Fisher Discrimination

> Pattern Recognition Homeworks
>
> Student: thu-zxs

Solutions

Problem 1

1.1

Solve the convex optimization problem:

$$
\begin{aligned}
\min_{w_0} f(\mathbf{w}, w_0) &= \frac{1}{2}\sum_{n=1}^{N}(\mathbf{w}^T\mathbf{x}_n + w_0 - t_n)^2 \\
\frac{\partial f(\mathbf{w}, w_0)}{\partial w_0} &= \sum_{n=1}^{N}(\mathbf{w}^T\mathbf{x}_n + w_0 - t_n) \\
&= 0
\end{aligned}
$$

yield

$$
w_0 = \frac{1}{N}\sum_{n=1}^{N}(-\mathbf{w}^T\mathbf{x}_n + t_n) = -\mathbf{w}^T\mathbf{m} + \frac{1}{N}\sum_{n=1}^{N} t_n
$$

by definition,

$$
\sum_{n=1}^{N} t_n = N_1 * \frac{N}{N_1} + N_2 * (-\frac{N}{N_2}) = 0
$$

so

$$
w_0 = -\mathbf{w}^T\mathbf{m}.
$$

1.2

Solve the convex optimization problem:

$$
\min_{\mathbf{w}} f(\mathbf{w}, w_0) = \frac{1}{2}\sum_{n=1}^{N}(\mathbf{w}^T\mathbf{x}_n - \mathbf{w}^T\mathbf{m} - t_n)^2
$$

$$
\begin{aligned}
\frac{\partial f(\mathbf{w}, w_0)}{\partial \mathbf{w}} &= \sum_{n=1}^{N} (\mathbf{w}^T\mathbf{x}_n - \mathbf{w}^T\mathbf{m} - t_n) * \mathbf{x}_n \\
&= \sum_{n=1}^{N} ((\mathbf{x}_n\mathbf{x}_n^T)\mathbf{w} - (\mathbf{x}_n\mathbf{m}^T)\mathbf{w} - \mathbf{x}_n t_n) \\
&= (\sum_{n=1}^{N} \mathbf{x}_n\mathbf{x}_n^T)\mathbf{w} - (\sum_{n=1}^{N} \mathbf{x}_n\mathbf{m}^T)\mathbf{w} - \sum_{n=1}^{N} \mathbf{x}_n t_n \\
&= 0
\end{aligned}
$$

where

$$
\begin{aligned}
&\sum_{n=1}^{N} \mathbf{x}_n\mathbf{x}_n^T - \sum_{n=1}^{N} \mathbf{x}_n\mathbf{m}^T \\
&= \sum_{n\in C_1} \mathbf{x}_n\mathbf{x}_n^T + \sum_{n\in C_2} \mathbf{x}_n\mathbf{x}_n^T - \frac{1}{N}\sum_{n=1}^{N} \mathbf{x}_n(\sum_{n=1}^{N} \mathbf{x}_n^T) \\
&= \sum_{n\in C_1} \mathbf{x}_n\mathbf{x}_n^T + \sum_{n\in C_2} \mathbf{x}_n\mathbf{x}_n^T - \frac{1}{N}(\sum_{n\in C_1} \mathbf{x}_n + \sum_{n\in C_2} \mathbf{x}_n)(\sum_{n\in C_1} \mathbf{x}_n + \sum_{n\in C_2} \mathbf{x}_n)^T \\
&= \sum_{n\in C_1} \mathbf{x}_n\mathbf{x}_n^T + \sum_{n\in C_2} \mathbf{x}_n\mathbf{x}_n^T + \frac{N_1N_2}{N}(-\mathbf{m}_1\mathbf{m}_2^T - \mathbf{m}_2\mathbf{m}_1^T + \mathbf{m}_1\mathbf{m}_1^T + \mathbf{m}_2\mathbf{m}_2^T) - N_1\mathbf{m}_1\mathbf{m}_1^T - N_2\mathbf{m}_2\mathbf{m}_2^T \\
&= \frac{N_1N_2}{N}(S_B) + \sum_{n\in C_1} \mathbf{x}_n\mathbf{x}_n^T - \sum_{n\in C_1} \mathbf{x}_n\mathbf{m}_1^T + \sum_{n\in C_2} \mathbf{x}_n\mathbf{x}_n^T - \sum_{n\in C_2} \mathbf{x}_n\mathbf{m}_2^T \\
&= \frac{N_1N_2}{N}(S_B) + \sum_{n\in C_1} \mathbf{x}_n\mathbf{x}_n^T - \sum_{n\in C_1} \mathbf{x}_n\mathbf{m}_1^T - \sum_{n\in C_1} -\mathbf{m}_1\mathbf{x}_n^T + \sum_{n\in C_1} \mathbf{m}_1\mathbf{m}_1^T \\
&+ \sum_{n\in C_2} \mathbf{x}_n\mathbf{x}_n^T - \sum_{n\in C_2} \mathbf{x}_n\mathbf{m}_2^T - \sum_{n\in C_2} -\mathbf{m}_2\mathbf{x}_n^T + \sum_{n\in C_2} \mathbf{m}_2\mathbf{m}_2^T \\
&= \frac{N_1N_2}{N}(S_B) + S_W
\end{aligned}
$$

and

$$
\sum_{n=1}^{N} \mathbf{x}_n t_n = \sum_{n\in C_1} \mathbf{x}_n \frac{N}{N_1} - \sum_{n\in C_2} \mathbf{x}_n \frac{N}{N_2} = N(\mathbf{m}_1 - \mathbf{m}_2)
$$

so the equation

$$
(\frac{N_1N_2}{N}(S_B) + S_W)\mathbf{w} = N(\mathbf{m}_1 - \mathbf{m}_2)
$$

is derived.

1.3

from equation (4),

$$
\frac{N_1N_2}{N}S_B\mathbf{w} = \frac{N_1N_2}{N}(\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^T\mathbf{w} = \frac{N_1N_2}{N}(\mathbf{m}_2 - \mathbf{m}_1)R,
$$

where R is a scalar. So we have

$$S_W\mathbf{w} = (-\frac{N_1N_2}{N}R - N)(\mathbf{m}_1 - \mathbf{m}_2)$$

thus

$$\mathbf{w} = (-\frac{N_1N_2}{N}R - N)S_W^{-1}(\mathbf{m}_1 - \mathbf{m}_2) \propto S_W^{-1}(\mathbf{m}_1 - \mathbf{m}_2),$$

resulting in the same form as that of fisher criterion.

Problem2

2.1

$$\begin{aligned}
S_B + S_W &= \sum_{k=1}^{K} N_k(\mathbf{m}_k - \mathbf{m})(\mathbf{m}_k - \mathbf{m})^T + \sum_{k=1}^{K}\sum_{n\in C_k}(\mathbf{x}_n - \mathbf{m}_k)(\mathbf{x}_n - \mathbf{m}_k)^T \\
&= \sum_{k=1}^{K}\sum_{n\in C_k}(\mathbf{m}_k - \mathbf{m})(\mathbf{m}_k - \mathbf{m})^T + \sum_{k=1}^{K}\sum_{n\in C_k}(\mathbf{x}_n - \mathbf{m}_k)(\mathbf{x}_n - \mathbf{m}_k)^T \\
&= \sum_{k=1}^{K}\sum_{n\in C_k}((\mathbf{m}_k - \mathbf{m})(\mathbf{m}_k - \mathbf{m})^T + (\mathbf{x}_n - \mathbf{m}_k)(\mathbf{x}_n - \mathbf{m}_k)^T) \\
&= \sum_{k=1}^{K}\sum_{n\in C_k}((\mathbf{m}_k - \mathbf{m})(\mathbf{m}_k^T - \mathbf{m}^T) + (\mathbf{x}_n - \mathbf{m}_k)(\mathbf{x}_n^T - \mathbf{m}_k^T)) \\
&= \sum_{k=1}^{K}\sum_{n\in C_k}((\mathbf{m}_k\mathbf{m}_k^T - \mathbf{m}\mathbf{m}_k^T - \mathbf{m}_k\mathbf{m}^T + \mathbf{m}\mathbf{m}^T) + (\mathbf{x}_n\mathbf{x}_n^T - \mathbf{m}_k\mathbf{x}_n^T - \mathbf{x}_n\mathbf{m}^T + \mathbf{m}_k\mathbf{m}_k^T)) \\
&= (\sum_{k=1}^{K}\sum_{n\in C_k}\mathbf{x}_n\mathbf{x}_n^T) - \mathbf{m}\mathbf{m}^T
\end{aligned}$$

$$\begin{aligned}
S_T &= \sum_{n=1}^{N}(\mathbf{x}_n - \mathbf{m})(\mathbf{x}_n - \mathbf{m})^T \\
&= \sum_{k=1}^{K}\sum_{n\in C_k}(\mathbf{x}_n - \mathbf{m})(\mathbf{x}_n - \mathbf{m})^T \\
&= \sum_{k=1}^{K}\sum_{n\in C_k}(\mathbf{x}_n\mathbf{x}_n - \mathbf{m}\mathbf{x}_n^T + \mathbf{x}_n\mathbf{m}^T + \mathbf{m}\mathbf{m}^T) \\
&= (\sum_{k=1}^{K}\sum_{n\in C_k}\mathbf{x}_n\mathbf{x}_n) - \mathbf{m}\mathbf{m}^T
\end{aligned}$$

So $S_T = S_W + S_B$

2.2

$$
\begin{aligned}
s_W &= \sum_{k=1}^{K}\sum_{n\in C_k}(\mathbf{y}_n-\tilde{\mathbf{m}}_k)(\mathbf{y}_n-\tilde{\mathbf{m}}_k)^T \\
&= \sum_{k=1}^{K}\sum_{n\in C_k}(\mathbf{W}^T\mathbf{x}_n-\mathbf{W}^T\mathbf{m}_k)(\mathbf{W}^T\mathbf{x}_n-\mathbf{W}^T\mathbf{m}_k)^T \\
&= \sum_{k=1}^{K}\sum_{n\in C_k}\mathbf{W}^T(\mathbf{x}_n-\mathbf{m}_k)(\mathbf{x}_n-\mathbf{m}_k)^T\mathbf{W} \\
&= \mathbf{W}^T S_W \mathbf{W}
\end{aligned}
$$

$$
\begin{aligned}
s_B &= \sum_{k=1}^{K} N_k(\tilde{\mathbf{m}}_k-\tilde{\mathbf{m}})(\tilde{\mathbf{m}}_k-\tilde{\mathbf{m}})^T \\
&= \sum_{k=1}^{K} N_k\mathbf{W}^T(\mathbf{m}_k-\mathbf{m})(\mathbf{m}_k-\mathbf{m})^T\mathbf{W} \\
&= \mathbf{W}^T S_B \mathbf{W}
\end{aligned}
$$

2.3

$$
\begin{aligned}
(s_W)_{11} &= \sum_{k=1}^{K}\sum_{n\in C_k}(\mathbf{W}^T(\mathbf{x}_n-\mathbf{m}_k))_1((\mathbf{x}_n-\mathbf{m}_k)^T\mathbf{W})_1 \\
&= \sum_{k=1}^{K}\sum_{n\in C_k}(\mathbf{w}_1^T(\mathbf{x}_n-\mathbf{m}_k))((\mathbf{x}_n-\mathbf{m}_k)^T\mathbf{w}_1) \\
&= \mathbf{w}_1^T(\sum_{k=1}^{K}\sum_{n\in C_k}(\mathbf{x}_n-\mathbf{m}_k)(\mathbf{x}_n-\mathbf{m}_k)^T)\mathbf{w}_1
\end{aligned}
$$

$$
J(\mathbf{W}) = \frac{\prod_i^{D'}\mathbf{w}_i^T S_B \mathbf{w}_i}{\prod_i^{D'}\mathbf{w}_i^T S_W \mathbf{w}_i}
$$

where $\mathbf{w_i}$ is the i-th column of $\mathbf{W}$

2.4

By Rayleigh quotient, we maximize:

$$
\max_{\mathbf{W}}\prod_i^{D'}\mathbf{w}_i^T S_B\mathbf{w}_i-\lambda(\prod_i^{D'}\mathbf{w}_i^T S_W\mathbf{w}_i-C)
$$

By solving this problem, each $\mathbf{w}_i^*$ satisfies:

$$
S_W^{-1}S_B\mathbf{w}_i^* = \lambda_i\mathbf{w}_i^*
$$

which means D' eigenvectors of $S_W^{-1}S_B$ sorted by descend order of D' eigenvalue.

2.5

By definiton, S_B is $rank(K-1)$, so $S_W^{-1}S_B$ has atmost $K-1$ non-zeros eigenvalue. So we have the opportunity to find atmost $K-1$ features.

Programming

3.1

Consider the least-square part of $J(\mathbf{w};\lambda)$:

$$f(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}(y_i - \mathbf{w}^T\phi(\mathbf{x}_i))^2$$

$$\frac{\partial f}{\partial \mathbf{w}} = \frac{1}{n}\sum_{i=1}^{n}-y_i\phi(x_i) + \frac{1}{n}(\sum_{i=1}^{n}\mathbf{w}^T\phi(x_i))\phi(x_i)$$

$$\begin{aligned}
\frac{\partial f}{\partial \mathbf{w}_k} &= (\frac{\partial f}{\partial \mathbf{w}})_k \\
&= \frac{1}{n}\sum_{i=1}^{n}-y_i\phi_k(x_i) + \frac{1}{n}(\sum_{i=1}^{n}\mathbf{w}^T\phi(x_i))\phi_k(x_i) \\
&= \frac{1}{n}\sum_{i=1}^{n}w_k\phi_k(x_i)\phi_k(x_i) + \frac{1}{n}\sum_{i=1}^{n}(\mathbf{w}_{-k}^T\phi_{-k}(x_i) - y_i)\phi_k(x_i) \\
&= a_k w_k - c_k
\end{aligned}$$

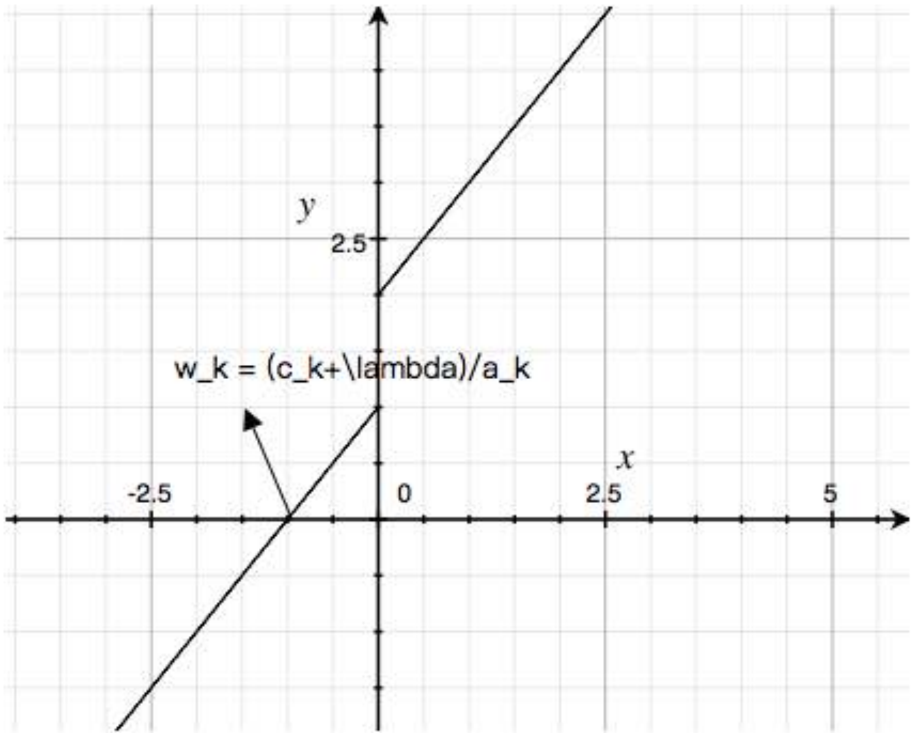
As a result of above deduction:

$$\partial_{w_k} J(\mathbf{w};\lambda) = a_k w_k - c_k + \lambda\partial_{w_k}|w_k|$$

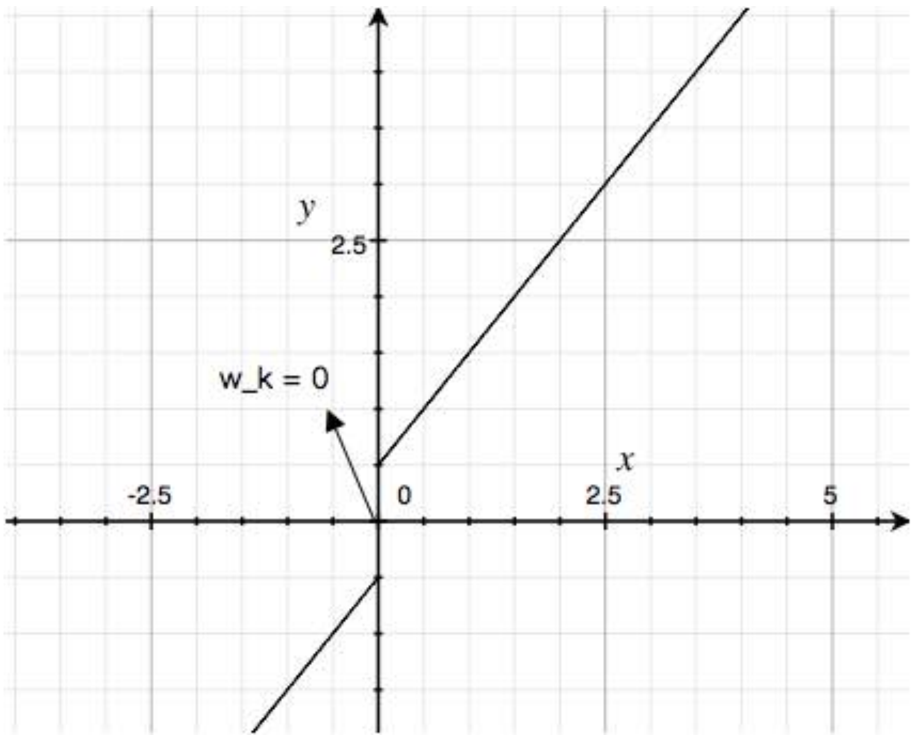
And c_k indicates the classification error when abandoning k-th feature.

3.2

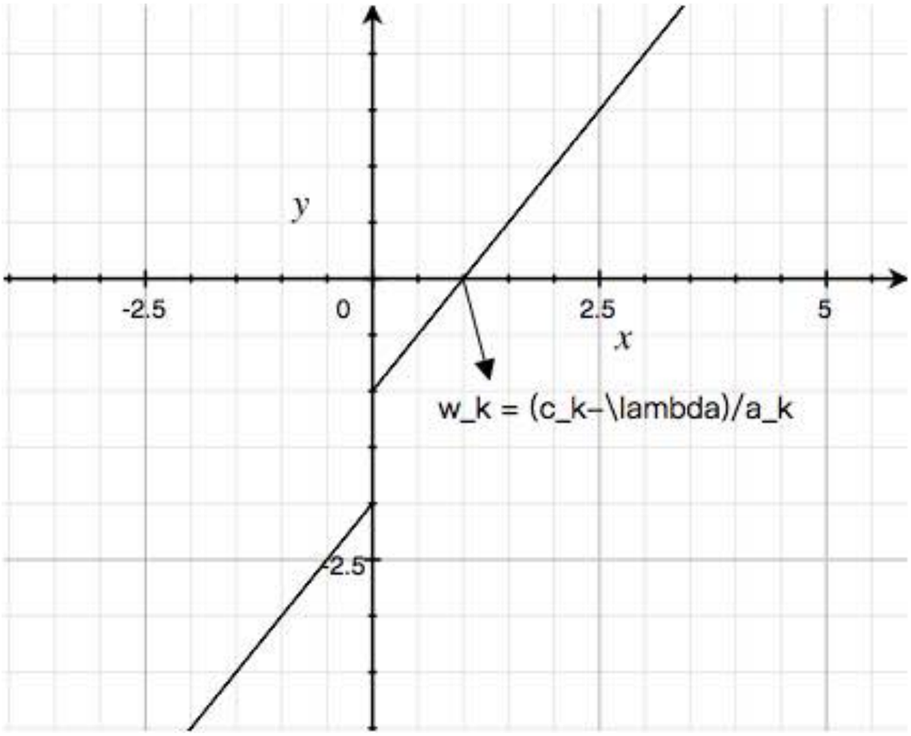
a) $c_k < -\lambda$ => $\hat{w}_k = \frac{c_k+\lambda}{a_k}$



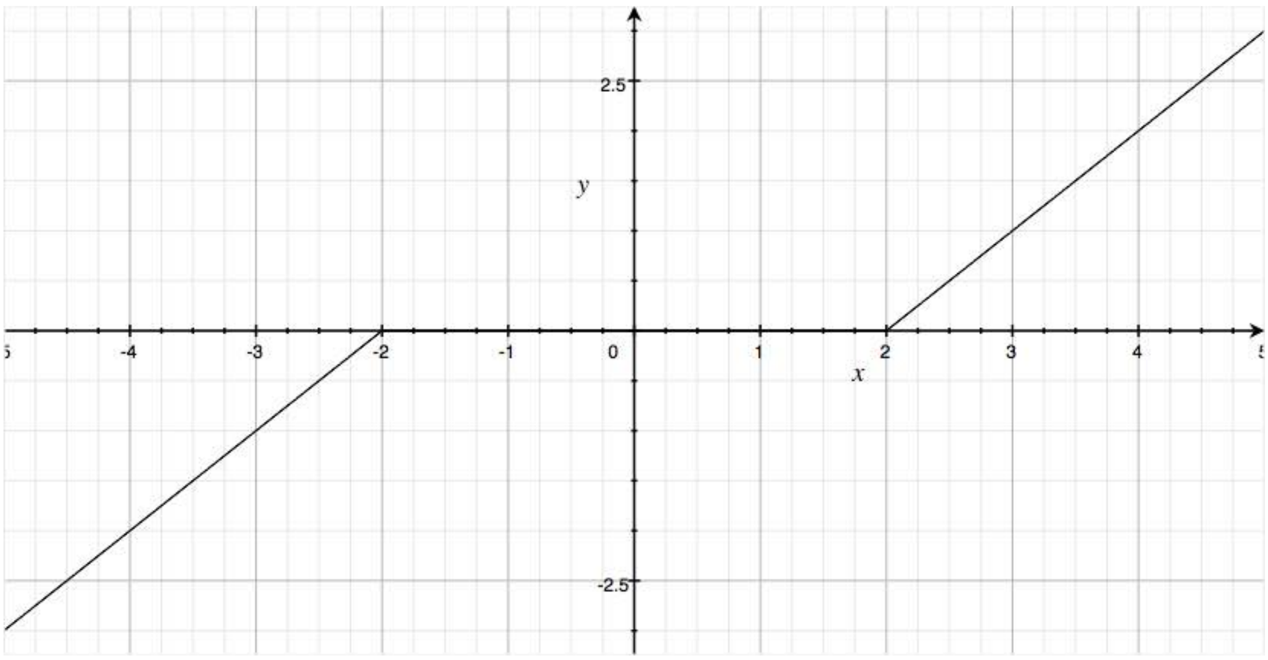
b)$-\lambda < c_k < \lambda$ => $\hat{w_k} = 0$



c) $c_k > \lambda$ => $\hat{w_k} = \frac{c_k - \lambda}{a_k}$



plot of $\hat{w}$ versus c_k:



λ decide the boundary for this piecewise function. The greater the λ, the larger the range where $\hat{w}_k$ is reduced to 0.

3.3

impletmented by `Python`

```
def least_sq_L1(X, y, _lambda, w_0):

    X = X.astype(float)
    n, M = X.shape
```

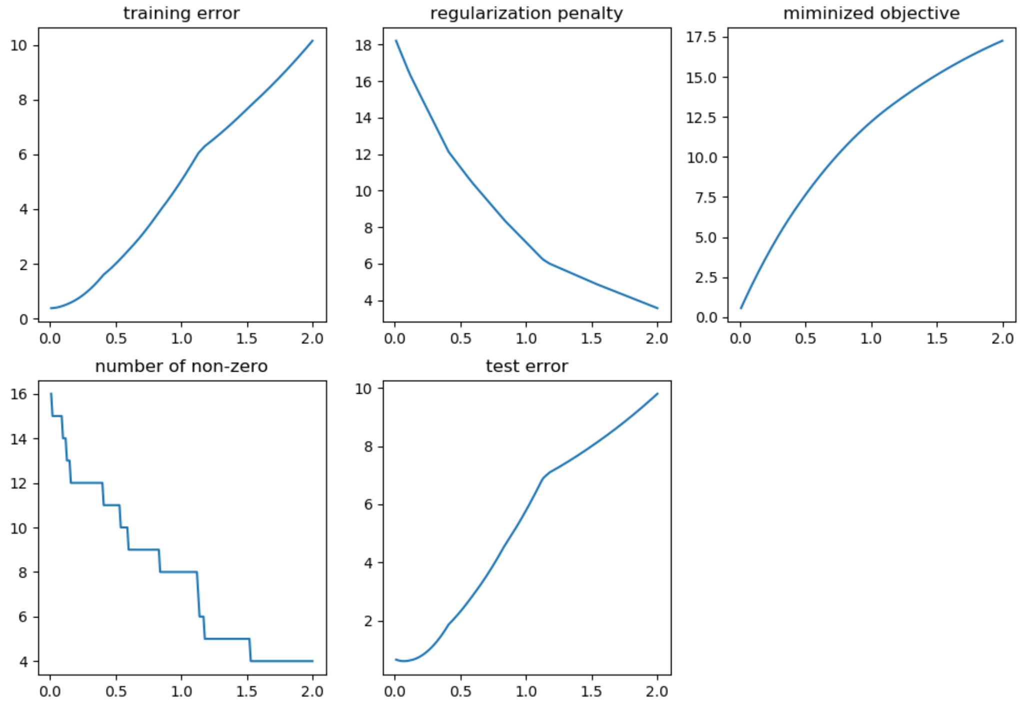
```
    a = np.sum(X**2, axis=0)/n

    w = w_0
    err_tol = 1e-8;
    while True:
        max_err = 0
        w_old = w.copy()
        for k in xrange(M):
            w_minus_k = np.delete(w, k, axis=1)
            phi_minus_k = np.delete(X, k, axis=1)
            c_k = np.sum((y-phi_minus_k.dot(w_minus_k.T))*X[:, k][:,
np.newaxis], axis=0)/float(n)
            if c_k < -_lambda:
                w[0][k] = (c_k + _lambda)/a[k]
            elif -_lambda <= c_k < _lambda:
                w[0][k] = 0
            elif c_k >= _lambda:
                w[0][k] = (c_k - _lambda)/a[k]
        max_err = np.max(np.abs(w - w_old))
        # print(max_err)
        if max_err < err_tol:
            break
    return w
```

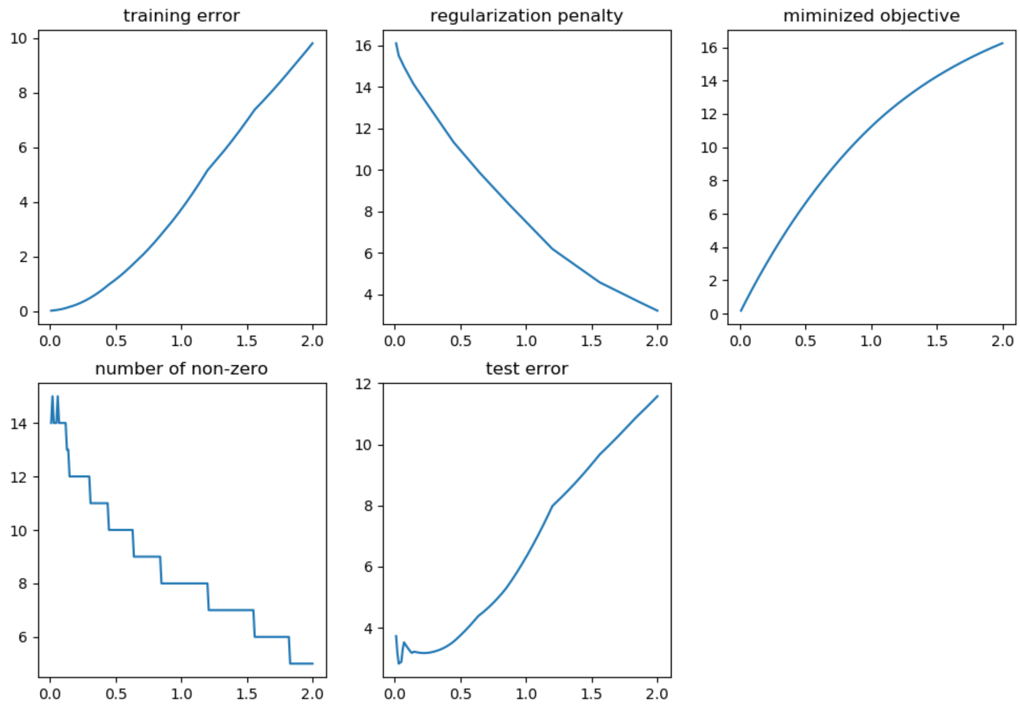
3.4

```
for l in xrange(_lambda.shape[0]):
        w = W[l, :][np.newaxis, :]
        trainError[l] = np.sum(0.5*(y-X.dot(w.T))**2, axis=0)/n #-------------
----(a)
        regPenalty[l] = np.sum(np.abs(w)) #-----------------------------------
----(b)
        objective[l] = trainError[l]+_lambda[l]*regPenalty[l] #---------------
----(c)
        num_non_zero[l] = np.sum(w!=0)                        #---------------
----(d)
        test_error[l] = np.sum(0.5*(y_test-X_test.dot(w.T))**2, axis=0)/n_test
#--(e)
```

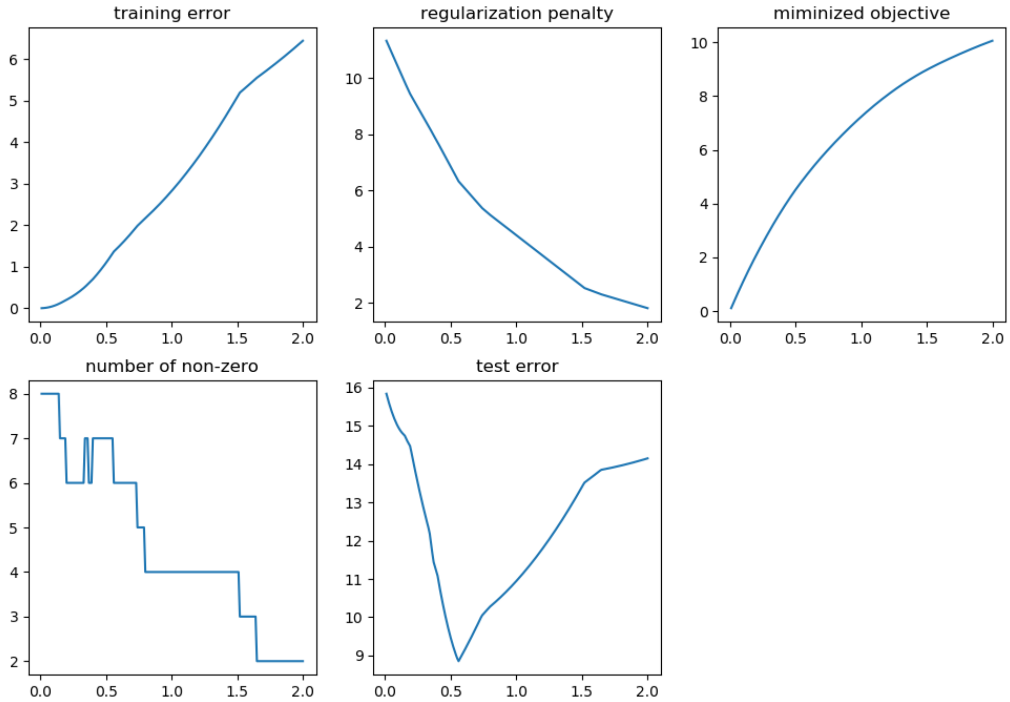
train_large



train_mid



train_small



the smaller the trainning set, the larger the λ is required to minimize the test error.