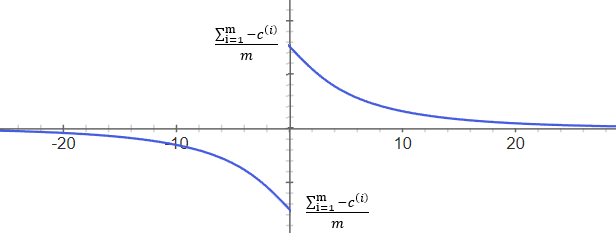
1. **Logistic Regression: Training stability**
2. dataset A converges very quickly, but dataset B doesn’t seem to converge ever.
3. when the datasets in logistic regression problem is perfectly separable, Logistic Regression will not converge. The proof below will show that if the data is perfectly separated, the gradient of loss function w.r.t. can never be 0, which means the log-likelihood loss can never reach to a maximum.

Based on the implementation in p01\_lr.py, the probability of a point is . For simplicity, assume only has one feature, is just a scaler. A correctly classified point should satisfy to make ; Otherwise the sample is misclassified. Gradient ascent used in p01\_lr.py is just to increase until it reaches the stationary point where .

To further simply the notation let’s assume , the log-likelihood loss can be written as:

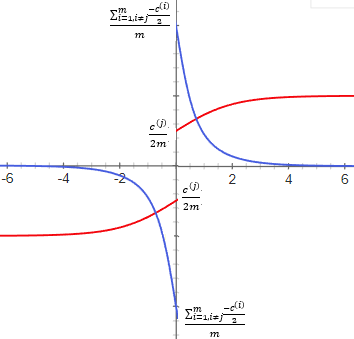
Take the derivative w.r.t. :

Since ,, we have for all . This explains why there is no convergence. The plot of looks something like the following. Note that when , we should have to make , which is the lower part of the diagram; when , we should have to make , which is the upper part of the diagram.

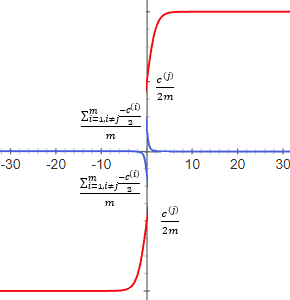


Now consider there is a misclassified point. Just one misclassified will make the algorithm converge. A misclassified point means in the assumption made above, , this will always make . Now we need to show that in this case will have a solution. Suppose the jth point is misclassified:

The left part of the above are all the correctly classified points which will take the value in () when , or () when , it will have the similar plot as plot; The right side of the above will take the value () when , ; () when . No matter whether is positive or negative, there will always be an intersection between and . Below is the plot of both and .



Note that there is a possibility that even though there is a misclassified point, never have a solution, like the following, there is no intersection between red and blue plot:



It turns out that this is impossible so long as the data points are generated IID from the same distribution; there is always a solution for in this case.

1. i. Using a different constant learning rate: Will not lead to convergence.

ii. Decreasing the learning rate over time (e.g. scaling the initial learning rate by , where t is the number of gradient descent iterations thus far): Will not lead to convergence.

iii. Linear scaling of the input features: Will not lead to convergence

iv. Adding a regularization term to the loss function: **Lead to convergence**.

v. Adding zero-mean Gaussian noise to the training data or labels: will have misclassified points. **Lead to convergence**.

1. In SVM, is going to be normalized and cannot be infinitely increased when trying to maximize the log-likelihood loss; strict separation issue would no longer be a problem.
2. **Model Calibration**
3. Prove calibration condition holds true for Logistic regression over range (a, b) = (0, 1). Firstly, the log-likelihood of all points is given by

The maximum likelihood parameter is just the solution given by . Note that

with . Use the fact that we include a bias term: , by observing the first component of and , we have:

1. Perfect calibration doesn’t mean perfect accuracy. If for any the property in the question holds true, consider two train examples with index i and j, they have different probability range and . By switching two samples’ probability ranges, the perfect calibration still holds true, but the two training examples will have different probability ranges separately.

Conversely if the model achieves perfect accuracy, it is perfectly calibrated. This can be explained by clapping to the probability of every single training sample, the perfect calibration condition holds true always.

1. What effect including L2 regularization in the logistic regression objective has on model calibration.

L2 regularization penalizes large value of , which makes not able to grow infinitely in order to get a larger probability, which means the calibration range is smaller than (0,1).

1. **Bayesian Interpretation of Regularization**
2. Show that if we assume that .
3. **Constructing** **kernels.**Apply Mercier’s theorem, to show a matrix is a kernel matrix, just need to show that the matrix is symmetric Positive Semi-Definite.
   1. :

It’s a kernel, it is obvious that if K1 and K2 are both symmetric and positive semi-definitive.

Frist, K is symmetric because, from:

Second, K is positive semi-definitive, because both K1 and K2 are PSD, K=K1+K2 should also be a PSD matrix.

* 1. : Not necessarily a kernel. Same as (a) it is symmetric, but not always positive definitive.
  2. : It is a kernel only when a >= 0.
  3. : It is a kernel only when a <= 0.
  4. : It is a kernel since:

Summarize the above 3 equations, which means K is symmetric.

Moreover, since both K1 and K2 are PSD, for any given vector x, we have , it implies that:

1. Kernelizing the Perceptron
   1. How you would apply the “kernel trick” to the perceptron
   2. Implement your approach by completing
   3. Which kernel performs badly and why does it fail?
2. Spam classification