Machine Learning Homework #3

Joe Ellis - jge2105

November 7, 2013

1 Problem 1

1.1 Problem1.1 - Discrete

We assume that we have a coin that can land on heads or tails, and we have 3 possible values for μ, which are $\frac{1}{2}, \frac{1}{4}, \frac{3}{4}$. The first problem that we address is to find the minimum number of tosses we'd need to see in order to conclude that $p(\mu = \frac{1}{2}) > \frac{1}{2}$. We will refer to this series of tosses as, D. Using Bayes rule we know that $p(\mu|D) = p(D|\mu)p(\mu)/p(D)$. For this example, we can assume that the prior on each choice of mu is equal. We know that the flipping of a coin is given by the bernoulli distribution, whereN_H and N_T are the number of heads and tails in our set respectively. We will ignore $p(\mu)$ in the equations since the priors are uniform. The quickest time that we can be sure that the $p(\mu = \frac{1}{2}|D) > \frac{1}{2}$ is $D = (H, T, H, T, H, T)$

$$p(\mu = \frac{1}{2}|D) = \frac{(\mu)^{N_H}(1-\mu)^{N_T}}{\sum_{\mu}(\mu)^{N_H}(1-\mu)^{N_T}} \tag{1}$$

$$= .5424 \tag{2}$$

The second test that we will do on this is to find the minimal set D such that $p(\mu = \frac{3}{4}|D) > \frac{1}{2}$. For this the smallest dataset that makes this a reality is $D = (H, H)$, by the same logic.

$$p(\mu = \frac{3}{4}|D) = \frac{(\mu)^{N_H}(1-\mu)^{N_T}}{\sum_{\mu}(\mu)^{N_H}(1-\mu)^{N_T}} \tag{3}$$

$$= .6429 \tag{4}$$

1.2 Problem 1.2 - Continuous

Consider 2 possible distributions: (A) $\mu\ uniform[0,1]$; (B) we have some reason to think the coin is likely to be fair, so we have a parabola that is peaked at $\frac{1}{2}$ and then goes to 0 at 0 and 1.

Use the two datasets, and answer a variety of questions for each, $D_1 = (H,T)$; $D_2 = (T,T,T)$.

We will use the following arguments with the beta distribution.

$$Beta(\mu, a, b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}(1-\mu)^{a-1}\mu^{b-1} \tag{5}$$

$$E(Beta(\mu)) = \frac{a}{a+b} \tag{6}$$

$$Var(Beta(\mu)) = \frac{ab}{(a+b)^2(a+b+1)} \tag{7}$$

$$\Gamma(n) = (n-1)! \tag{8}$$

1.2.1 A – 1

P(H) given the prior The probability of $p(H) = \int_{\mu} p(H|\mu)p(\mu)$, and since μ is uniform then we know that the $p(\mu) = 1$, because the area under the prior must integrate to 1 to be a true probability distribution. Therefore, we have

$$p(H) = \int_{\mu} p(H|\mu)p(\mu) \tag{9}$$

$$p(H) = \int_{\mu} p(H|\mu) \tag{10}$$

$$p(H) = \frac{\mu^2}{2}|_0^1 \tag{11}$$

$$p(H) = \frac{1}{2}. \tag{12}$$

$p(\mu|D)$ This is the posterior distribution given the dataset D. We know that the $Beta(\mu, 1, 1) = uniformdistribution.$

$$p(\mu|D) = \frac{p(D|\mu)p(\mu)}{p(D)} \tag{13}$$
$$= \frac{p(D|\mu)}{p(D)} \tag{14}$$
$$= Beta(\mu, 2, 2) \tag{15}$$
$$= \frac{3!}{1!1!}\mu(1-\mu) \tag{16}$$
$$= 6\mu(1-\mu) \tag{17}$$

μ_{ML} GIVEN D The maximum likelihood estimate of this value is the peak of the function above, which is at $\mu = \frac{1}{2}$. This has been proven in class from the lectures, that the ML solution is $\mu = \frac{N_H}{N_H+N_T}$, where N_H is the number of heads and N_T is the number of tails.

μ_{MAP} GIVEN D We know that $\mu_{MAP} = argmaxp(\mu|D)p(\mu)$, but since our prior is uniform here $\mu_{MAP} = \mu_{ML}$. Therefore, $\mu_{MAP} = \frac{1}{2}$.

$p(H|D)$ Now let's find the full bayesian integral for the value given the found μ, that we have for our dataset. Also, remember for our problem $p(H|\mu) = \mu$), because μ is the probability that for any given toss we get heads.

$$p(H|D) = \int_u p(H|\mu)p(\mu|D)d\mu \tag{18}$$
$$= \int_u \mu p(\mu|D)d\mu \tag{19}$$
$$= E(p(\mu|D)) \tag{20}$$
$$= E(Beta(\mu, 2, 2)) \tag{21}$$
$$= \frac{2}{2+2} \tag{22}$$
$$= \frac{1}{2} \tag{23}$$
$$\tag{24}$$

$Var(p(\mu|D))$

$$Var(p(\mu|D)) = Var(Beta(\mu, 2, 2)) \tag{25}$$
$$= \frac{2*2}{(2+2)^2(2+2+1)} \tag{26}$$
$$= 0.05 \tag{27}$$

1.2.2 A – 2

P(H) GIVEN THE PRIOR Same as seen in A-1, because the prior did not change.

$p(\mu|D)$ This is the posterior distribution given the dataset D.

$$
\begin{aligned}
p(\mu|D) &= \frac{p(D|\mu)p(\mu)}{p(D)} && (28)\\
&= \frac{p(D|\mu)}{p(D)} && (29)\\
&= Beta(\mu, 1, 4) && (30)\\
&= \frac{4!}{0!3!}(1-\mu)^3 && (31)\\
&= 4(1-\mu)^3 && (32)
\end{aligned}
$$

μ_{ML} GIVEN D The maximum likelihood estimate of this value is the peak of the function above, which is at $\mu = 0$. This has been proven in class from the lectures, that the ML solution is $\mu = \frac{N_H}{N_H+N_T}$, where N_H is the number of heads and N_T is the number of tails, and we did not see any heads.

μ_{MAP} GIVEN D We know that $\mu_{MAP} = argmaxp(\mu|D)p(\mu)$, but since our prior is uniform here $\mu_{MAP} = \mu_{ML}$. Therefore, $\mu_{MAP} = 0$.

$p(H|D)$ Now let's find the full bayesian integral for the value given the found μ, that we have for our dataset. Also, remember for our problem $p(H|\mu) = \mu$), because μ is the probability that for any given toss we get heads.

$$
\begin{aligned}
p(H|D) &= \int_u p(H|\mu)p(\mu|D)d\mu && (33)\\
&= \int_u \mu p(\mu|D)d\mu && (34)\\
&= E(p(\mu|D)) && (35)\\
&= E(Beta(\mu, 1, 4)) && (36)\\
&= \frac{1}{1+4} && (37)\\
&= \frac{1}{5} && (38)\\
& && (39)
\end{aligned}
$$

$Var(p(\mu|D))$

$$Var(p(\mu|D)) = Var(Beta(\mu, 1, 4)) \tag{40}$$
$$= \frac{1 * 4}{(5)^2(1 + 4 + 1)} \tag{41}$$
$$= 0.0267 \tag{42}$$

1.2.3 B–1

P(H) GIVEN THE PRIOR The probability of $p(H) = \int_\mu p(H|\mu)p(\mu)$, and $p(\mu)$ is not uniform, which means that we need to include it in the integral. We will use the function for $p(\mu) = Beta(\mu, 2, 2)$. Therefore,

$$p(H) = \int_\mu p(H|\mu)p(\mu) \tag{43}$$
$$P(H) = \frac{3!}{1!1!} \int_\mu \mu * (\mu)(1 - \mu) \tag{44}$$
$$p(H) = \frac{3!}{1!1!} \int_\mu \mu^2 - \mu^3 \tag{45}$$
$$P(H) = 6\frac{(x^3)}{3} - \frac{(x^4)}{4}|_0^1 \tag{46}$$
$$P(H) = 6(\frac{1}{12}) \tag{47}$$
$$P(H) = \frac{1}{2}. \tag{48}$$
$$\tag{49}$$

$p(\mu|D)$ This is the posterior distribution given the dataset D.

$$p(\mu|D) = \frac{p(D|\mu)p(\mu)}{p(D)} \tag{50}$$
$$= \frac{p(D|\mu)}{p(D)} \tag{51}$$
$$= Beta(\mu, 3, 3) \tag{52}$$
$$= \frac{5!}{2!2!}(\mu)^2(1 - \mu)^2 \tag{53}$$
$$= 30\mu^2(1 - \mu)^2 \tag{54}$$

μ_{ML} GIVEN D This portion of the function is the same as A-1, because ML assumes we have uniform priors.

μ_{MAP} GIVEN D For this more complicated function we have not seen in class, let's take the derivative and set equal to zero.

$$\frac{\partial}{\partial \mu} 30\mu^2(1-\mu)^2 = 0 \tag{55}$$
$$\frac{\partial}{\partial \mu} \mu^2(1-2\mu+\mu^2) = 0 \tag{56}$$
$$\frac{\partial}{\partial \mu} \mu^2 - 2\mu^3 + \mu^4 = 0 \tag{57}$$
$$2\mu - 6\mu^2 + 4\mu^3 = 0 \tag{58}$$
$$\frac{1}{2} = \mu \tag{59}$$
$$\tag{60}$$

$p(H|D)$ Now let's find the full bayesian integral for the value given the found μ, that we have for our dataset. Also, remember for our problem $p(H|\mu) = \mu$), because μ is the probability that for any given toss we get heads.

$$p(H|D) = \int_u p(H|\mu)p(\mu|D)d\mu \tag{61}$$
$$= \int_u \mu p(\mu|D)d\mu \tag{62}$$
$$= E(p(\mu|D)) \tag{63}$$
$$= E(Beta(\mu, 3, 3)) \tag{64}$$
$$= \frac{3}{3+3} \tag{65}$$
$$= \frac{1}{2} \tag{66}$$
$$\tag{67}$$

$Var(p(\mu|D))$

$$Var(p(\mu|D)) = Var(Beta(\mu, 3, 3)) \tag{68}$$
$$= \frac{3*3}{(6)^2(3+3+1)} \tag{69}$$
$$= .0357 \tag{70}$$

1.3 B – 2

P(H) GIVEN THE PRIOR See the same as B-1.

$p(\mu|D)$ This is the posterior distribution given the dataset D.

$$
\begin{aligned}
p(\mu|D) &= \frac{p(D|\mu)p(\mu)}{p(D)} && (71)\\
&= \frac{p(D|\mu)}{p(D)} && (72)\\
&= Beta(\mu, 2, 5) && (73)\\
&= \frac{6!}{1!5!}(\mu)^1(1-\mu)^5 && (74)\\
&= 6\mu(1-\mu)^5 && (75)
\end{aligned}
$$

μ_{ML} GIVEN D For this answer please see A-2, because ML assumes that we have a uniform prior.

μ_{MAP} GIVEN D For this more complicated function we have not seen in class, let's take the derivative and set equal to zero.

$$
\begin{aligned}
\frac{\partial}{\partial\mu}6\mu(1-\mu)^5 &= 0 && (76)\\
-5 * \mu(1-\mu)^4 + (1-\mu)^5 &= 0 && (77)\\
-5\mu + (1 - mu) &= 0 && (78)\\
1 &= 6\mu && (79)\\
\mu &= \frac{1}{6} && (80)
\end{aligned}
$$

$p(H|D)$ Now let's find the full bayesian integral for the value given the found μ, that we have for our dataset. Also, remember for our problem $p(H|\mu) = \mu$), because μ is the probability that for any given toss we get heads.

$$
\begin{aligned}
p(H|D) &= \int_u p(H|\mu)p(\mu|D)d\mu && (81)\\
&= \int_u \mu p(\mu|D)d\mu && (82)\\
&= E(p(\mu|D)) && (83)\\
&= E(Beta(\mu, 2, 5)) && (84)\\
&= \frac{2}{2+5} && (85)\\
&= \frac{2}{7} && (86)\\
& && (87)
\end{aligned}
$$

$Var(p(\mu|D))$

$$Var(p(\mu|D)) = Var(Beta(\mu, 2, 5)) \tag{88}$$
$$= \frac{2 * 5}{(7)^2(2 + 5 + 1)} \tag{89}$$
$$= .0255 \tag{90}$$

2 Problem 2

We are assuming that we have a good test for swine flu that is highly accurate, with the probabilities given below. We want to find the likelihood if I (Joe) take the test and it outputs true, what is the probability that I have swine flu.

$$p(test = true|flu = True) = 0.99 \tag{91}$$
$$p(test = false|flu = false) = 0.98 \tag{92}$$
$$p(flu = true) = .0001 \tag{93}$$
$$p(flu = false) = .999 \tag{94}$$

Let's use Bayes Rule to solve this problem.

$$p(flu = true|test = True) = \frac{p(test = true|flu = True)p(flu = true)}{p(test = true)} \tag{95}$$
$$= \frac{0.99 * 0.0001}{0.99 * 0.0001 + (1 - 0.98) * 0.999} \tag{96}$$
$$= 0.0049 \tag{97}$$
$$\tag{98}$$

So still not that likely, shows the power of the prior.

3 Problem 3

3.1 Problem A

We want to show that $u \perp (v, w)|x \Rightarrow u \perp v|x$.

PROOF Was not able to solve this proof.

3.2 Problem B

We want to show that $A \perp (B, C) \Rightarrow A \perp B$.

PROOF

$$p(A, B, C) = p(A)p(B, C) \tag{99}$$

$$\int_C p(A, B, C) = \int_C p(A)p(B, C) \tag{100}$$

$$p(A, B) = p(A)p(B) \tag{101}$$

$$A \perp B \tag{102}$$

4 Problem 4

The weibull distribution is a probablility distribution over non-negative scalar values. The distribution is as follows, $p(x|\lambda) = \frac{3}{\lambda}(\frac{x}{\lambda})^2 exp(-(\frac{x}{\lambda})^3)$. Given a dataset $D = (x_0, x_1, ..., x_N)$, what is the ML estimate of λ. Let's use the log-likelihood.

$$l(D|\lambda) = \sum log(p(x_i|\lambda) \tag{103}$$

$$= \sum log(\frac{3}{\lambda}) + 2log(\frac{x_i}{\lambda}) - (\frac{x_i}{\lambda})^3 \tag{104}$$

$$= \sum log(3) - log(\lambda) + 2log(x_i) - 2log(\lambda) - (\frac{x_i}{\lambda})^3 \tag{105}$$

Now let's take the derivative of the log-likelihood, and set to 0.

$$\frac{\partial l}{\partial \lambda} = \frac{\partial}{\partial \lambda} \sum log(3) - log(\lambda) + 2log(x_i) - 2log(\lambda) - (\frac{x_i}{\lambda})^3 \tag{106}$$

$$\sum -\frac{1}{\lambda} - \frac{2}{\lambda} + 3\lambda^{-4}x_i^3 = 0 \tag{107}$$

$$\frac{-3N}{\lambda} + \sum 3\lambda^{-4}x_i^3 = 0 \tag{108}$$

$$\frac{3N}{\lambda} = \sum 3\lambda^{-4}x_i^3 \tag{109}$$

$$\frac{3N\lambda^4}{3\lambda} = \lambda^4 \sum 3\lambda^{-4}x_i^3 \tag{110}$$

$$\lambda^3 = \sum \frac{x_i^3}{N} \tag{111}$$

$$\lambda = (\sum \frac{x_i^3}{N})^{1/3} \tag{112}$$

5 Problem 5

In problem 5 we create a kernelized logistic regression function using the RBF, kernel. We optimize the loss function that can be seen below using stochastic gradient descent, basic gradient descent using these methods. The optimization function can be seen below,

$$J(w) = \sum_{i=1}^{N} log(\sigma(y_i w^T k_i)) + \lambda w^T w. \tag{113}$$

We then take the gradient, described here.

$$\frac{\partial J(w)}{\partial w} = \sum_{i=1}^{M} -y_i k_i log(\sigma(-y_i w^T k_i)) + \frac{\lambda}{2} w \tag{114}$$

Using this update step, we end up with the update rule here,

$$w^t = w^{t-1} + \eta(\sum_{i=1}^{M} -y_i k_i log(\sigma(-y_i w^T k_i)) + \lambda w). \tag{115}$$

We remove the 2 from the gradient without the loss of generality. Finally, we select $\eta = .001$, and the Regularizer Penalty as $\lambda = .01$. We were able to get a training accuracy of 91.16%, and test accuracy 79.503% of using mini-batch gradient descent with 100 points. Using full gradient descent, which is the same as batch gradient descent, just using the entire batch for the descent we were able to achieve training accuracy of 80.1%, and testing accuracy of 60.52%. This worse results on the full gradient descent, is due to an iteration limit that I set on the descent due to the fact that it was taking to long to converge. Therefore, the stochastic gradient descent method worked much better for our purposes and converged much quicker. The stochastic method achieves convergence much quicker, and we used randomly chosen points for each update step. All of the code that was used to create can be found in the zip file attached to the write up. Also, my memory was large enough to read the entire Kernel Gram matrix into memory, and because it was much quicker to do so I calculated the gram matrix at the beginning. A plot of the decreasing cost function during the stochastic gradient descent algorithm can be seen in Figure 1.

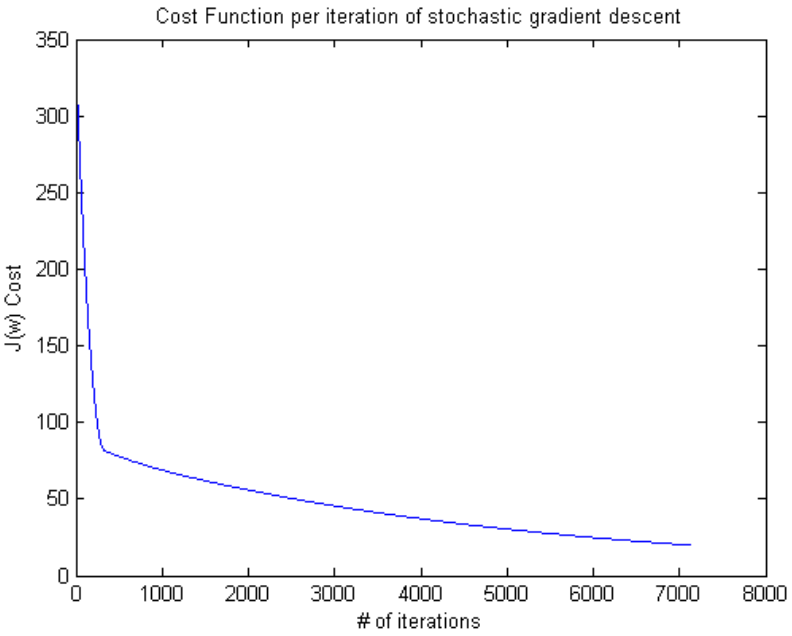

Figure 1: Cost Function during Stochastic Gradient Descent