1. (a) training set $\{(x_i, y_i)\}_{i=1}^{n}$

$$l(w) = \sum_{i=1}^{n} l_i(w) = \sum_{i=1}^{n} -y_i \log h(x_i) - (1-y_i)\log(1-h(x_i))$$

$$h(x) = g(w^T x) = \frac{1}{1+\exp(-w^T x)}$$

$$P(y=1 \mid x; w) = h(x)$$

$$\nabla l(w) = \sum_{i=1}^{n} -y_i \cdot \nabla \log h(x_i) - (1-y_i)\nabla \log(1-h(x_i))$$

$$= \sum_{i=1}^{n} -y_i \frac{\nabla h(x_i)}{h(x_i)} - (1-y_i)\frac{1-\nabla h(x_i)}{1-h(x_i)}$$

$$\nabla h(x_i) = \nabla\left(1+\exp(-w^T x_i)\right)^{-1} = -1 \cdot (-x_i) \cdot \exp(-w^T x_i) \cdot \left(1+\exp(-w^T x_i)\right)^{-2}.$$

$$= \frac{x_i \cdot \exp(-w^T x_i)}{\left(1+\exp(-w^T x_i)\right)^2} = x_i \cdot h(x)^2 \cdot \exp(-w^T x_i)$$

$$\nabla l(w) = \sum_{i=1}^{n} -y_i \cdot \frac{x_i \cdot h(x_i)^2 \cdot \exp(-w^T x_i)}{h(x_i)} - (1-y_i) \cdot \frac{\cancel{\nabla} 1 - x_i\, h(x_i)^2 \cdot \exp(-w^T x_i))}{1-h(x_i)}$$

$$= \sum_{i=1}^{n} -y_i\, x_i\, \cancel{h(x_i)} \frac{\exp(-w^T x_i)}{1+\exp(-w^T x_i)} - (1-y_i)\frac{x_i}{1+\exp(-w^T x_i)}.$$

$$= \sum_{i=1}^{n} -y_i x_i (1-h(x_i)) - (1-y_i) x_i \cdot h(x_i)$$

$$= \sum_{i=1}^{n} 2 y_i x_i h(x_i) - y_i x_i - x_i h.$$

(b) $\nabla^2(L(w)) = \nabla X^T h(x)$

$= XSX^T$, where S is diagonal.

$h(x_i) > 0$ and $\exp(-w^T x_i) > 0$.

it is positive semidefinite and $L(w)$ is convex.

The global one.

2. (a) $f(x;\alpha,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\alpha)^2}{2\sigma^2}\right)$

$L(\theta) = \prod_{i=1} f(x_i;\theta)$

$\log L(\alpha;x) = n \cdot \log(2\pi\sigma^2)^{\frac{1}{2}} - \frac{1}{2\sigma^2} \cdot \sum_{i=1}^{n} (x_i-\alpha)^2$

$\frac{\partial \log L(\alpha;x)}{\partial x} = -\frac{1}{\sigma^2} \sum_{i=1}^{n} (x_i-\alpha) = 0$

$$\boxed{\hat{\alpha} = \frac{1}{n} \sum_{i=1}^{n} x_i}$$

$\frac{\partial \log L(\alpha;x)}{\partial(\sigma^2)} = -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4} \sum (x_i-\alpha)^2.$

$$\begin{cases} \frac{1}{\sigma^2} \sum (x_i-\hat{\alpha}) = 0. \\ -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4} \cdot \sum (x_i-\alpha)^2 = 0 \end{cases}$$

$$\begin{cases} \hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^{n} (x-\hat{\alpha})^2. \\ \hat{\alpha} = \frac{1}{n} \sum_{i=1}^{n} x_i \end{cases}$$

b) $f(x;\mu,\Sigma) = \frac{1}{\sqrt{(2\pi)^d|\Sigma|}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$

$$L(\mu; x_1, x_2, \ldots, x_n) = \frac{1}{(2\pi)^{np/2}|\Sigma|^{n/2}} \cdot \exp\left(-\frac{1}{2}\sum_{k=1}^{n}(x_k-\mu)^T\Sigma^{-1}(x_k-\mu)\right)$$

$x_k,\ k \in \{1, 2, \ldots, n\}$

Let S denotes $S = \sum_{k=1}^{n}(x_k-\bar{x})(x_k-\bar{x})^T$.

$$L(\mu; x_1, \ldots, x_n) = \frac{1}{(2\pi)^{np/2} \cdot |\Sigma|^{n/2}} \cdot \exp\left(-\frac{1}{2}tr(\Sigma^{-1}S) - \frac{1}{2}n(\bar{x}-\mu)^T\Sigma(\bar{x}-\mu)\right)$$

$$\log l(\mu; x_1, \ldots, x_n) = \log\frac{1}{(2\pi)^{np/2}|\Sigma|^{\frac{n}{2}}} - \frac{n}{2}\log|\Sigma| - \frac{1}{2}tr(\Sigma^{-1}S) - \frac{1}{2}n(\bar{x}-\mu)^T\Sigma^{-1}(\bar{x}-\mu)$$

$$\frac{\partial \log l(\mu; x_1, \ldots, x_n)}{\partial \mu} = 0$$

$\therefore \bar{x} - \mu = 0$

$\therefore \hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i$

(a)

$$H(X) - H(X|Y) = -\int p(x) \ln(p(x)) dx + \int p(x,y) \ln(x|y) dy\, dx$$

$$= -\int \left(\int p(x,y) \ln(x|y) dy \right) - p(x) \ln P(x) dx.$$

$$= -\int \left(\int p(x,y) \cdot (\ln P(y|x) + \ln P(x) - \ln P(y)) dy \right) - P(x) \ln P(x) dx$$

$$= \int \left(\int p(x,y) \ln P(y|x) dy + \int p(x,y) dy \cdot \ln P(x) - \int p(x,y) \ln P(y) dy - P(x) \ln P(x) \right) dx$$

$$= -H(y|x) + \iint p(x,y) \ln \frac{P(x)}{P(y)} - P(x) \ln P(x) dy\, dx.$$

$$= -H(y|x) + \iint P(x,y) \ln P(y) dy\, dx$$

$$= -H(y|x) + \int P(y) \ln P(y) dy$$

$$= -H(y|x) + H(y)$$

$$= H(y) - H(y|x)$$

$$= I(x,y)$$

(b) $I(X,Y) = H(X) - H(X|Y)$

$$= -\int p(x) \ln p(x)\, dx + \int p(X,Y) \ln(X|Y)\, dx\, dY.$$

$$= -\int P(f(Y)) \ln P(f(Y))\, df(Y) + \int P(f(Y),Y) \underbrace{\ln P(f(Y)|Y)}_{=1} df(Y)\, d$$

$$= -\int P(Y) \ln P(Y)\, dY$$

$$= H(Y)$$

$$I(X,Y) = -\int P(f(Y)) \ln P(f(Y))\, d f(Y)$$

$$= H(f(Y))$$

$$= H(X)$$

$I(X,Y) = H(X) = H(Y)$

(C) $\hat{p}(x) \triangleq \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}[x = x_i]$ — ①

$$\min_{\theta} D_{KL}(\hat{p} \| q) \triangleq \min_{\theta} -\int \hat{p}(x) \ln \frac{q(x|\theta)}{\hat{p}(x)} dx$$

$$= \min_{\theta} -\int \hat{p}(x) \ln q(x|\theta)\, dx + \int \hat{p}(x) \ln \hat{p}(x)\, dx.$$

$$\propto \min_{\theta} -\int \hat{p}(x) \ln q(x|\theta)\, dx.$$

plug in ① $\Rightarrow$ $= \min_{\theta} -\int \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}[x = x_i] \ln q(x|\theta)\, dx$

$$= \min_{\theta} -\frac{1}{N}\sum_{i=1}^{N} \int \delta(x - x_i) \ln q(x|\theta)\, dx$$

$$= \min_{\theta} -\frac{1}{N}\sum_{i=1}^{N} \ln q(x|\theta) \propto \max_{\theta} q(x|\theta)$$

$\max_{\theta} q(x|\theta)$ is the maximum likelihood estimation given D

(d) objective: $\max \int_{-\infty}^{\infty} p(x) \ln p(x)\, dx$

constraints:
$$\begin{cases} \int_{-\infty}^{\infty} p(x)\, dx = 1 \\ \int_{-\infty}^{\infty} x\, p(x)\, dx = \mu \\ \int_{-\infty}^{\infty} (x-\mu)^2 p(x)\, dx = \delta^2 \end{cases}$$

$$-\int_{-\infty}^{\infty} p(x) \ln p(x)\, dx + \lambda_1 \left(\int_{-\infty}^{\infty} p(x)\, dx - 1 \right) + \lambda_2 \left(\int_{-\infty}^{\infty} x\, p(x)\, dx - \mu \right)$$
$$+ \lambda_3 \left(\int_{-\infty}^{\infty} (x-\mu)^2 p(x)\, dx - \delta^2 \right) = F(p(x))$$

$$\frac{\partial F(p(x))}{\partial x} = 0$$

$$\Rightarrow \hat{p}(x) = \frac{1}{\sqrt{2\pi\delta}} \exp\left(-\frac{(x-\mu)^2}{2\delta^2} \right)$$

4. (a) $w = \arg\min \sum c_i (y_i - \beta^T x_i - b)^2$

$$= \arg\min \| C^{\frac{1}{2}} (y - Xw) \|_2^2$$

where C is diagonal.

$$\frac{\partial L(w)}{\partial w} = -2 (C^{\frac{1}{2}} X)^T C^{\frac{1}{2}} (y - Xw) = 0$$

$$\hat{w} = ((CX)^T X)^{-1} (CX)^T y = (X^T C X)^{-1} X^T C y.$$

$c_i = 1$, $\hat{w} = (X^T X)^{-1} X^T y$, $\hat{w} = \begin{bmatrix} b \\ \beta \end{bmatrix}$

$$y_i = \beta^T x_i + b + \varepsilon_i$$

$y = Xw + \varepsilon$. where $\varepsilon | X \sim N(0, \sigma^2 I)$

$$\max P(y|w, X) = \max - (y - Xw)^T (\sigma^2 I)^{-1} (y - Xw)$$

$$= \min_w \| y - Xw \|_2^2$$

$$\hat{w}_{ML} = (X^T X)^{-1} X^T y. = \hat{w}_{LS}$$

(b) $y|X, w \sim N(Xw, \Sigma)$

$$\max P(y|X,w) = \min (y-Xw)^T \Sigma^{-1} (y-Xw)$$
$$= \min \|\Sigma^{-\frac{1}{2}}(y-Xw)\|_2^2$$

$$\hat{w}_{LS} = (X^T \Sigma X)^{-1} X^T \Sigma^{-1} y.$$

The MLE of w with different noise variance for each i is equivalent to weight LS. with matric $C = \Sigma^{-1}$.

5) 1a) $\min_{w,b} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{N} \varepsilon_i$

subject to $t^{(i)}(w^T x^{(i)}+b) \geq 1-\varepsilon_i$, where $\varepsilon_i \geq 0$

$$S_i \geq \max\left[0, 1 - t^{(i)}(\underline{w}^T x^{(i)}+b)\right]$$

so $\min\left(\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} S_i\right)$ is equivalent to the

~~$S_i \geq \max$~~ $\min \frac{1}{2}\|w\|_2^2 + C\sum \max\left(0, 1-t^{(i)}(\underline{w}^T x^{(i)}+b)\right)$

1b) $\rho(w,b) = \min d(x_i, H) = \min \frac{|w^T x_i + b|}{\|w\|_2}$.

scale w and b by $\frac{1}{\min|w^T x_i + b|}$,

$\min \rho(\hat{w},\hat{b}) = \frac{1}{\|\hat{w}\|_2}$

$y_i(|w^T x_i + b|) = 1 \Rightarrow \rho_{min} = \frac{n}{\|w\|_2}$ where $n = \min|w^T x_i + b|$.

$\rho_i = \frac{|w^T x_i + b|}{\|w\|_2}$ $\forall i$ $t^{(i)}(w^T x^{(i)}+b) \geq 1-\varepsilon_i$

$\therefore \frac{\rho_i}{\rho_{min}} = \varepsilon_i^*$ and $\rho_i \propto \varepsilon_i^*$

(C) ②

$$\min\left(\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\max(0, 1-t^{(i)}(w^Tx^{(i)}+b))\right) \quad \text{--- ①}$$

when $C \to \infty$, ① is equivalant to $\min(\max(0, 1-t^{(i)}(w^Tx^{(i)}+b)))$

$t^{(i)}(w^Tx^{(i)}+b)=1 \Leftrightarrow x^{(i)}$ is the closest point to the margin.

The SVM hard margin

$\min \frac{1}{2}\|w\|_2^2$ s.t. the closest point to the margin

these two are equivalant.

5.
(d)
In[85]:
score_dict
Out[85]:

```
{0.10000000000000001: 0.63428571428571423,
 0.20000000000000001: 0.7142857142857143,
 0.29999999999999999: 0.72571428571428576,
 0.40000000000000002: 0.7371428571428571,
 0.5: 0.73142857142857143,
 0.59999999999999998: 0.73142857142857143,
 0.69999999999999996: 0.7371428571428571,
 0.79999999999999993: 0.72571428571428576,
 0.89999999999999991: 0.74285714285714288,
 0.99999999999999989: 0.7371428571428571,
 1.0999999999999999: 0.7371428571428571,
 1.2: 0.7371428571428571,
 1.3: 0.74285714285714288,
 1.3999999999999999: 0.74857142857142855,
 1.5: 0.74857142857142855,
 1.5999999999999999: 0.74857142857142855,
 1.7: 0.75428571428571434,
 1.8: 0.75428571428571434,
 1.8999999999999999: 0.75428571428571434,
 2.0: 0.75428571428571434}
```

The best C is 1.7, 1.8, 1.9, 2.0.
The best accuracy is 0.77238805970149249.

The classification accuracy of hard margin SVM is 0.735074626866.

I believe that the best accuracy of soft margin is bigger than hard margin because soft margin can ignore some outliers, but the hard margin train all the noise.

And the code I implement shows below:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Tue Nov  7 17:29:49 2017

@author: liuchangbai
```

```
"""


import pandas as pd
import numpy as np
import os,random

os.chdir("/Users/liuchangbai/Desktop/courses/Machine-Learning/Homework/HW3_export")

data = pd.read_csv("diabetes_scale.csv", sep = ",", names = ['label', 'feature1',
'feature2','feature3',
                                         'feature4','feature5','feature6','feature7','feature8'])

test = data[500:768]
data = data[0:500]

# cross validation
y = data['label']
x = data[['feature1', 'feature2','feature3','feature4','feature5','feature6','feature7','feature8']]

y_final = test['label']
x_final = test[['feature1',
'feature2','feature3','feature4','feature5','feature6','feature7','feature8']]

from sklearn.cross_validation import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y,test_size=0.35, random_state=42)


# C value
c_list = np.linspace(0.1, 2, 20)
score_dict = {}

for c_value in c_list:
  # Support Vector Machine
  from sklearn import svm
  c_value = 2.0
  clf = svm.SVC(C = c_value)

  # fit
  clf.fit(x_train, y_train)
  y_pred = clf.predict(x_test)

  # get prediction score
  from sklearn import metrics
```

```
    score = metrics.accuracy_score(y_test,y_pred)
    print(score)

    score_dict[c_value] = score


c_value = 1.7
clf = svm.SVC(C = c_value)

y_predict = clf.predict(x_final)
soft_score = metrics.accuracy_score(y_final, y_predict)



# Hard Margin
hdm = svm.SVC(C = 1* np.exp(6))
hdm.fit(x_train, y_train)

y_pred = hdm.predict(x_final)

# get prediction score

print(metrics.accuracy_score(y_final,y_pred))
```