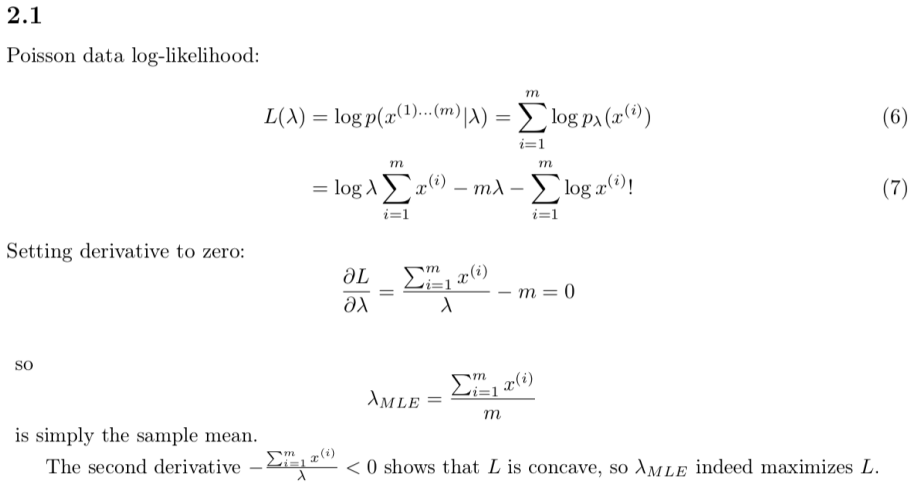
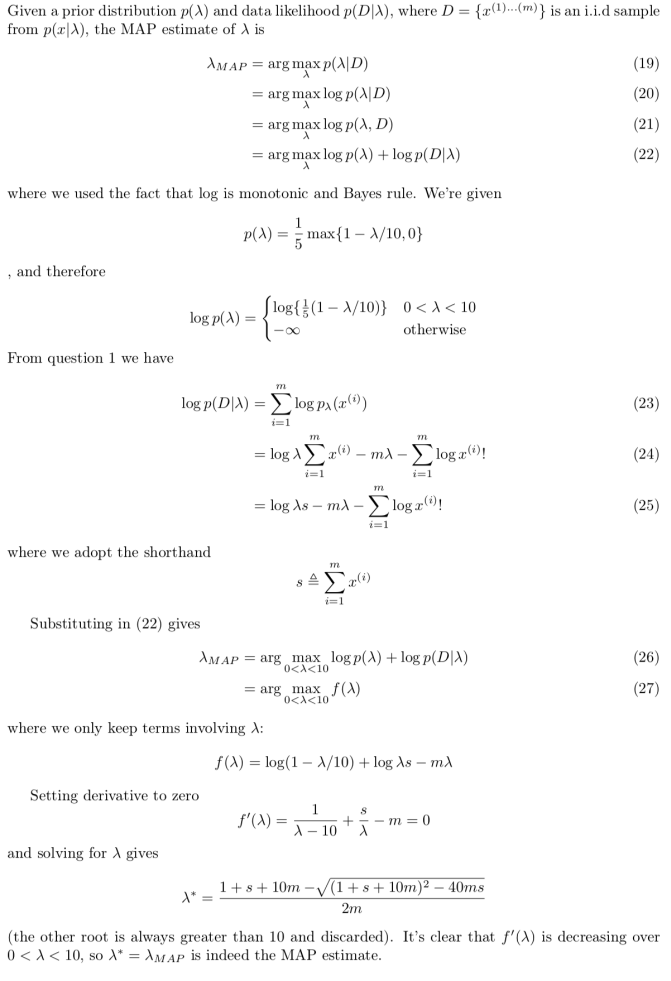
# Problem 1 Poisson MLE

**1.1**



**1.2**



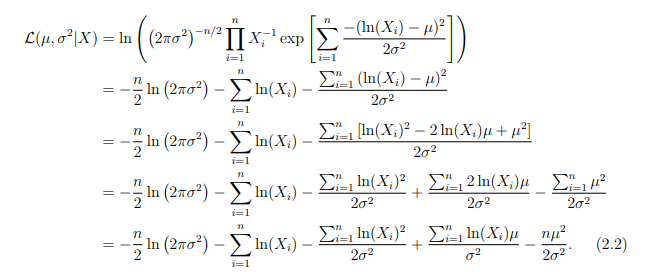
**1.3**

Any reasonable answer worth full credit. Example: we don’t favor this prior cause it greatly changes the form of MLE estimator, a conjugate prior like Gamma distribution can be used.

# Problem 2 Log-Normal MLE

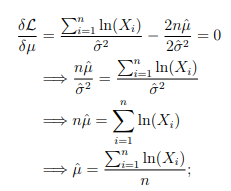
**2.1**

The log-likelihood is simply

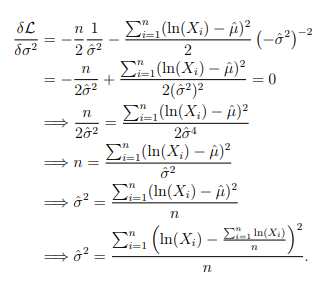


**2.2**

Take derivative respect to , we can get the MLE estimator,



Take derivative respect to , we can get the MLE estimator (note that is fixed to ),



To prove these are the MLE estimators, we also need to show that

* This is a local maxima, this is done by showing that the hessian matrix of likelihood function is negative-semidefinite at the point and
* Show that the boundary points of cannot achieve higher value.

Details are omitted here, for reference, you can check the attachments in this solution.

**2.3**

This is a log-normal distribution where



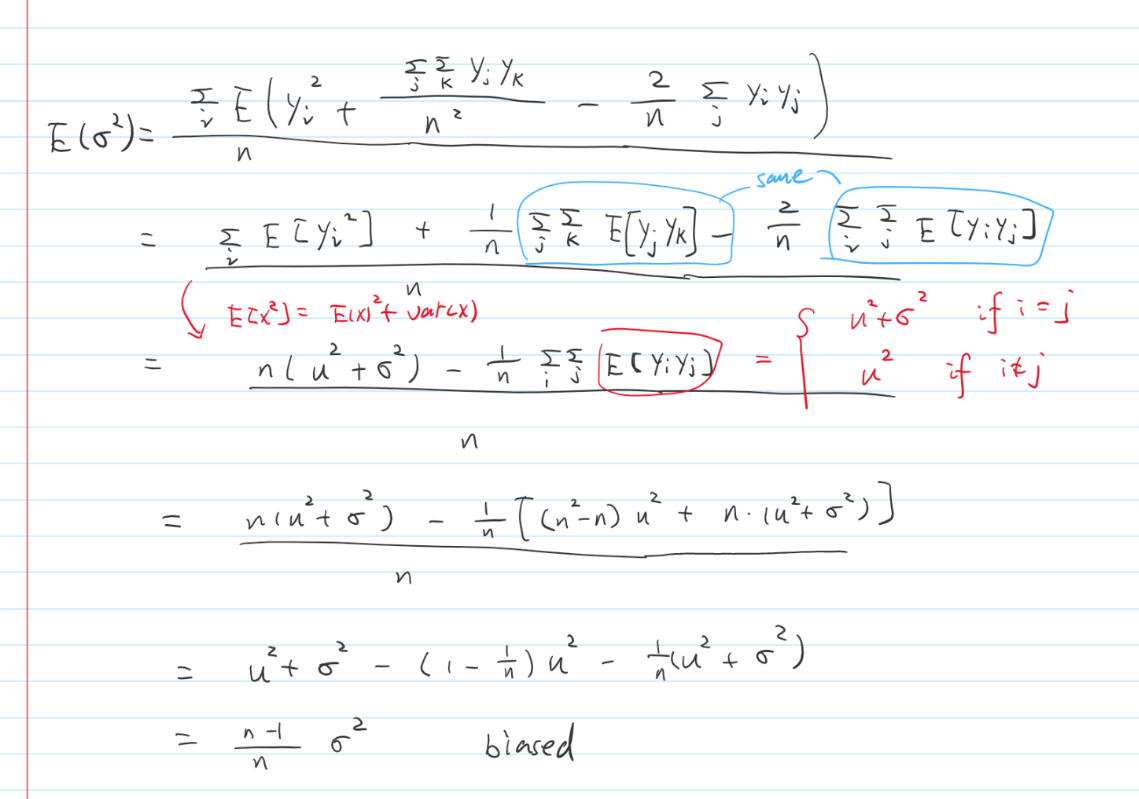
To put it in the another form, we have



This means

Thus, is unbiased.

Similarly, you can prove



Note that the means

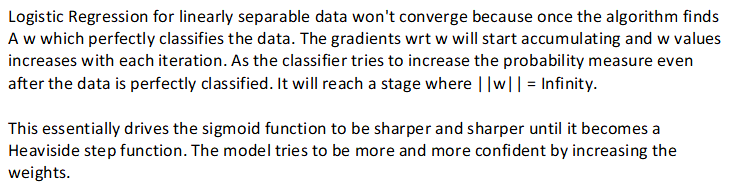
**2.4**

Any reasonable response worth full credit. Example: note that the observations are always greater than zero, therefore the parameter can never be negative. However, Gaussian distribution always assign probability mass for negative values, so it should not be preferred.

# Problem 3 Logistic Regression

**3.1**

The accuracy on test set is 0.769.



(From Krishnateja Killamsetty)

**3.2**

I pick as regularization constant, the result is shown below (my code has more results)

Train accuracy: 1.0

Valid accuracy: 0.7692307692307693

Test accuracy: 0.7884615384615384

Learned W:

1.03,0.27,-1.31,1.06,-0.27,-0.79,-0.46,-1.11,1.38,0.14,0.43,0.72,0.63,0.01,-0.15,-0.81,-0.88,0.31,0.40,0.91,-0.41,-0.04,1.22,0.37,-0.22,-0.58,-0.22,0.38,0.20,0.21,-1.38,0.63,0.56,-0.64,-0.24,-0.73,-0.48,0.60,-0.18,-0.85,0.03,-0.39,0.33,0.50,0.50,-0.13,0.40,0.97,1.19,-1.36,0.77,0.62,0.32,0.27,-0.87,0.50,-0.83,1.07,0.14,0.30

Learned b: 0.5860328876576976

**3.3**

I pick as regularization constant, the result is shown below (my code has more results)

Train accuracy: 0.9903846153846154

Valid accuracy: 0.75

Test accuracy: 0.8076923076923077

Learned W:

1.23,0.00,-1.28,1.11,0.00,-1.10,-0.09,-1.34,1.65,-0.00,0.01,0.75,0.75,-0.00,-0.00,-0.85,-0.66,0.01,0.01,1.21,-0.31,0.00,1.16,0.01,0.00,-0.74,-0.01,0.22,0.01,0.29,-0.99,0.58,0.11,-0.46,-0.01,-0.88,-0.01,0.00,-0.01,-0.69,-0.00,-0.00,0.01,0.78,0.01,-0.00,-0.00,0.98,1.33,-1.38,1.14,0.56,0.18,0.14,-0.78,0.61,-0.87,0.95,0.06,0.17

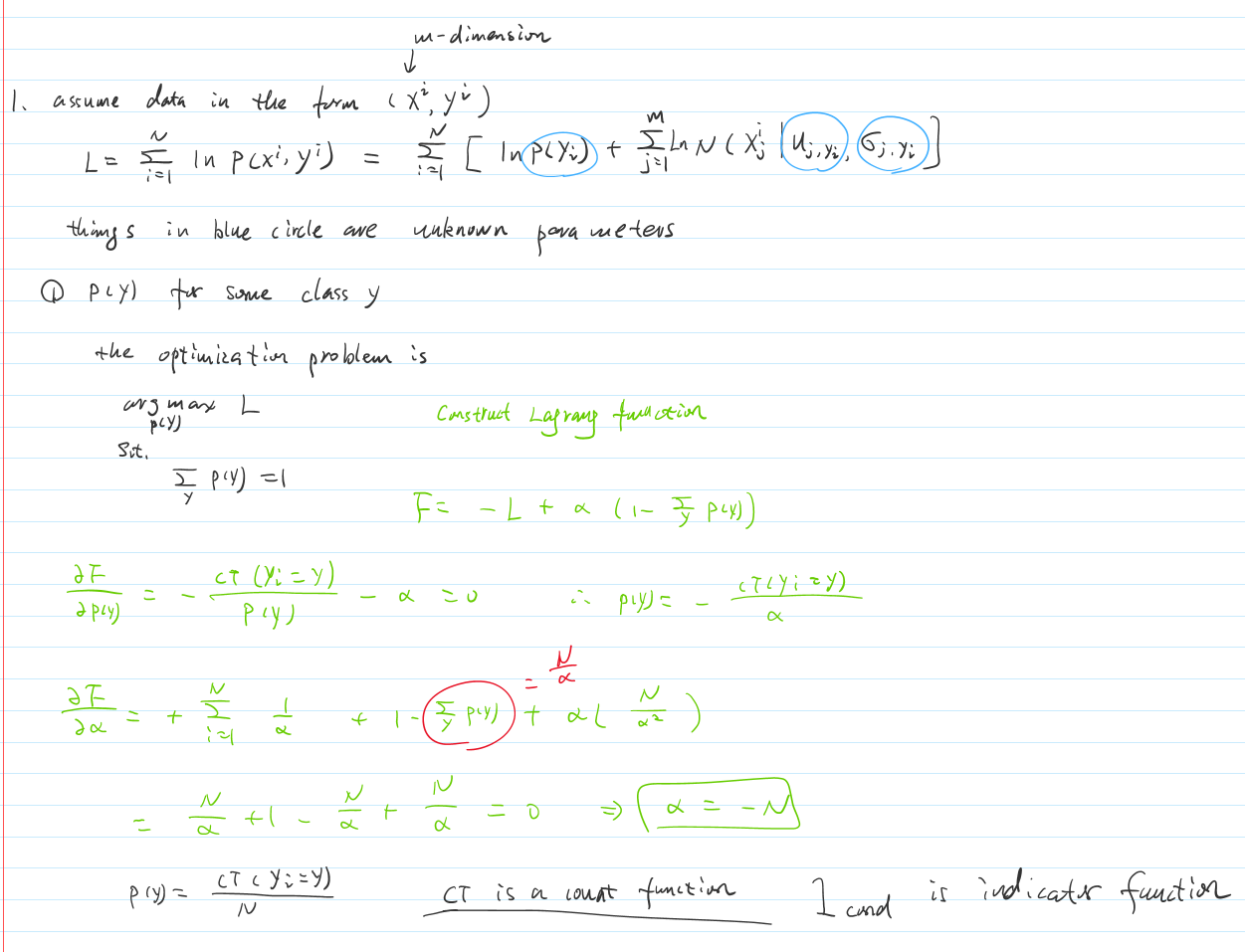
Learned b: 0.46684986876493856

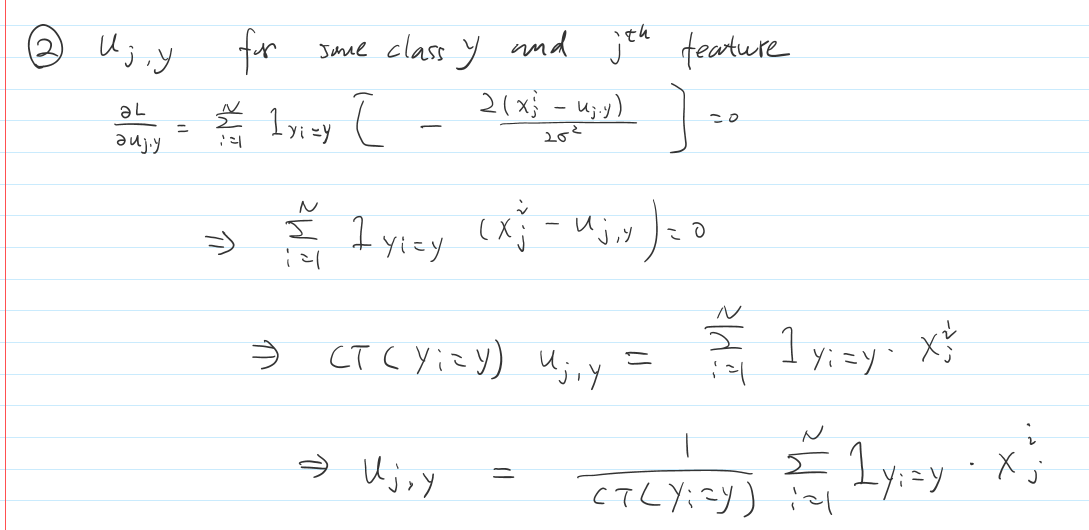
**3.4**

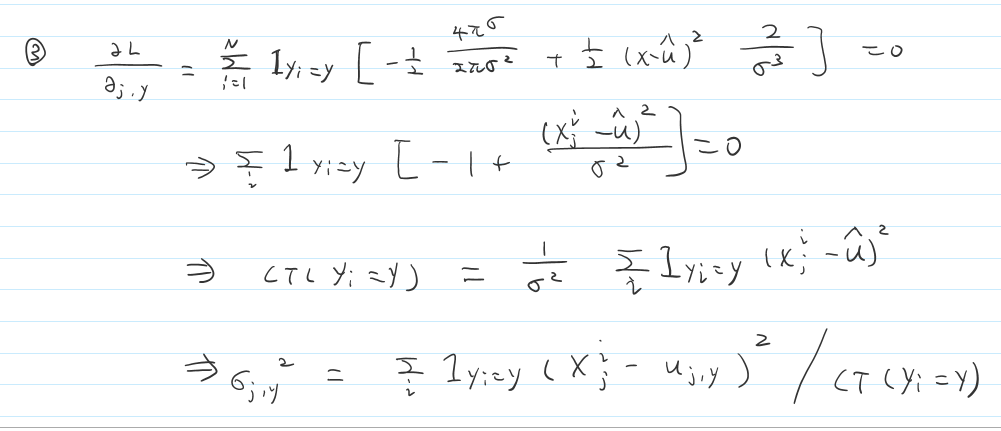
It is clear that L1 regularization tends to generate sparser results.

# Problem 3 Naïve Bayes

**4.1**







**4.2**

The accuracy is 0.6923

**4.3**

Any reasonable answer worth full credit. Example: we can use Normal-inverse-gamma distribution, which is the conjugate prior for normal distribution with unknown mean and variance parameters.

**4.4**

Any reasonable answer worth full credit. Example: This assumption inside naïve bayes is that all features are independent, which is almost impossible for a dataset with 60 features. Thus, its performance is kind of limited and we can see the test accuracy is not very high.