Machine Learning (CS 6140) Homework 1

Xiao Wang

October 12, 2017

Problem 1

1. **False**,
   Counterexample: A and B are both event: roll a fair dice and get a 6. They are independent.
   Thus $P(B|A) = P(B) = \frac{1}{6}, P(A|B^c) = P(A) = \frac{1}{6}, P(B|A) + P(A|B^c) = \frac{1}{3} \neq 1$

2. **False**,
   Use the same counterexample in (1), $P(A|B) + P(A|B^c) = P(A) + P(A) = \frac{1}{3} \neq 1$

3. **True**,
   $\because B^c$ and $A \cap B$ are exclusive
   $\therefore P(B^c \cup (A \cap B)) = P(B^c) + P(A \cap B)$
   $\therefore P(B^c \cup (A \cap B)) + P(A^c \cap B)$
   $= P(B^c) + P(A \cap B) + P(A^c \cap B)$
   $= P(B^c) + [P(A \cap B) + P(A^c \cap B)]$
   $= P(B^c) + P(B) = 1$

4. **False**,
   Mutually independence doesn't guarantee mutual exclusivity,
   Counterexample: A_1, A_2, A_3 are all event: toss a fair coin and get the head. They are independent.
   Left side $= 1 - (1/2)^3 = 7/8$,
   Right side $= 1/2 + 1/2 + 1/2 = 3/2$,
   Left side $\neq$ Right side.

5. **True**,
   $\because \{(A_i, B_i)\}_{i=1}^n$ are mutually independent.
   $\therefore P(A_1, ..., A_n | B_1, ..., B_n)$
   $= \frac{P(A_1, B_1, ..., A_n, B_n)}{P(B_1, ..., B_n)}$
   $= \frac{\prod_{i=1}^n P(A_i, B_i)}{\prod_{i=1}^n P(B_i)}$
   $= \prod_{i=1}^n P(A_i|B_i)$

Problem 2

1. Multi-variate Gaussian distribution (k-dimensional), $X \sim \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$

$$
\begin{aligned}
f_{\mathbf{X}}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) &= \frac{\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)}{\sqrt{(2\pi)^k |\boldsymbol{\Sigma}|}} \\
&= \frac{\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)}{\sqrt{|2\pi\boldsymbol{\Sigma}|}}
\end{aligned}
$$

2. Laplace distribution with mean μ and variance $2\sigma^2$

$$
f_X(x; \mu, \sigma) = \frac{1}{2\sigma} \exp\left(-\frac{|x-\mu|}{\sigma}\right)
$$

3. Bernoulli distribution, $X \sim \text{Bernoulli}(p), 0 < p < 1$

$$
f(k; p) = \begin{cases} p & \text{if } k = 1, \\ 1-p & \text{if } k = 0. \end{cases}
$$

4. Multinomial distribution with N trials and L outcomes with probabilities $\theta_1, ..., \theta_L$

$$
f(x_1, ..., x_L; N, \theta_1, ..., \theta_L) = \begin{cases} \frac{n!}{\prod_{i=1}^L x_i!} \prod_{i=1}^L \theta_i^{x_i} & \text{, when } \sum_{i=1}^L x_i = n \\ 0 & \text{, otherwise} \end{cases}
$$

5. Dirichlet distribution of order L with parameters $\alpha_1, ..., \alpha_L$

$$
f(x_1, ..., x_L; \alpha_1, ..., \alpha_L) = \frac{1}{\mathrm{B}(\boldsymbol{\alpha})} \prod_{i=1}^{L} x_i^{\alpha_i - 1}
$$

6. Uniform distribution, $X \sim \text{Unif}(a, b), a < b$

$$
f(x) = \begin{cases} \frac{1}{b-a} & \text{, for } a \le x \le b \\ 0 & \text{, otherwise} \end{cases}
$$

7. Exponential distribution, $X \sim \text{Exp}(\lambda), \lambda > 0$

$$
f(x; \lambda) = \begin{cases} \lambda e^{\lambda x} & \text{, when } x \ge 0 \\ 0 & \text{, otherwise} \end{cases}
$$

8. Poisson distribution, $X \sim \text{Poisson}(\lambda), \lambda > 0$

$$
f(k; \lambda) = \Pr(X = k) = \frac{\lambda^k e^{-\lambda}}{k!}, k = 0, 1, 2, ...
$$

Problem 3

1. **True**,

$$\boldsymbol{x^\top A x = x^\top B^\top B x = (Bx)^\top Bx = \|Bx\|_2^2 \geq 0}$$

2. **True**,

$$|\boldsymbol{A} - \lambda \boldsymbol{I}| = -\lambda^3 + 16\lambda^2 - 45\lambda = 0$$
$$\lambda_1 = 0, \lambda_2 = 8 - \sqrt{19}, \lambda_3 = 8 + \sqrt{19}$$
$$\text{All eigenvalues are non-negative}$$

3. **False**,
Counterexample: $\boldsymbol{B} = [-1], \boldsymbol{A} = [-1] + [-1] + [1] = [-1]$
Eigenvalue of $\boldsymbol{A}$ is $-1 < 0$

Problem 4

a)

$$\because \nabla^2 J_1(\boldsymbol{\theta}) = 2\boldsymbol{X}^\top \boldsymbol{X}, \text{ is positive-semidefinite}$$
$$\therefore \text{Basic regression is convex}$$
$$\because \nabla^2 J_2(\boldsymbol{\theta}) = 2(\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I_{d\times d}}), \text{ is positive-semidefinite}$$
$$\therefore \text{Ridge regression is convex}$$
$$\because \nabla^2 J_3(\boldsymbol{\theta}) = 2\boldsymbol{X}^\top \boldsymbol{X}, \text{ is positive-semidefinite}$$
$$\therefore \text{Lasso regression is convex}$$

b) For Basic regression, according to the closed form solution $\boldsymbol{\theta_{min} = (X^T X)^{-1} X^T Y}$, we need to make sure $\boldsymbol{X^T X}$ is invertible, which requires $\boldsymbol{X}$ to have independent columns.

For Ridge regression, according to the closed form solution $\boldsymbol{\theta_{min} = (X^T X + \lambda I)^{-1} X^T Y}$, we need to make sure $\boldsymbol{\theta_{min} = (X^T X + \lambda I)}$ is invertible.

For Lasso regression, according to *Tibshirani, Ryan J. "The lasso problem and uniqueness." Electronic Journal of Statistics 7 (2013): 1456-1490,* the conditions for unique solution is:
For any $\boldsymbol{X, Y}$ and $\lambda > 0$, null$(\boldsymbol{X_\varepsilon}) = \{0\}$ or equivalently rank$(X_\varepsilon) = |\varepsilon|$,
where $\varepsilon = \{i \in \{1, ..., d\} : |X_i^\top (Y - X\theta)| = \lambda\}$

Problem 5

$$\boldsymbol{J_i(\theta)} = l_\delta(y_i - \boldsymbol{\theta}^\top \boldsymbol{x_i})$$

$$\boldsymbol{J_i'(\theta)} = \begin{cases} \delta \boldsymbol{x_i} & , y_i - \boldsymbol{\theta}^\top \boldsymbol{x_i} < -\delta \\ -(y_i - \boldsymbol{\theta}^\top \boldsymbol{x_i})\boldsymbol{x_i} & , |y_i - \boldsymbol{\theta}^\top \boldsymbol{x_i}| \leq \delta \\ -\delta \boldsymbol{x_i} & , y_i - \boldsymbol{\theta}^\top \boldsymbol{x_i} > \delta \end{cases}$$

a) $\boldsymbol{\theta_0}$ = a random R^d
   while ($|\theta_{k+1} - \theta_k| \geq \varepsilon$):
      $\boldsymbol{\theta_{k+1}} = \boldsymbol{\theta_k} - \rho \sum_{i=1}^N \boldsymbol{J_i'(\theta)}|_{\boldsymbol{\theta_k}}$
   return $\boldsymbol{\theta}$

b) Randomly shuffle the training set
   $\boldsymbol{\theta_0}$ = a random R^d
   while ($|\theta_{k+1} - \theta_k| \geq \varepsilon$):
      for i in $1..N$:
         $\boldsymbol{\theta_{k+1}} = \boldsymbol{\theta_k} - \rho \boldsymbol{J_i'(\theta)}|_{\boldsymbol{\theta_k}}$
   return $\boldsymbol{\theta}$

Problem 6

$$P\left(\left|\hat{\theta} - \theta^o\right| \geq 0.1\right) \leq 2e^{-0.1^2 N} \leq 1 - 0.95 = 0.05$$
$$\ln 2 - 0.01N \leq \ln 0.05$$
$$N \geq 100 \ln 40 \approx 369$$

Problem 7

$$\begin{aligned} P(\boldsymbol{\theta}|\boldsymbol{x}, y) &\propto P(\boldsymbol{\theta}, \boldsymbol{x}, y) \\ &\propto P(y|\boldsymbol{x}, \boldsymbol{\theta})P(\boldsymbol{x}|\boldsymbol{\theta})P(\boldsymbol{\theta}) \\ &\propto P(y|\boldsymbol{x}, \boldsymbol{\theta})P(\boldsymbol{\theta}) \\ \log P(\boldsymbol{\theta}|\boldsymbol{x}, y) &= \log P(y|\boldsymbol{x}, \boldsymbol{\theta}) + \log P(\boldsymbol{\theta}) \end{aligned}$$

1.

$$\begin{aligned} \log P(\boldsymbol{\theta}|\boldsymbol{x}, y) &= \log P(y|\boldsymbol{x}, \boldsymbol{\theta}) + \log P(\boldsymbol{\theta}) \\ &= -\frac{\|\boldsymbol{X\theta - Y}\|_2^2}{2\sigma^2} - \frac{\lambda \|\boldsymbol{\theta}\|_2^2}{2} \end{aligned}$$

Take derivative w.r.t. $\boldsymbol{\theta}$ and let it be $\mathbf{0}$:

$$\begin{aligned} \frac{\partial \log P(\boldsymbol{\theta}|\boldsymbol{x}, y)}{\partial \boldsymbol{\theta}} &= \frac{\boldsymbol{X^\top (X\theta - Y)}}{\sigma^2} - \lambda \boldsymbol{\theta} = \mathbf{0} \\ \hat{\boldsymbol{\theta}}_{\text{MAP}} &= \frac{\boldsymbol{X^\top Y}}{\boldsymbol{X^\top X} - \lambda \sigma^2} \end{aligned}$$

2.

$$\begin{aligned}\log P(\boldsymbol{\theta}|\boldsymbol{x}, y) &= \log P(y|\boldsymbol{x}, \boldsymbol{\theta}) + \log P(\boldsymbol{\theta}) \\ &= -\frac{\|\boldsymbol{X\theta} - \boldsymbol{Y}\|_2^2}{2\sigma^2} - \lambda\|\boldsymbol{\theta}\|_1 \\ \hat{\boldsymbol{\theta}}_{\text{MAP}} &= \arg\max_{\boldsymbol{\theta}} \left[-\frac{\|\boldsymbol{X\theta} - \boldsymbol{Y}\|_2^2}{2\sigma^2} - \lambda\|\boldsymbol{\theta}\|_1\right]\end{aligned}$$

Problem 8

1.

$$\begin{aligned}P(\theta|D) &= P(D|\theta)P(\theta) \\ &= \begin{cases} 0.5\left[\theta^{N_1}\left(1-\theta\right)^{N-N_1}\right] & \text{, if } \theta = 0.5 \text{ or } 0.6 \\ 0 & \text{, otherwise} \end{cases}\end{aligned}$$

In order to maximize $P(\theta|D)$, we need to compare 0.5^N and $0.6^{N_1} \times 0.4^{N-N_1}$,
First let's try to solve $0.5^N > 0.6^{N_1} \times 0.4^{N-N_1}$,

$$\begin{aligned}0.5^N &> 0.6^{N_1} \times 0.4^{N-N_1} \\ e^{N\ln 0.5} &> e^{N_1 \ln 0.6} \times e^{(N-N_1)\ln 0.4} \\ e^{(\ln 0.5 - \ln 0.4)N} &> e^{(\ln 0.6 - \ln 0.4)N_1} \\ (\ln 0.5 - \ln 0.4)N &> (\ln 0.6 - \ln 0.4)N_1 \\ \frac{N_1}{N} &< \frac{\ln 0.5 - \ln 0.4}{\ln 0.6 - \ln 0.4} \approx 0.550340\end{aligned}$$

Thus, we can derive the MAP estimate

$$\hat{\theta} = \begin{cases} 0.5 & \text{, if } \frac{N_1}{N} < \frac{\ln 0.5 - \ln 0.4}{\ln 0.6 - \ln 0.4} \\ 0.6 & \text{, if } \frac{N_1}{N} > \frac{\ln 0.5 - \ln 0.4}{\ln 0.6 - \ln 0.4} \end{cases}$$

2. When N is small, the prior used above leads to a better estimate because the biggest possible error is 0.11 while the Beta prior may lead to bigger error.

   When N is large, Beta prior leads to a better estimate because with large samples, Beta prior can give very accurate $\hat{\theta}$. While the smallest error of the prior above is 0.01.

Problem 9

$$\theta = [\mu, \Sigma, \phi], Z = \sqrt{(2\pi)^k \det(\Sigma)}$$
$$P(x|y) = \frac{1}{Z} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right)$$

$$\begin{aligned}\log L(\theta) &= \log P(x, y|\theta) = \log P(y|\theta) + \log P(x|y, \theta) \\ &= \sum_{i=1}^{N} \log \phi_{y^i} - \log Z - \frac{1}{2}\left(x^i - \mu_{y^i}\right)^\top \Sigma_{y^i}^{-1}\left(x^i - \mu_{y^i}\right)\end{aligned}$$

Take derivative w.r.t μ

$$\begin{aligned}\frac{\partial \log L}{\partial \mu_k} &= -\sum_{i=1}^{N} \mathbb{1}(y^i = k)\Sigma^{-1}(x^i - \mu_k) = 0 \\ \mu_k &= \frac{\sum_{i=1}^{N} \mathbb{1}(y^i = k)x^i}{\sum_{i=1}^{N} \mathbb{1}(y^i = k)}\end{aligned}$$

Take derivative w.r.t Σ^{-1}

$$\begin{aligned}\frac{\partial \log L}{\partial \Sigma_k^{-1}} &= -\sum_{i=1}^{N} \mathbb{1}(y^i = k)\left[-\frac{-\partial \log Z_k}{\partial \Sigma_k^{-1}} - \frac{1}{2}\left(x^i - \mu_k\right)\left(x^i - \mu_k\right)^\top\right] \\ &= -\sum_{i=1}^{N} \mathbb{1}(y^i = k)\left[\frac{1}{2}\Sigma_k - \frac{1}{2}\left(x^i - \mu_k\right)\left(x^i - \mu_k\right)^\top\right] = 0 \\ \Sigma_k &= \frac{\sum_{i=1}^{N} \mathbb{1}(y^i = k)\left(x^i - \mu_k\right)\left(x^i - \mu_k\right)^\top}{\sum_{i=1}^{N} \mathbb{1}(y^i = k)}\end{aligned}$$

Take derivative w.r.t. ϕ (Same as Bernoulli)

$$\phi_k = \frac{\sum_{i=1}^{N} \mathbb{1}(y^i = k)}{N}$$

Problem 10

a) Please see code attached in email.

b)

```
Enter polynomial degree n: 2
Enter the SGD batch_size: 10
Closed-form: (time elapsed: 0.141607s)
theta:
[[ 9.49203678]
 [ 4.79191663]
 [ 1.52906587]]
training error:  24.7419196363
test error:  4413.5867675
```

```
SGD: (time elapsed: 67.850271s)
theta:
[[ 9.30037904]
 [ 4.79381276]
 [ 1.54907739]]
training error:  24.7579786663
test error:  4452.75062185
```

```
Enter polynomial degree n: 3
Enter the SGD batch_size: 10
Closed-form: (time elapsed: 0.086794s)
theta:
[[ 10.00815033]
 [  0.20418927]
 [  1.47413164]
 [  0.47320168]]
training error:  3.96786315703
test error:  53.8311491635
```

```
SGD: (time elapsed: 75.174964s)
theta:
[[ 9.81429812]
 [ 0.22700942]
 [ 1.4946132 ]
 [ 0.47150315]]
training error:  3.98438117027
test error:  58.2873294465
```

```
Enter polynomial degree n: 5
Enter the SGD batch_size: 10
Closed-form: (time elapsed: 0.060946s)
theta:
[[  9.84545670e+00]
 [  2.03644970e-01]
 [  1.57564738e+00]
 [  4.75176601e-01]
 [ -7.40094164e-03]
 [ -1.77139419e-04]]
training error:  3.94688107228
test error:  41.5094372635
```

```
SGD: (time elapsed: 94.778757s)
theta:
[[  9.56054545e+00]
 [  3.91341050e-01]
 [  1.63165387e+00]
 [  4.30716678e-01]
 [ -6.35521522e-03]
 [ -1.60030454e-04]]
training error:  4.53961142786
test error:  102.739849019
```

During my experiments, I find that as the size of the mini-batch becomes bigger, the speed becomes slower and testing error becomes smaller.

c)

```
Enter polynomial degree n: 2
Enter the K-fold k: 2
Enter the SGD batch_size: 5
Closed-form: (time elapsed: 0.053398s)
theta:
[[ 30.58703919]
 [ 13.4976292 ]
 [ -0.32367572]]
training error:  868.208697501
test error:  1746.52302338

10.0 4937.55710403
1.0 2673.71273049
0.1 2170.48483805
0.01 2131.98688717
0.001 2137.38245084
0.0001 2138.10959663
1e-05 2138.18421681
Best lambda: 0.01
SGD: (time elapsed: 65.346378s)
theta:
[[ 29.19883675]
 [ 13.53345861]
 [ -0.29610306]]
training error:  869.255589614
test error:  1766.27254489
```

```
Enter polynomial degree n: 3
Enter the K-fold k: 2
Enter the SGD batch_size: 5
Closed-form: (time elapsed: 0.056309s)
theta:
[[ 9.84129188]
 [-0.8605458 ]
 [ 1.40263187]
 [ 0.51037625]]
training error:  44.2518755788
test error:  210.528764077

10.0 202.47331726
1.0 158.707285216
0.1 116.664398555
0.01 111.133357756
0.001 111.705643569
0.0001 111.79026242
1e-05 111.799029737
Best lambda:  0.01

SGD: (time elapsed: 72.355802s)
theta:
[[ 9.20635992]
 [-0.97891546]
 [ 1.42215151]
 [ 0.52698934]]
training error:  49.9598261489
test error:  189.677295885
```

```
Enter polynomial degree n: 5
Enter the K-fold k: 2
Enter the SGD batch_size: 5

Closed-form: (time elapsed: 0.083742s)
theta:
[[  1.06879498e+01]
 [  1.02301402e-01]
 [  1.13584026e+00]
 [  4.15127537e-01]
 [  9.05648400e-03]
 [  1.99464784e-03]]
training error:  42.3717141141
test error:  208.676641722

regression.py:89: RuntimeWarning:
overflow encountered in multiply
  gradient += (theta.T.dot(X_batches[i][j])
  - y_batches[i][j]) * X_batches[i][j] + 2 * la *
regression.py:89: RuntimeWarning:
invalid value encountered in add
  gradient += (theta.T.dot(X_batches[i][j])
  - y_batches[i][j]) * X_batches[i][j] + 2 * la *
regression.py:91: RuntimeWarning:
invalid value encountered in subtract
  theta -= step_size * gradient
```

Test error is a convex function of both λ and n.
This is why we are possible to run into underfitting and overfitting situations.