n(\lambda)] \) can be simply calculated as

\[ \text{tr}[A_n(\lambda)] = \sum_{i=1}^{n} \frac{1}{1 + \mu_i}. \quad (8) \]

Note that \( M \) is a semi-positive definite sparse matrix. The number of nonzeros of \( M \) is strictly less than \((k + 1)^2 n\) and empirically \( 3kn \) when \( k \) nearest neighbors are used to build the Laplacian matrix \( L \). The symmetric reverse Cuthill-McKee ordering [Cuthill and McKee 1969] of \( M \) returns a permutation \( q = \text{symrcm}(M) \) such that \( M(q, q) \) tends to have its nonzero elements closer to the diagonal, thus becoming a band matrix. For a real symmetric sparse matrix \( M, M(q, q) \) has the same eigenvalues as \( M \). Thus we compute the eigenvalues of \( M(q, q) \) instead of \( M \) as it takes much less time to compute the eigenvalues of a symmetric band matrix than a non-banded one. Our fast implementation benefits greatly from the sparsity of \( M \) and is very efficient in both spatial and temporal cost.

Choice of Laplacian. The discrete Laplacian \( L \) of a mesh could be defined in various ways. Instead of using the geometric Laplacian which might be unreliable for noisy data, we employ the graph Laplacian of the input mesh \( P \). We have proved that, with the graph Laplacian, as long as the points are sampled uniformly (more precisely, the points satisfying the quasi-uniform assumption, see its definition in the supplementary material), the estimated base mesh \( \hat{S} \) computed by our DLRS model with the optimal \( \lambda \) can well approximate the true underlying surface \( S \) (see detail in next subsection). As we will see in Section 5, we can also modify the Laplacian matrix \( L \) according to shape features in our DLRS model.

Pseudocode. The pseudocode of computing the estimated base surface can be seen in Algorithm 1.

**Algorithm 1 Computing the estimated base mesh \( S \) by our DLRS model.**

**Input:** the points \( P \) and its Laplacian \( L \)

**Output:** \( \hat{S} = \text{DLRS}(P, L) \)

1: Calculate \( M = L^T L \) and its eigenvalues \( \mu_1, \ldots, \mu_n \).
2: Compute the optimal smoothness parameter by minimizing the GCV function in (6), i.e., \( \hat{\lambda}_G = \arg \min_{\lambda > 0} \text{GCV}_n(\lambda) \).
3: Obtain \( \hat{S} \) by solving linear systems \( (I_n + \hat{\lambda}_G M) S = P \).

4.2 Asymptotic Properties

We have established the theoretical properties (convergence rate and asymptotic optimality) of our DLRS model under some regularity conditions. All the technical proofs can be found in the supplementary material. Specifically, we have the following two main theorems.

Denote the error between the estimated base mesh surface and the true underlying surface as

\[ r_n(\lambda) = \frac{1}{n} \| \hat{S}_n(\lambda) - S \|_F^2. \]

**Theorem 1.** Assume that \( P \) is the equidistributed sample of a \( C^2 \)-smooth surface \( S \). As \( n \to \infty \) and \( \lambda \sim n^{-2/3} \) is chosen, we have with probability one

\[ \mathbb{E}[r_n(\lambda)] = O(n^{-2}). \]

**Theorem 2.** If the smoothness parameter \( \hat{\lambda}_G \) is the GCV choice according to (7), then the estimated base mesh surface \( \hat{S}_n(\hat{\lambda}_G) \) from
our DLRS model is asymptotically optimal, i.e.,
\[
\frac{r_n(\lambda_C)}{\lambda_C} \to_p 1
\] (9)
where \(\to_p\) means the convergence in probability.

From the above theorems, it is seen that the estimated base mesh surface \(\hat{S}\) asymptotically converges to the ground truth surface \(S\) with probability one as the sample size goes to infinity.

5 Feature Recovering via \(\ell_1\)-analysis Compressed Sensing

If the true underlying surface \(S\) is \(C^2\)-smooth, the denoised mesh \(\hat{S}\) obtained by our model DLRS is guaranteed to be an asymptotically optimal approximation to \(S\). However, the underlying surface \(S\) might not be a \(C^2\)-smooth surface as a whole but a piecewise \(C^0\) smooth surface with \(C^0\) sharp features.

The presence of shape features may result in generating artifacts in regions of \(C^2\) during the denoising phase, see the small bumps in the denoised meshes shown in Figures 2(b) and 8(c). This is because features are regarded as large-scale “noise” and the large “errors” produced will be distributed to other regions in \(\hat{S}\) by the global optimization (4). In this section we propose a novel approach for recovering the features from the residual between the base mesh \(\tilde{S}\) and the given noisy mesh \(P\).

5.1 \(\ell_1\)-analysis compressed sensing on residual

**Residual.** By the denoising phase we now have the estimated base surface \(\tilde{S} = \{\tilde{s}_i\}_{i=1}^{n}\) from the input noisy mesh \(P = \{p_i\}_{i=1}^{n}\). The residual between \(\tilde{S}\) and \(P\) is defined as
\[
b_i = (p_i - \tilde{s}_i)^T \hat{n}_i, \quad i = 1, \cdots, n.
\] (10)
where \(\hat{n}_i\) is the unit normal vector of surface \(\tilde{S}\) at \(\tilde{s}_i\). Denote \(b = (b_1, \cdots, b_n)^T\) as the residual vector.

As an inference of asymptotic properties presented in Section 4, the residual signal \(b\) is essentially i.i.d. noise when the true underlying surface of input mesh \(P\) is at least \(C^0\)-continuous as a whole. But in case the underlying surface containing sharp features, the residual \(b\) inevitably mixes the information from features and the noise, as illustrated in Figure 3 (left), and is decomposed as
\[
b = h + z,
\] (11)
where \(h = (h_1, \cdots, h_n)^T\) is the unknown signal of the features and \(z = (z_1, \cdots, z_n)^T\) is the measurement errors.

\(\ell_1\)-analysis compressed sensing. The compressed sensing theory asserts that if the unknown signal is reasonably sparse, it is possible to recover it under suitable conditions on the sensing matrix by an \(\ell_1\)-norm convex programming [Candès and Tao 2005; Donoho 2006]. The techniques hold for signals which are sparse in the standard coordinate basis or sparse with respect to some orthonormal basis. However, there are numerous applications in which a signal of interest is usually not sparse in an orthonormal basis but in a coherent dictionary, see detail in [Candès et al. 2010].

**Coherent dictionary for shape features.** Considering the signal of shape feature \(h\) in our case, we discover that shape features can be represented as sparse in some coherent dictionary which is constructed by the pseudo-inverse matrix \(L^+\) of the Laplacian matrix \(L\) of the shape. It is not surprising to see that the Laplacian of feature \(Lh\) is indeed a sparse signal and has quickly decaying coefficients, as illustrated in Figure 3 (right). Furthermore, Figure 4 illustrates the basis functions corresponding to three columns of \(L^+\), which means that \(L^+\) is a coherent dictionary for representing the \(C^0\) feature signals on the shape.

\(\ell_1\)-analysis on residual. As we have found a coherent dictionary \(D = L^+\) for representing \(h\) sparsely, i.e., \(D^*h\) is sparse, we can thus formulate the problem of recovering feature signals \(h\) from the residual \(b\) as an \(\ell_1\)-analysis compressed sensing optimization.

In our setting, the sensing matrix is the identity matrix, and the coherence dictionary is \(D = L^+\). Thus we have the following \(\ell_1\)-analysis compressed sensing optimization:
\[
\begin{align*}
\min_h \|D^* h\|_1 = \|Lh\|_1 & \quad \text{s.t.} \quad \|h - b\|_2 \leq \epsilon, \\
\end{align*}
\] (12)
where \(\epsilon^2\) is a likely upper bound on the noise power \(\|z\|_2^2\). The roles of the penalty and the constraint in (12) might also be reversed if we choose to constrain the sparsity and obtain the best fit for that sparsity. Here we prefer to solve an equivalent optimization:
\[
\begin{align*}
\min_h \|h - b\|_2^2 & \quad \text{s.t.} \quad \|Lh\|_1 \leq \tau, \\
\end{align*}
\] (13)
where $\tau$ is a tunable parameter controlling the sparsity. We call $\tau$ the sparsity parameter.

Denote $\hat{h}$ as the solution to the optimization (13). We assert that $L\hat{h}$ provides accurate and reliable locations of sharp features. Otherwise, if $L\hat{h}$ incorrectly locates the features, $\hat{h}$ might bias the original signal with large error.

**Weighted $\ell_1$-analysis.** Consider the weighted $\ell_1$-analysis on the residual

$$\min_{\hat{b}} \| \hat{h} - \hat{b} \|_2$$

s.t. $$\| W(L\hat{h}) \|_1 = \sum_{i=1}^{n} w_i |L_i \hat{h}| \leq \tau$$

where $W = \text{diag}(w_1, \cdots, w_n)$ and $w_1, \cdots, w_n$ are positive weights. The weighted $\ell_1$-analysis optimization (14) can be regarded as a relaxation of an $\ell_0$-minimization problem. It is desired that the weights could be to counteract the influence of the signal magnitude on the $\ell_1$-penalty function. Ideally, the weights are inversely proportional to the true signal magnitude, i.e.,

$$w_i = \begin{cases} \frac{1}{\| L_i \hat{S} \|} & L_i \hat{h} \neq 0, \\ \infty & L_i \hat{h} = 0. \end{cases} \quad (15)$$

The large entries in $W$ force the Laplacian of feature $L\hat{h}$ to concentrate on the indices where $w_i$ is small. These constructed weights precisely correspond to the indices where $L\hat{h}$ is nonzero. It is impossible to construct the precise weights (15) without knowing the feature signal $\hat{h}$ itself, but this suggests more generally that large weights could be used to discourage nonzero entries in the recovered $L\hat{h}$, while small weights could be used to encourage nonzero entries.

An iterative algorithm of reweighted $\ell_1$-minimization is proposed by [Candes et al. 2008] to enhance the sparsity in signal recovery. There exists such a possibility of constructing a favorable set of weights based on an approximation of $L\hat{h}$ or on other information about the vector magnitudes. Based on the geometric information of the estimated base surface $\hat{S}$, we design the weights as follows

$$w_i = \frac{1}{\rho + \| L_i \hat{S} \|}, i = 1, \cdots, n,$$

where $\rho$ is a small number ($\rho = 10^{-7}$ by default) that provides numerical stability and should be set slightly smaller than the expected nonzero magnitudes of $L\hat{h}$. With these well designed weights, we then perform a weighted $\ell_1$-analysis (14) on the residual for recovering the feature signal.

### 5.2 Iterative feature recovering

If we set large value of the sparsity parameter $\tau$, the optimization (14) will recover features but may introduce some non-feature points in the result. Thus we prefer to choose a small value of $\tau$. For some models with large portion of features, the solution to (14) returns only the most prominent (sharpest) features.

To recover the other features, our idea is to modify the rows corresponding to the identified features in the Laplacian matrix $L$ and perform the weighted $\ell_1$-analysis optimization (14) in an iterative manner. In each iteration, we identify a part of features. After marking the identified features, we find more features in the next iteration. This is reasonable as features become sparser in the latter iterations as the identified features in previous iterations have already marked and do not contribute to the feature recovering.

#### Feature types and classifications.

Generally there are two types of sharp features, corners and creases, as shown in Figure 5, on 3D shapes. A corner point is the one at which the tangent of any passing curve on the surface is discontinuous. A crease curve introduces the discontinuities of first derivatives across it, but preserves $C^2$-continuity along it. Corners can also be the intersections of several creases.

After identifying locations of the features by the weighted $\ell_1$-analysis (14) on the residual, we adopt a simple scheme to classify their types. If a feature is isolated from others, it is identified as a corner. If a feature point has a few feature points in its neighbor, we compute a dominant direction by PCA and classify it as a crease along this direction, see Figure 5.

#### Feature aware modification of Laplacian matrix.

If the vertex $s_i$ is identified as a corner, we remove the Laplacian penalty $\mathcal{L}(s_i) = L_i S$ by setting the $i$-th row of Laplacian matrix $L_i = 0$. If the vertex $s_i$ is identified as a point on a crease $E$, we only remove the straddling smoothness penalties and yield a term of the form

$$\mathcal{L}(s_i) = L_i S = \sum_{j \in N(i) \cap E} (s_j - s_{i}) = s_{i_{k-}} + s_{i_{k+}} - 2s_{i},$$

where $s_{i_{k-}}$ and $s_{i_{k+}}$ are the adjacent neighbors of $s_i$ along the crease $E$, as illustrated in Figure 5(b).

#### Termination condition of iterations.

We adopt a statistical method of nonparametric test for checking whether the residual $b$ contains more features. Specifically, residuals were randomly divided into two sets $b_1$ and $b_2$. We use a two-sample Kolmogorov-Smirnov test [Massey 1951] to compare the distributions of the values in the two sets $b_1$ and $b_2$. We state that

- The null hypothesis $H_0$: $b_1$ and $b_2$ are from the same continuous distribution.
- The alternative hypothesis $H_1$: they are from different continuous distributions.

This hypothesis does not specify what that common distribution is (e.g. normal or not normal). In statistics, a result is called statistically significant if it is unlikely to have occurred by chance alone, according to a pre-determined threshold probability, the significance level. The result is 1 if the test rejects the null hypothesis at the $\alpha$ significance level; 0 otherwise. We use the significance level $\alpha = 0.05$ in our implementation.

#### Pseudo-code.

The pseudo-code of the iterative feature recovering via $\ell_1$-analysis can be found in Algorithm 2.
Our approach computes the optimal smoothness parameter to denoise the mesh. (a) the ground truth 8-like model which is a $C^2$-smooth surface; (b) the model artificially corrupted by severe synthetic noise; (c)-(g) denoised results by the global smoothing approach with various parameters $\lambda$ in (4). Our approach obtains the best smoothing result (e) using the optimal parameter $\lambda = 2.02$.

Algorithm 2 Iterated procedure for feature recovering

Input: the points $P$, its Laplacian $L$, and the sparsity parameter $\tau$

Output: denoised mesh with features

1: Call Algorithm 1, $\hat{S} = DLRS(P, L)$.
2: Calculate the residual $b$ according to (10).
3: Recover the features $\hat{h}$ by the weighted $\ell_1$-analysis (14), and get the reliable locations of $\hat{h}$ indicated by $L\hat{h}$.
4: If the result of the two-sample Kolmogorov-Smirnov test on residual $b$ is 1, go to Step 5; Otherwise, exit and output the current $\hat{S}$.
5: Classify the features $\hat{h}$ and modify accordingly the Laplacian matrix $L$ based on their feature types.
6: Go to Step 1.

6 Experimental Results

We have implemented our approach and tested it on a large variety of models with different types of features. All the examples presented in this paper were made on a dual-core 3GHz machine with 8G memory (see more results in the accompanying video).

Like previous works, we mostly use ground truth models with synthetic noise for evaluating our method and comparing with other approaches. The synthetic noise is generated by an i.i.d. random variable generator with zero mean and a standard deviation $\sigma$ which is proportional to the diagonal of the bounding box of the model.

Figure 6 shows a denoising example on a model which is discretized from a $C^2$-smooth 8-like surface. We added severe synthetic noise ($\sigma = 0.02$) on the model as shown in Figure 6(b). The denoised results with different values of the smoothness parameter $\lambda$ are shown in Figure 6(c-g). It is seen that small $\lambda$ cannot filter out the noise (see (c) and (d)) while large $\lambda$ may shrink the object (see (f) and (g)). Thanks to the GCV method, our algorithm can automatically choose an optimal value of $\lambda = 2.02$ and obtain the best denoised result as shown in Figure 6(e).

The torus model in Figure 7 (upper) is also a $C^2$-smooth surface. We added different levels of noise on the model and our algorithm obtains good denoised results as shown in Figure 7 (also see the color maps which encode the difference between the denoised models and the ground truth model).

For $C^2$-smooth surfaces we only need to perform the first phase to obtain the final denoised results. For non-smooth surfaces with sharp features, we need to perform the second phase to recover the features. Figure 8(a) shows a surface with sharp features. We added some synthetic noise on it shown in Figure 8(b) and applied our smoothing algorithm on (b). The denoised result filters out the
The sharp features are progressively recovered using our method, as shown in Figure 8(d) and (e). All sharp features are correctly recovered in the final result (f).

We also tested our algorithm on the cube model with corners and creases, as shown in Figure 7 (below). It is seen from the results that our approach can faithfully recover features corrupted by different levels of noise, even by heavily added noise. Figure 1 shows a more complex model with many sharp features. Sharp features are correctly recovered and the resulting mesh has slight difference with the ground truth from the color map.

Comparisons. We compare our approach with the vertex based bilateral filtering method [Fleishman et al. 2003] and the normal based bilateral filtering method [Zheng et al. 2010]. Figure 9 shows the comparisons, with close-up views showing the differences. For the methods of [Fleishman et al. 2003] and [Zheng et al. 2010], we chose the parameters that produce visually the best denoised results as shown in the figure. It is seen that vertex based bilateral filtering often blurs sharp features. Normal based bilateral filtering preserves the sharp feature for light noise (see upper row in Figure 9), however, may lose features for heavy noise and introduce visual artifacts in the result (see lower row in Figure 9).

Real point cloud data. Our algorithm can also be applied for denoising real scanning data which is generally point cloud without connectivity. The idea is to build a graph connecting the points and then denoise the data using its graph Laplacian. For each point in the cloud, we find its neighbors within a certain distance and thus obtain a graph for all the points. The Laplacian matrix can be defined on the graph thus our algorithm can be applied. Figure 10 shows a few examples of denoised results on real point data, which shows that our algorithm removes the noise in the raw data while preserving the features well.

Parameter. Previous methods need to fine tune various parameters to produce the best results for different inputs. There is only one parameter in our approach, i.e., the sparsity parameter $\tau$ used in the weighted $\ell_1$-analysis (14). We prefer to set a small value of $\tau$ and perform the iterative process to recover features from the residual sequentially.

Using an iterative procedure to recover the features on shapes tends to allow for successively better estimation of the nonzero coefficients of Laplacian of features. Even though the iteration may find inaccurate feature estimates with an inappropriate sparsity parameter, the largest coefficients of $Lh$ are most likely to be identified as nonzero. As shown in Figure 11, the solutions $\hat{h}$ from the weighted $\ell_1$-analysis optimization (14) with different $\tau = 0.1, 1, 5$ provide accurate and reliable location of features. Once these locations are identified, their influence is eliminated by modifying the corresponding rows in the Laplacian matrix. Then it allows more sensitivity for identifying the remaining features whose Laplacian coefficients are small but nonzero.

In our implementation, we set a default value of $\tau = 0.01 \times n$ where $n$ is the sample size which works quite well for most models tested in our experiments. We also allow the user to adjust $\tau$. As $\tau$ constrains the sparsity of the features detected from the residual, it is very intuitive for the user to adjust its value. If the input mesh is supposed to have lots of features, the user can set a slightly larger value of $\tau$ to recover features faster. However, large $\tau$ likely returns fake features in the results.

Timing. The most time consuming part in our algorithm is the computation of the eigenvalues of the matrix $M$ in the denoising phase (Section 4). Although we have already proposed some techniques for speeding up its computation, it still takes lots of time for large size of data. The phase of feature recovering is very fast and can be ignored according to the computation time of the first phase. Generally, the total running time is about 10-50 seconds for a mesh with $5k-10k$ vertices.

Limitations. Our approach relies on the residual between the base mesh and the input noisy mesh to identify and discover sharp features. Our denoising scheme can be applied to either meshes or point clouds. The advantage of denoising a mesh rather than a point cloud, is that the connectivity information implicitly defines the surface topology and serves as a means for fast access to neighboring samples. Thus if we have the correct topology of the underlying surface, our scheme is very robust to noise, even for very heavy
Figure 9: Comparisons with previous approaches. From left to right: the ground truth Fandisk model, the models corrupted by different levels of artificial noise ($\sigma = 0.005$ in the upper row and $\sigma = 0.01$ in the lower row), denoised results with vertex based bilateral mesh filtering [Fleishman et al. 2003], normal based bilateral mesh filtering [Zheng et al. 2010], and our approach. The small regions with the frames are magnified to clearly show the differences. All meshes are flat-shaded to show faceting.

Figure 10: Applying our approach to real point clouds. The second row is the scanning raw data and the third row is the denoised results using our method. Their close-up views are shown in the first and the fourth rows respectively.

noise as shown in Figure 7. For point clouds, it is not easy to get the correct topology information. In our implementation, we adopt the nearest neighbors to build the graph of the points. Thus, if the graph incorrectly reflects the topology, the result might not be reliable. This is a fundamental limitation of any point-based processes.

Our $\ell_1$-analysis based feature recovering scheme relies on the assumption that the noises are i.i.d. random variables, which are correct for most of the noise and in most of the previous papers. However, if the noise is not i.i.d., our algorithm may fail.
7 Conclusion and Future Work

This paper presents a two-phase approach for decoupling features and noises on discrete surfaces. The first phase generates a base mesh which is obtained by denoising the input data using a global Laplacian regularization smoothing optimization. The smoothness parameter is automatically chosen by adopting the generalized cross validation scheme and is proved to be asymptotically optimal. The second phase extracts sharp features from the residual between the base mesh and the input mesh based on an $\ell_1$-analysis compressed sensing optimization. This is achieved by our insight that the pseudo-inverse matrix of Laplacian of a mesh is a coherent dictionary for representing a $C^0$ signal on the mesh and its sharp features can be sparsely represented in this basis. We have tested our approach on a large number of mesh surfaces with various feature types. Experimental results have shown that our method can faithfully decouple noises and sharp features.

Future work. First, we would like to apply the emerging framework of $\ell_1$-analysis compressed sensing to other problems in geometry processing. Second, the framework of $\ell_1$-analysis has potentials on detecting higher-order features. The key is to design some coherent dictionaries which can sparsely represent the features. It is worthwhile to investigate it. Third, we observe that the information of a shape can be separated into two orthogonal components: a normal component, which encodes the geometric information, and a tangential component, which encodes the parameter information. Our current scheme modifies the geometry of the mesh by moving vertices along their normal directions. It is highly worthwhile to analyze and remove the noise of the input data in the parameterization domain. We believe that this is feasible but not straightforward.

References


CANDÈS, E., ELDAR, Y., NEEDELL, D., AND RANDALL, P. 2010. Compressed sensing with coheren and redundant dictio-


1 Proof of Theorem 1 in the submission

Before we prove Theorem 1 in the submission, we have some propositions.

Proposition 1.1. For any \( f \in \mathcal{H}^2(\Omega) \), there exists a matrix \( M_{11,2} \) (depending on \( \Omega \)) such that

\[
|f|_{11,2}^2 = \min_{\phi \in \mathcal{H}_0^2(\Omega)} \frac{1}{n} \phi^T M_{11,2} f \tag{1}
\]

where \( f = (f_1, \ldots, f_n)^T = (f(X_1), \ldots, f(X_n))^T \) is the vector of function values at \( \Pi = \{X_i\}_{i=1}^n \).

The proof of the above proposition can be found in standard functional analysis textbooks and thus the details are omitted.

Proposition 1.2. If \( \Omega \) is a bounded 2-dimensional manifold and \( \mu_n \) is the largest eigenvalues of matrix \( M_{11,2} \), then \( n \delta_{\text{max}}^2 \) and \( \delta_{\text{max}} \mu_n \) are both bounded from above.

Proof. We omit the proof here and will make it available in the final version.

We present one more proposition which will be useful to prove the main theorems.

Proposition 1.3. Suppose that \( \xi_{j} \leq \xi_2 \leq \xi_{j} \leq \xi_{2} m \) for \( m > 0 \) and \( j = 1, 2, \ldots \), where \( \xi_{1}, \xi_{2} > 0 \) are constants. Then we have for \( n > 0, \lambda > 0 \),

\[
\sum_{j=1}^{n} \frac{1}{(1 + \lambda \mu_j)^2} = O(\lambda^{-1/m}).
\]

Proof. We omit the proof here and will make it available in the final version.

1.1 Convergence rate

We are now ready to exhibit the Rayleigh quotient inequalities connecting the semi-norms in \( \mathcal{H}^2(\Omega) \) and their discretized version.

Lemma 1.4. Let \( \Omega \) satisfy (A.1) and \( f \neq 0 \) satisfy (A.3). Then there exists constant \( \gamma_1 > 0 \) (depending only on \( \Omega, \xi_0, \xi_2 \)) and \( \delta_0 > 0 \), such that if \( \delta_{\text{max}} \leq \delta_0 \) we have

\[
|f|_{11,0}^2 \geq \gamma_1 (|f|_{11,2}^2 + \delta_{\text{max}}^2 |f|_{11,2}^2).
\]

for any \( |f|_{11,0}^{2} \neq 0 \).

Proof. We omit the proof here and will make it available in the final version.
**Lemma 1.5.** Assume the same conditions as in Lemma 1. Then there exists constant \( \gamma_2 > 0 \) (depending only on \( \Omega, \xi_0, \xi_1 \)) and \( \delta_0 > 0 \), such that if \( \delta_{\max} \leq \delta_0 \) we have
\[
\frac{|f|_{H^2,0}}{|f|_{H^1,0}} \geq \gamma_2 (|f|_{H^1,0} + \delta_{\max} |f|_{H^2,0}),
\]
for any \( f \neq 0 \).

**Proof.** We omit the proof here and will make it available in the final version.

**Lemma 1.4 and Lemma 1.5 build a connection between the continuous semi-norms and discrete semi-norms. This enables us to study the behavior of the eigenvalues of \( M_{H^2} \) through studying the variational eigenvalue problem. Let \( \mu_1 \leq \cdots \leq \mu_n \) be the eigenvalues of \( M_{H^2} \) in ascending order. Clearly \( \{\mu_i\} \) are non-negative real numbers since the matrix \( M_{H^2} \) is semi-positive define. Next we study the behavior of these eigenvalues and show that they can be bounded by the discrete spectrum of the differential operator \( (-\Delta_{\Omega})^2 \), where \( \Delta_{\Omega} \) is the Laplacian-Beltrami operator on \( \Omega \).

**Lemma 1.6.** Let \( \Omega \) satisfy (A.1) and \( \Pi = \{X_i\}_{i=1}^n \) satisfy (A.2). Then there exist constants \( c_1, c_2 > 0 \) such that
\[
eq 1 \rho_1 \leq \mu_j \leq c_2 \rho_j,
\]
where \( \rho_1 \leq \rho_2 \leq \cdots \leq \rho_n \) are the first \( n \) eigenvalues of the variational eigenvalue problem
\[
\int_{\Omega} \phi \Delta^2_{\Omega} \psi = \rho \int_{\Omega} \phi \psi, \quad \forall \psi \in \mathcal{H}^2(\Omega).
\]

**Proof.** We omit the proof here and will make it available in the final version.

**Lemma 1.7.** Suppose \( \Omega \) satisfy (A.1). Let \( \{\mu_1 \leq \cdots \leq \mu_n\} \) be the eigenvalues of \( M_{H^2} \) in ascending order. Then there exist constants \( c_3, c_4 > 0 \) such that for \( 2 < j \leq n \) we have
\[
c_3 \rho_j^2 \leq \mu_j \leq c_4 \rho_j^2.
\]

**Proof.** We omit the proof here and will make it available in the final version.

**Theorem 1.8.** Let \( f \) be an element of \( \mathcal{H}^2(\Omega) \) and the samples satisfy
\[
y_i = f(X_i) + \epsilon_i, \quad i = 1, \ldots, n,
\]
where \( y_1, \ldots, y_n \) are the observed functional values at \( \Pi = \{X_i\}_{i=1}^n \subset \Omega \), and \( \epsilon_1, \ldots, \epsilon_n \) are i.i.d random variables with zero-mean and finite variance \( \sigma^2 > 0 \). Suppose (A.1) and (A.2) are fulfilled. Let \( \hat{f}_n(\lambda) = A_n(\lambda)y = (I_n + \lambda M_{H^2})^{-1}y \) be the estimator from the DLRS model. Denote \( r_n(\lambda) = n^{-1/2} |\hat{f}_n(\lambda) - f|^2 \). As \( n \to \infty \) and \( \lambda \sim n^{-2/3} \) is chosen, then
\[
\mathbb{E}[r_n(\lambda)] = O(n^{-\frac{2}{3}}).
\]

**Proof.** We omit the proof here and will make it available in the final version.

1.2 **Proof of Theorem 1 in the submission**

Using Theorem 1.8, we can easily prove Theorem 1 in the submission. Specifically, in the DLRS model we let the unknown function \( f \) be a \( C^2 \)-smooth surface \( S \) itself and the observation samples \( y = (y_1, \ldots, y_n)^T \) be the noisy samples of surface position \( P = (p_1, \ldots, p_n)^T \). Therefore we come to the conclusion of Theorem 1 in the submission.

2 **Proof of Theorem 2 in the submission**

We will show that the DLRS estimator satisfies some general conditions and then prove the asymptotic optimality of GCV under our proposed framework.

Let \( \hat{f}_n(\lambda) = A_n(\lambda)y = (I_n + \lambda M)^{-1}y \) be the estimator of our DLRS model and denote \( r_n(\lambda) = n^{-1/2} |\hat{f}_n(\lambda) - f|^2 \). The asymptotic optimality of GCV is defined as
\[
\inf_{\lambda \in \mathbb{R}_+} \frac{r_n(\lambda)}{\mathbb{E}[r_n(\lambda)]} \to_p 1,
\]
which verifies the closeness between the values of risk function given by the GCV choice \( \hat{\lambda} \) to the theoretically optimal choice \( \lambda^* = \arg \inf_{\lambda \in \mathbb{R}_+} r_n(\lambda) \).

The main result here is to show that our estimator satisfies the following three conditions.

(C.1) \( \inf_{\lambda \in \mathbb{R}_+} n\mathbb{E}[r_n(\lambda)] \to \infty \).

(C.2) There exists a sequence \( \{\lambda_n\} \) such that \( r_n(\lambda_n) \to_p 0 \) (the convergence in probability).

(C.3) Let \( 0 \leq \kappa_1 \leq \cdots \leq \kappa_n \) be the eigenvalues of \( K(\lambda) = \lambda M \).

For any \( \ell \) such that \( \ell n \to 0 \), then
\[
\frac{(n^{-1} \sum_{i=\ell+1}^{\kappa_\ell} \kappa_i^{-1})^2}{\sum_{i=\ell+1}^{\kappa_\ell} \kappa_i} \to 0
\]
as \( n \to \infty \).

The condition (C.1) states that the convergence rate of the risk function to zero should be lower than \( O(n^{-1}) \). Otherwise the estimates may possess unattainably small risk.

Denote \( \text{null}(\Delta_{\Omega}) \) the null space of Laplacian operator \( \Delta_{\Omega} \). Actually from the behavior of eigenvalues as shown in Lemma 1.7, it is not difficult to verify that our proposed model meets the condition (C.1) except for \( f \in \text{null}(\Delta_{\Omega}) \).

**Lemma 2.1.** If \( f \notin \text{null}(\Delta_{\Omega}) \), the estimator \( \hat{f}_n(\lambda) \) from our DLRS model holds
\[
\inf_{\lambda \in \mathbb{R}_+} n\mathbb{E}[r_n(\lambda)] \to \infty.
\]

This verifies the condition (C.1).

**Proof.** We omit the proof here and will make it available in the final version.

**Lemma 2.2.** Under condition (C.1), we have in probability
\[
\sup_{\lambda > 0} \left| \frac{r_n(\lambda)}{\mathbb{E}[r_n(\lambda)]} - 1 \right| \to 0.
\]

**Proof.** We omit the proof here and will make it available in the final version.

The condition (C.2) shows that the risk function \( r_n(\lambda_n) \) converge to zero in probability with appropriate sequence \( \{\lambda_n\} \). Obviously, the conclusion of condition (C.2) can be easily derived from Theorem 1.8 and Lemma 2.2. Therefore, the condition (C.2) holds true.

The condition (C.3) gives a ratio
\[
\frac{(n^{-1} \sum_{i=\ell+1}^{\kappa_\ell} \kappa_i^{-1})^2}{\sum_{i=\ell+1}^{\kappa_\ell} \kappa_i},
\]
which is defined on the eigenvalues of \( K(\lambda) = \lambda M \) and often plays an important role in the asymptotic analysis.
Lemma 2.3. In our model, for any $\ell$ such that $\frac{\ell}{n} \to 0$ and $\kappa_{\ell+1} > 0$, then the ratio of (7) converges to zero as $n$ (the sample size) goes to infinity. This verifies the condition (C.3).

Proof. We omit the proof here and will make it available in the final version.

2.1 Asymptotic optimality theorem

By conclusion, we have verified that the three conditions (C.1), (C.2), and (C.3) hold true for our model. Then we will prove the asymptotic optimality of GCV under these three conditions.

Lemma 2.4. Under the condition (C.2), we have

\[ n^{-1} \text{tr}[I_n - A_n(\lambda_n)] \to 1, \tag{8} \]

and

\[ n^{-1} \| (I_n - A_n(\lambda_n)) y \|^2 \to \sigma^2. \tag{9} \]

Proof. We omit the proof here and will make it available in the final version.

Lemma 2.5. Under the condition (C.3), for $\lambda_n$ such that $r_n(\lambda_n) \to 0$, we have

\[ \frac{\left( n^{-1} \text{tr}[A_n(\lambda_n)] \right)^2}{n^{-1} \text{tr}[A_n(\lambda_n^*)]^2} \to 0. \tag{10} \]

Proof. We omit the proof here and will make it available in the final version.

Lemma 2.6. For any $\hat{\lambda}$ such that $r_n(\hat{\lambda}) \to 0$ and

\[ \frac{\left( n^{-1} \text{tr}[A_n(\hat{\lambda})] \right)^2}{n^{-1} \text{tr}[A_n(\hat{\lambda})^2]} \to 0, \tag{11} \]

under the condition (C.1) we have

\[ \frac{\text{SURE}_n(\hat{\lambda}) - \hat{r}_n(\hat{\lambda}) - n^{-1} \| e \|^2 + \sigma^2}{r_n(\lambda)} \to_p 0, \tag{12} \]

and

\[ \frac{n^{-1} \| \hat{\ell}_n(\hat{\lambda}) - \hat{\ell}_n(\hat{\lambda}) \|^2}{r_n(\lambda)} \to_p 0, \tag{13} \]

where $\text{SURE}_n(\lambda) = \sigma^2 - \sigma^4 \frac{(n^{-1} \text{tr}[I_n - A_n(\lambda)])^2}{n^{-1} \| (I_n - A_n(\lambda)) y \|^2}$, $\hat{\ell}_n(\lambda) = y - \sigma^2 \frac{\text{tr}[I_n - A_n(\lambda)]}{\| (I_n - A_n(\lambda)) y \|^2} (I_n - A_n(\lambda)) y$ and $\hat{r}_n(\lambda) = n^{-1} \| \ell_n(\lambda) - \ell \|^2$.

Proof of the Lemma 2.6 is omitted here and will be available in the final version.

Lemma 2.7. Under conditions (C.2) and (C.3), $\hat{\ell}_n(\hat{\lambda}_G)$ is consistent, i.e., $r_n(\hat{\lambda}_G) \to 0$, where $\hat{\lambda}_G$ is chosen by GCV.

Proof of the Lemma 2.7 is omitted and will be available in the final version.

2.2 Proof of Theorem 2 in the submission

Theorem 2.8. Under conditions (C.1), (C.2) and (C.3), $\hat{\ell}_n(\hat{\lambda}_G)$ is asymptotically optimal, where $\hat{\lambda}_G$ is the GCV choice.

Proof. From the condition (C.2), for $\lambda_n^*$ that is the minimizer of $r_n(\lambda)$, we have $r_n(\lambda_n^*) \to 0$. According to Lemma 2.5, we have

\[ \frac{\left( n^{-1} \text{tr}[A_n(\lambda_n^*)] \right)^2}{n^{-1} \text{tr}[A_n(\lambda_n^*)^2]} \to 0. \tag{14} \]

Hence from Lemma 2.6, we have $\text{SURE}_n(\lambda_n^*) - n^{-1} \| e_n \|^2 + \sigma^2 = r_n(\lambda_n^*)(1 + o_p(1))$.

On the other hand, from Lemma 2.7 this also holds for $\hat{\lambda} = \hat{\lambda}_G$. Therefore we have

\[ \text{SURE}_n(\hat{\lambda}_G) - n^{-1} \| e_n \|^2 + \sigma^2 = r_n(\hat{\lambda}_G)(1 + o_p(1)) \tag{15} \]

Since $\text{SURE}_n(\hat{\lambda}_G) \leq \text{SURE}_n(\lambda_n^*)$ and $r_n(\lambda_n^*) \leq r_n(\hat{\lambda}_G)$, we have $r_n(\hat{\lambda}_G)/r_n(\lambda_n^*) \to 1$ in probability.

References


