Inverse Problem In Imaging
Assignment 2

Idan Grady

1

It is given that $v(1,x) = \sum_{k=1}^{\infty} a_k \exp{(-k^2)} sin(kx)$
and $a_k = < u, sin(\dot{k}) >$. Hence, finding the singular system of K is trivial.

$$Ku(x) = \sum_{0}^{\infty} (\exp{(-k^2)})(< u, sin(k) >)(sin(kx))$$

$$\sigma = \exp{(-k^2)}$$
$$< u, v_k >= a_k$$
$$u_k(x) = sin(kx)$$

2

2.1 (finite)

Lets start with the first option.

$$
\begin{aligned}
\min_u ||Ku - f||^2 + aR(u) = \\
& \min_u ||Ku - f||^2 + a||u||^2 \\
K = U\Sigma V^T \\
U = C_k * U_k \\
& (Ku - f)^T(Ku - f) + a||u||^2 = \\
& (Ku)^2 - 2K^T uf + f^2 + a||u||^2 = \\
& 0 = (K^T Ku) - K^T f + au = \\
& (K^T Ku) - K^T f + au = \\
& (V\Sigma U^T U\Sigma V^T + aI)u - (V\Sigma U^T)f = \\
& (V\Sigma^2 V^T + aI)u - (V\Sigma U^T)f = \\
V = V^{-1} \\
VV^T = I \\
& V(\Sigma^2 + aI)V^T u - (V\Sigma U^T)f = \\
& (\Sigma^2 + aI)V^T u - (\Sigma U^T)f \\
& u = \frac{\Sigma}{\Sigma^2 + aI} VU^T f \\
& K^\dagger f = \sum^{\infty} \frac{1}{\sigma + a\sigma^{-1}} < f, u_k > v_k(x)
\end{aligned}
$$

2.1.2 (Infinite)

$$K = \sum_{k=1}^{\infty} \sigma_k \langle \cdot, u_k \rangle v_k$$

$$u(x) = \sum_{k=1}^{\infty} a_k u_k(x)$$

$$\min_u ||Ku - f||^2 + a||u||^2$$

$$\min_{a_k} \left\| \sum_{k=1}^{\infty} \sigma_k \langle u, u_k \rangle v_k - f \right\|^2 + a \left\| \sum_{k=1}^{\infty} a_k u_k \right\|^2$$

$$\min_{a_k} \left\| \sum_{k=1}^{\infty} \sigma_k \langle \sum_{j=1}^{\infty} a_j u_j, u_k \rangle v_k - f \right\|^2 + a \left\| \sum_{k=1}^{\infty} a_k u_k \right\|^2$$

$$\min_{a_k} \left\| \sum_{k=1}^{\infty} \sigma_k a_k v_k - f \right\|^2 + a \left\| \sum_{k=1}^{\infty} a_k u_k \right\|^2$$

$$a_k = \frac{\sigma_k \langle f, v_k \rangle}{\sigma_k^2 + a}$$

$$u^* = \sum_{k=1}^{\infty} a_k u_k = \sum_{k=1}^{\infty} \frac{\sigma_k \langle f, v_k \rangle}{\sigma_k^2 + a} u_k$$

and hence,

$$K_a^{\dagger} f = \sum_{k=1}^{\infty} \frac{\sigma_k}{\sigma_k^2 + a} \langle f, v_k \rangle u_k(x)$$

2.2

Using the fact that v_k form an orthonormal basis, we can express u(x) as: $u(x) = \sum^{\infty} a_k v_k(x)$ because v_k is constant, we would need to differentiate with respect to a_k.

$$\min_u ||Ku - f||^2 + a||u'||^2 =$$

$$\min_{a_k} ||K(\sum^{\infty} a_k v_k) - f||^2 + a||(\sum^{\infty} a_k' v_k)||^2 =$$

$$2(K(\sum^{\infty} a_k v_k) - f) * K v_k + 2a(\sum^{\infty} a_k v_k) * v_k$$

$$0 = 2(\frac{a_k}{\sigma_k} - < f, u_k >) + 2a(a_k k^2)$$

$$a_k = \frac{< f, u_k >}{\sigma_k + a k^2 \sigma^{-1}}$$

Hence, when substituting it

$$K_a^{\dagger} f = \sum^{\infty} \frac{< f, u_k >}{\sigma_k + ak^2\sigma^{-1}} v_k$$

3

Recall that in order to satisfied the Picard Condition we need

$$\sum^{\infty} \frac{| < f, u_k > |^2}{\sigma_k^2} < \infty \tag{1}$$

Where $| < f, u_k >$ is the generlized Fourier coefficients noted as f_k. Before, lets check whether there is a need for
we have calculated in excersie 1 $Ku(x)$ and found that $\sigma = \exp{(-k^2)}$. Hence

$$K_a^{\dagger} f = \sum_{k=0}^{\infty} \frac{|f_k|^2}{\sigma_k} v_k(x)$$

3.1

we can clearly see that

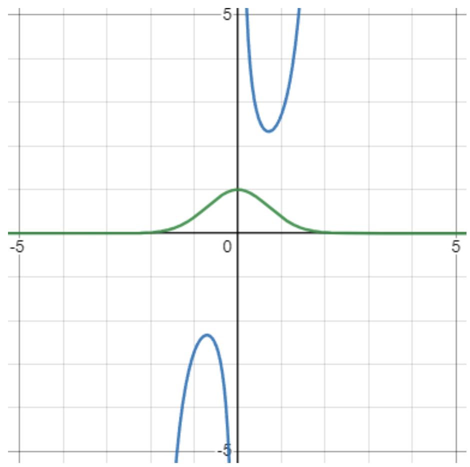
$$\sum_{k=0}^{\infty} \frac{|exp(-2k^2)|^2}{exp(-k^2)} = \sum_{k=0}^{\infty} \frac{exp(-4k^2)}{exp(-k^2)} < \infty$$

Which means that we meet the Picard Condition and do not need regularisation.

3.2

when $f_k = k^{-1}$, we see that we violate the Picard condition. To visually see the difference

Figure 1: Green: $f_k = \exp(-2k^2)$ and Blue: $f_k = k^{-1}$



Hence, we can see that we do not converge. Mathematically speaking this can be reasoned when comparing once again the

$$\frac{f_k}{\sigma_k} < 1$$
$$k^{-1} > \exp(-k^2)$$

Hence, we would check the two regularisation options when

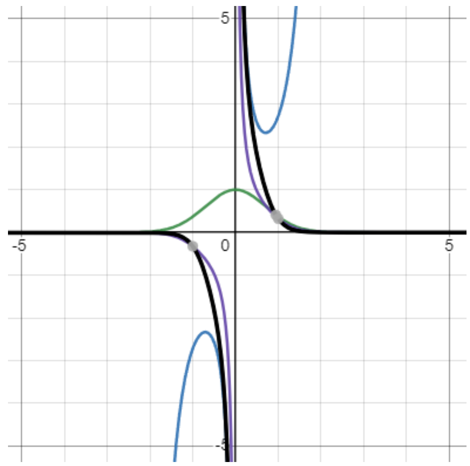
- $R(u) = ||u||^2$
- $R(u) = ||u'||^2$

We have proven that their correspondent σ_k are given as (substitute $a = 1$)

- $R(u) = ||u||^2 \rightarrow \sigma_k = \frac{1}{\sigma_k + \sigma_k^{-1}} = \frac{1}{\exp(-k^2) + \exp(k^2)}$
- $R(u) = ||u'||^2 \rightarrow \sigma_k = \frac{1}{\sigma_k + k^2\sigma^{-1}} = \frac{1}{\exp(-k^2) + k^2 \exp(k^2)}$

Just by the first glace, we can clearly see that the $||u'||^2 <= ||u||^2$ which should result in either in a longer convergence or even in some specific different cases of f_k could be unstable. To visually observe all 4 options

Figure 2: Green: $f_k = \exp(-2k^2)$ and Blue: $f_k = k^{-1}$
Lila: $R(u) = ||u||^2$black: $R(u) = ||u'||^2$



Hence, we can see that both regularisation would converge.

4

For Bias variance decomposition lets remind that

$$||K_a^\dagger f^\delta - K^\dagger f||_u \leq ||K_a^\dagger f - K^\dagger f||_u + ||K_a^\dagger f^\delta - K_a^\dagger f||_u \tag{2}$$

Also, lets observe the difference between the true f and f^δ for different (amplitude) noise level.

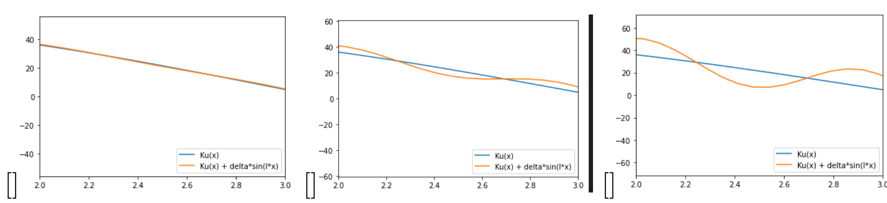


Figure 3: Different when delta = [0.5,5,15]

Lets observe the differences of regularisation when integrating over deltas. To note the delta represents the noise amplitude.

(I didnt fully get what you meant with you suggestions, so due to lack of time, I submit it as I derived). Note: I compared only the alphas.

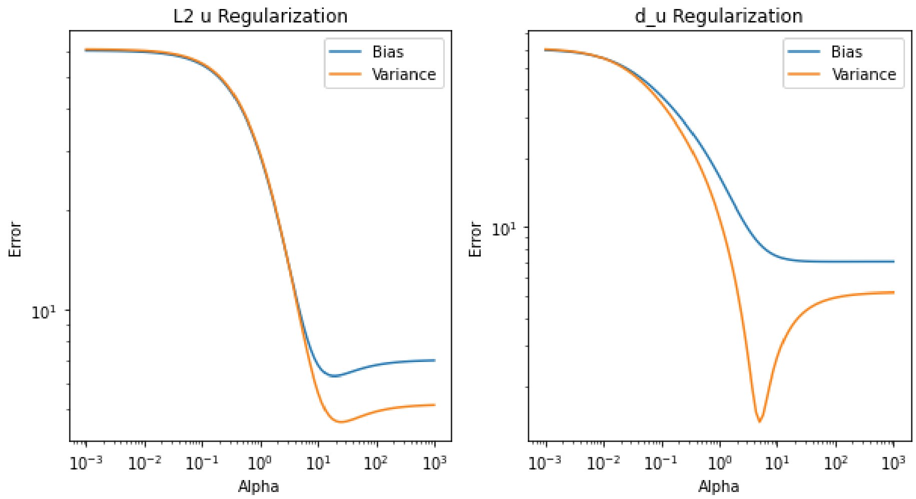


Figure 4: Different in regularisation over alphas

The figures show that the bias and variance for L2 regularization are lower than those for $||u'||^2$ regularization across a wide range of alpha values. When compared to $||u'||^2$ regularization, L2 regularization achieves a better balance between minimizing overfitting (variance) and underfitting (bias). It is also noticeable that the alpha that minimizes the error for both the bias and the variance appears to be different. where is less in $||u'||^2$ than in $||u||^2$ based regularisation.

Code

```
import numpy as np
import matplotlib.pyplot as plt

num_points = 100

def u(x, k):
    return np.sin(k * x)

def v(x, k):
    return np.sin(k * x)

def f_delta(x, k, l, delta):
    return Ku(x, k) + delta * np.sin(l * x)

def Ku(x, k):
    return sum(sigma_k(k) * inner_product(u(x, k), v(x, k), x) * u(
                                      x, k) for k in range(1,
                                      num_points))

def sigma_k(k):
    return np.exp(-k**2)
```

```
def inner_product(u, v, x):
    return (2 / np.pi) * np.trapz(u * v, x)

def k_a_L2(x, f_delta, v, u_k, sigma_k, alpha, k_max):
    return sum((1 / (sigma_k(-k) + alpha * sigma_k(k))) *
                                     inner_product(f_delta, v(x, k
                                     ), x) * u_k(x, k) for k in
                                     range(1, k_max+1))

def k_a_dL2(x, f_delta, v, u_k, sigma_k, alpha, k_max):
    return sum((1 / (sigma_k(-k) + alpha * k**2 * sigma_k(k))) *
                                     inner_product(f_delta, v(x, k
                                     ), x) * u_k(x, k) for k in
                                     range(1, k_max+1))

k = 5
l = 7
delta = 1
num_points = 100
alpha_values = np.logspace(-3, 3, num_points)
x = np.linspace(0, 2 * np.pi, num_points)

bias_u = []
variance_u = []

bias_du = []
variance_du = []

deff_var = []
def_bias = []

for alpha in alpha_values:
    f_delta_current = f_delta(x, k, l, delta)

    u_alpha = k_a_L2(x, f_delta_current, v, u, sigma_k, alpha, k)
    bias = np.linalg.norm(u_alpha - u(x, k))
    variance = np.linalg.norm(u_alpha - Ku(x, k))

    bias_u.append(bias)
    variance_u.append(variance)

    u_alpha_d = k_a_dL2(x, f_delta_current, v, u, sigma_k, alpha, k
                                     )
    bias_d = np.linalg.norm(u_alpha_d - u(x, k))
    variance_d = np.linalg.norm(u_alpha_d - Ku(x, k))

    bias_du.append(bias_d)
    variance_du.append(variance_d)


plt.figure(figsize=(10, 5))

plt.subplot(121)
plt.loglog(alpha_values, bias_u, label='Bias')
plt.loglog(alpha_values, variance_u, label='Variance')
plt.xlabel('Alpha')
```

```
plt.ylabel('Error')
plt.title('L2 u Regularization')
plt.legend()

plt.subplot(122)
plt.loglog(alpha_values, bias_du, label='Bias')
plt.loglog(alpha_values, variance_du, label='Variance')
plt.xlabel('Alpha')
plt.ylabel('Error')
plt.title('d_u Regularization')
plt.legend()

plt.show()
```