## Week 1 – Linear Classifiers and Logistic Regression

1. A linear classifier assigns the predicted class based on the sign of .
   1. True.
2. For a conditional probability distribution over , where takes on two values (+1, -1), .
   1. True.
3. Which function does logistic regression use to “squeeze” the real line to [0, 1]?
   1. Logistic.
4. If , what is true about ?
   1. .
5. Consider training a 1 vs. all multiclass classifier for the problem of digit recognition using logistic regression. There are 10 digits, thus there are 10 classes. How many logistic regression classifiers will we have to train?
   1. 10.

## Week 2 – Training Linear Classifiers

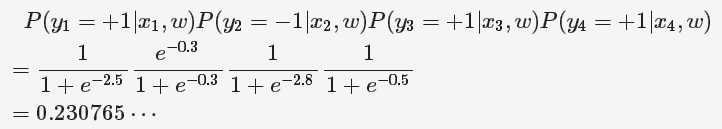
1. A linear classifier can only learn positive coefficients.
   1. False.
2. In order to train a logistic regression model, we find the weights that maximise the likelihood of the model.
   1. True.
3. The data likelihood is the product of the probability of the inputs given the weights andresponse .
   1. False.
4. Consider the situation where the inputs are 1-dimensional, the current estimates of the weights are and where is the intercept and is the weight for , and the data is:

|  |  |
| --- | --- |
| **x** | **y** |
| 2.5 | +1 |
| 0.3 | -1 |
| 2.8 | +1 |
| 0.5 | +1 |

Calculate the likelihood of this data.

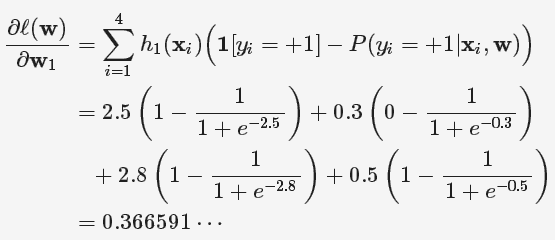
Given, and

|  |  |  |
| --- | --- | --- |
| **x** | **y** |  |
| 2.5 | +1 |  |
| 0.3 | -1 |  |
| 2.8 | +1 |  |
| 0.5 | +1 |  |



1. Calculate the derivate of the log-likelihood with respect to .

|  |  |  |  |
| --- | --- | --- | --- |
| **x** | **y** |  |  |
| 2.5 | +1 | 0.9241 | 0.1898 |
| 0.3 | -1 | 0.4256 |  |
| 2.8 | +1 | 0.9427 |  |
| 0.5 | +1 | 0.6225 |  |



1. What is true about gradient descent?
   1. It is an iterative algorithm.
   2. It finds the maximum by “hill climbing”.

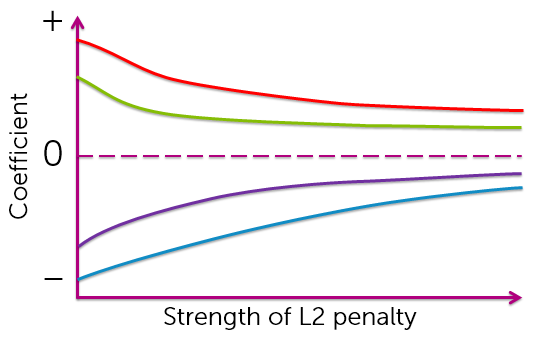
## Week 2 – Overfitting and Regularisation in Logistic Regression

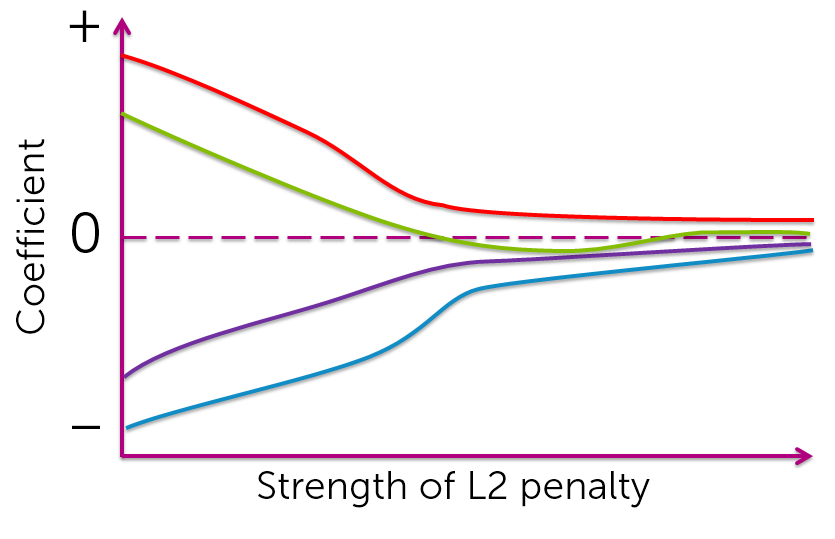
1. Consider four classifiers, whose classification performance is given by the following table:

|  |  |  |
| --- | --- | --- |
|  | Training error | Validation error |
| Classifier 1 | 0.2 | 0.6 |
| Classifier 2 | 0.8 | 0.6 |
| Classifier 3 | 0.2 | 0.2 |
| Classifier 4 | 0.5 | 0.4 |

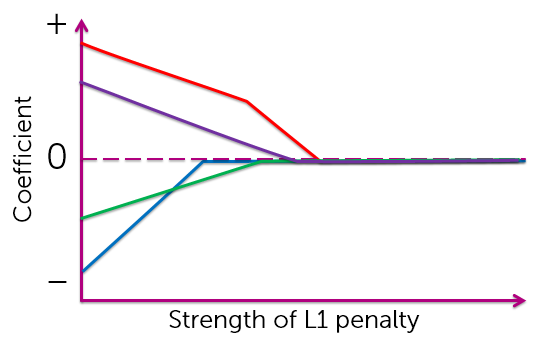
* 1. Classifier 1.

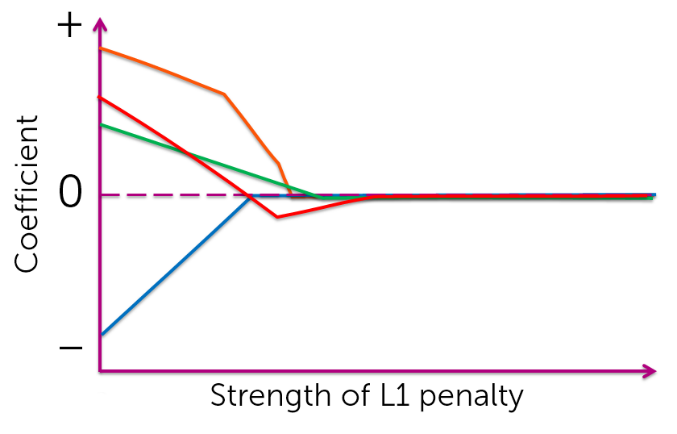
1. Suppose a classifier classifies 23,100 examples correctly and 1,900 examples incorrectly. What is the accuracy?
2. Accuracy and error measured on the same dataset always sum to 1.
   1. True.
3. What are correct descriptions of complex models?
   1. Complex models accommodate many features.
   2. Complex models tend to provide lower training error than simple models.
   3. Complex models tend to exhibit high variance in response to perturbation in the training data.
   4. Complex models tend to exhibit low bias, capturing many patterns in the training data that simple models may have missed.
4. What are symptoms of overfitting in the context of logistic regression?
   1. Large estimated coefficients.
   2. Complex decision boundary.
   3. Overconfident predictions of class probabilities.
5. Suppose we perform L2 regularized logistic regression to fit a sentiment classifier. Which of the following plots describe a possible coefficient path? (Assume that the algorithm runs for a wide range of L2 penalty values and each coefficient plot is zoomed out enough to capture all long-term trends.)





1. Suppose we perform L1 regularized logistic regression to fit a sentiment classifier. Which of the following plots describe a possible coefficient path? (Assume that the algorithm runs for a wide range of L1 penalty values and each coefficient plot is zoomed out enough to capture all long-term trends.)





1. In the context of L2 regularized logistic regression, which of the following occurs as we increase the L2 penalty ?
   1. The L2 norm of the set of coefficients gets smaller.
   2. The decision boundary becomes less complex.
   3. The classifier has lower variance.

## Week 3 – Decision Trees

Consider the following dataset:

|  |  |  |  |
| --- | --- | --- | --- |
| X1 | X2 | X3 | Y |
| 1 | 1 | 1 | +1 |
| 0 | 1 | 0 | -1 |
| 1 | 0 | 1 | -1 |
| 0 | 0 | 1 | +1 |

1. If we train a decision tree with this data, what feature will we split on at the root?
   1. X3 (error is 0.25 compared to 0.5 for both X1 and X2).
2. Fully train the decision tree (until each leaf has data points of the data output label). What is the depth?
   1. 3 (split on X3, then on X1 or X2 (it doesn’t matter which as the error is the same) and then the other variable).
3. What is the training error?
   1. 0.
4. Now consider a tree that splits on X1 at the root, then splits on X2 in the first level, and has leaves at the second level. What is the depth?
   1. 2.
5. What is the training error?
   1. 0,
6. Which has smaller depth, the first or the second decision tree?
   1. The second.
7. When deciding to split a node, we find the best feature to split on that minimises classification error.
   1. True.
8. If we have trained a decision tree, then remove one of the points in the dataset and re-train the decision tree. The split learned at the root of the tree:
   1. Could be the same or could be different.
9. Consider two datasets D1 and D2, where D2 has the same data points as D1, but has an extra feature for each data point. Let T1 be the decision tree trained with D1, and T2 be the tree trained with D2. Which of the following is true?
   1. Too little information to guarantee anything.
10. Which rule is more appropriate for splitting on real-valued features?
    1. Split using thresholds (e.g. income < 60k or income >= 60k).
11. Decision trees (with depth > 1) are always linear classifiers.
    1. True.
12. We are training a decision tree, and we are at a node. Each data point is , where are features, and is the label. The data at the node is:

|  |  |  |
| --- | --- | --- |
| X1 | X2 | Y |
| 0 | 1 | +1 |
| 1 | 0 | -1 |
| 0 | 1 | +1 |
| 1 | 1 | +1 |

Which feature results in the best split?

* 1. X2 (error is 0 compared to 0.25 for X1)

1. If you are training a decision tree, and you are at a node in which all of its data has the same y-value, you should:
   1. Create a leaf that predicts the y-value of all the data.
2. Logistic regression with polynomial degree 1 features will always have equal or lower training error than decision stumps (depth 1 decision trees).
   1. True.

## Week 4 - Preventing Overfitting in Decision Trees

1. When training decision trees, smaller depths USUALLY lead to lower training errors.
   1. False.
2. If no two data points have the same input values, we can always train a decision tree that achieves 0 training error.
   1. True.
3. If one decision tree has lower training error than another, then it will always have a better test error.
   1. False.
4. What is true for decision trees.
   1. Model complexity increases with depth.
5. Pruning and early stopping in decision trees is used to:
   1. Combat overfitting.
6. What are early stopping methods?
   1. Stop when the tree hits a certain depth.
   2. Stop when a node has too few data points (minimum node size).
   3. Stop when the best split results in too small an error reduction.
7. Consider a decision tree with a minimum node size = 1000, and a second decision tree trained on the same dataset and parameters except with a minimum node size = 100. What is always true?
   1. The depth of the second tree will be at least the depth of the first.
   2. The number of nodes in the second tree will be at least as great as the number of nodes in the first.
   3. The training error of the second tree will be equal to or smaller than the training error of the first.
8. Suppose we are training a decision tree and we are at a node. The data at this node is:

|  |  |  |
| --- | --- | --- |
| X1 | X2 | Y |
| 0 | 1 | +1 |
| 1 | 0 | +1 |
| 0 | 1 | +1 |
| 1 | 1 | -1 |

What is the classification error at this node (assuming a majority classifier)?

* 1. 0.25

1. If we split on X1, what is the classification error?
   1. 0.25
2. If we split on X2, what is the classification error?
   1. 0.25.
3. If our parameter for minimum gain in error reduction is 0.1, do we split or stop early?
   1. Stop early.

## Week 4 – Handling Missing Data

1. Skipping data points (i.e., skipping rows of the data) that have missing features only works when the learning algorithm we are using is decision tree learning.
   1. False.
2. What are potential downsides of skipping features with missing values (i.e., skipping columns of the data) to handle missing data?
   1. So many features are skipped that accuracy can degrade.
   2. If an input at prediction time has a feature missing that was always present during training, this approach is not applicable.
3. It’s always better to remove missing data points (i.e., rows) as opposed to removing missing features (i.e., columns).
   1. False.
4. Consider a dataset with N training points. After imputing missing values, the number of data points in the data set is:
   1. N.
5. Consider a dataset with D features. After imputing missing values, the number of features in the data set is:
   1. D.
6. Which of the following are always true when imputing missing data?
   1. Imputed values can be used in any classification algorithm.
   2. Imputed values can be used when there is missing data at prediction time.
7. Consider data that has binary features (i.e. the feature values are 0 or 1) with some feature values of some data points missing. When learning the best feature split at a node, how would we best modify the decision tree learning algorithm to handle data points with missing values for a feature?
   1. We choose to assign missing values to the branch of the tree (either the one with feature value equal to 0 or with feature value equal to 1) that minimises classification error.

## Week 5 – Boosting

1. What is NOT an ensemble method?
   1. Single decision trees.
2. Each binary classifier in an ensemble makes predictions on an input as listed in the table below. Based on the ensemble coefficients also in the table, what is the final ensemble model’s prediction for ?

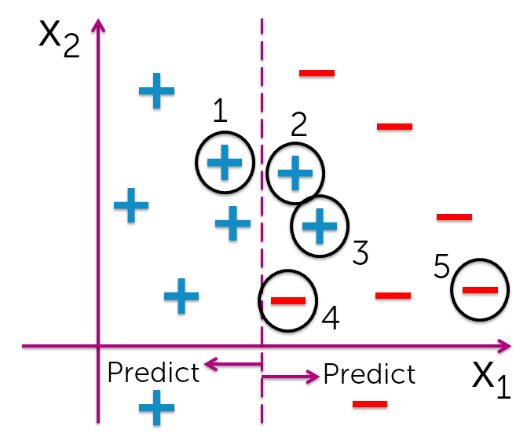
|  |  |  |
| --- | --- | --- |
|  | Classifier coefficient | Prediction for |
| Classifier 1 | 0.61 | +1 |
| Classifier 2 | 0.53 | -1 |
| Classifier 3 | 0.88 | -1 |
| Classifier 4 | 0.34 | +1 |

1. Boosted trees tend to be more robust to overfitting than decision trees.
   1. True.
2. AdaBoost focuses on data points it incorrectly predicted by increasing those weights in the dataset.
   1. True.
3. Let be the coefficient for a weak learner . Which of the following conditions must be true so that ?
4. If you were using AdaBoost and in an iteration of the algorithm were faced with the following classifiers, which one would you be more inclined to use in the ensemble?
5. Suppose you are training a decision stump in an iteration of AdaBoost and are at a node. Each data point is where are features and is the label. Also included are the weights of the data:

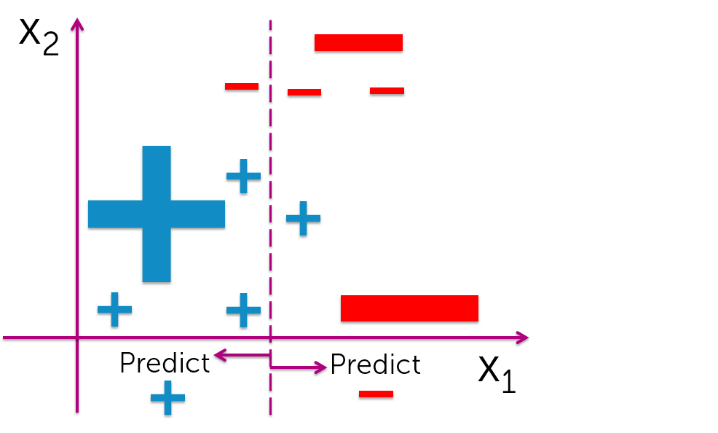
|  |  |  |  |
| --- | --- | --- | --- |
| Weight | X1 | X2 | Y |
| 0.3 | 0 | 1 | +1 |
| 0.35 | 1 | 0 | -1 |
| 0.1 | 0 | 1 | +1 |
| 0.25 | 1 | 1 | +1 |

Suppose we assign the same class label to all data in this node – the class label with the greater total weight. What is the weighted error at the node?

1. After each iteration of AdaBoost, the weights on the data points are typically normalised to sum to 1. This is done because:
   1. Of issues with numerical instability (underflow/overflow).
2. If we train a series of weak binary classifiers using AdaBoost and in one iteration, it produces the following decision boundary, which of the five circled data points will receive higher weight in the next iteration?



* 1. 2 and 3.

1. Suppose we are running AdaBoost using decision tree stumps. At a particular iteration, the data points have weights according to the image (larger points indicate heavier weights). Which decision tree stump is most likely to be fit in the next iteration?
   1. 
2. AdaBoost can boost any kind of classifier, not just a decision tree stump.
   1. True.

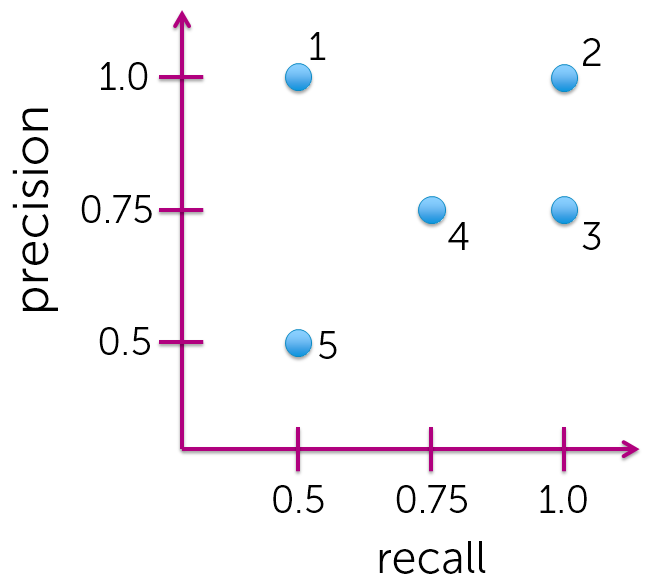
## Week 6 – Precision-Recall

1. Suppose a binary classifier produced the following confusion matrix:

|  |  |  |
| --- | --- | --- |
|  | **Predicted Positive** | **Predicted Negative** |
| **Actual Positive** | 5600 | 40 |
| **Actual Negative** | 1900 | 2460 |

What is the **accuracy** of this classifier?

1. This classifier is better than random guessing.
   1. True. (
2. This classifier is better than the majority class classifier.
   1. True.
3. Which of the following points in the precision-recall space correspond to this classifier?

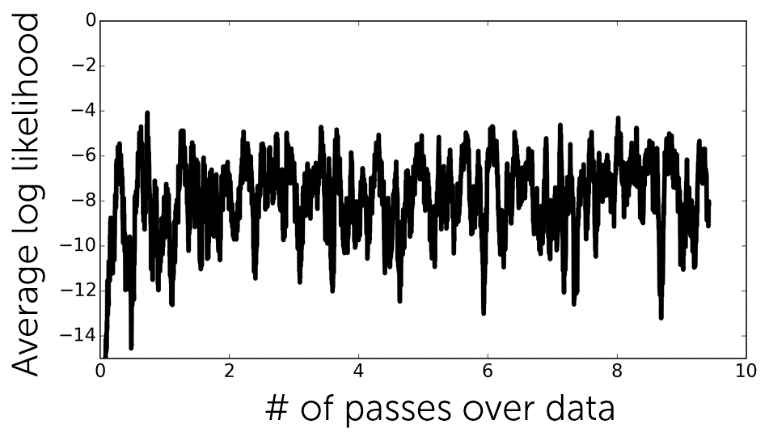


3

1. What best describes this classifier.
   1. It is optimistic.
2. Suppose we are fitting a logistic regression model on a dataset where the vast majority of the data points are labelled as positive. To compensate for overfitting the dominant class, we should:
   1. Require higher confidence for positive predictions.
3. It is often the case that false positives and false negatives incur different costs. In situations where false negatives cost much more than false positives, we should:
   1. Require lower confidence for positive predictions.
4. We are interested in reducing the number of false negatives. Which metric should we primarily look at?
   1. Recall.
5. Suppose we set the threshold for positive predictions at 0.9. Which is the lowest score that is classified as positive?
   1. 2.2

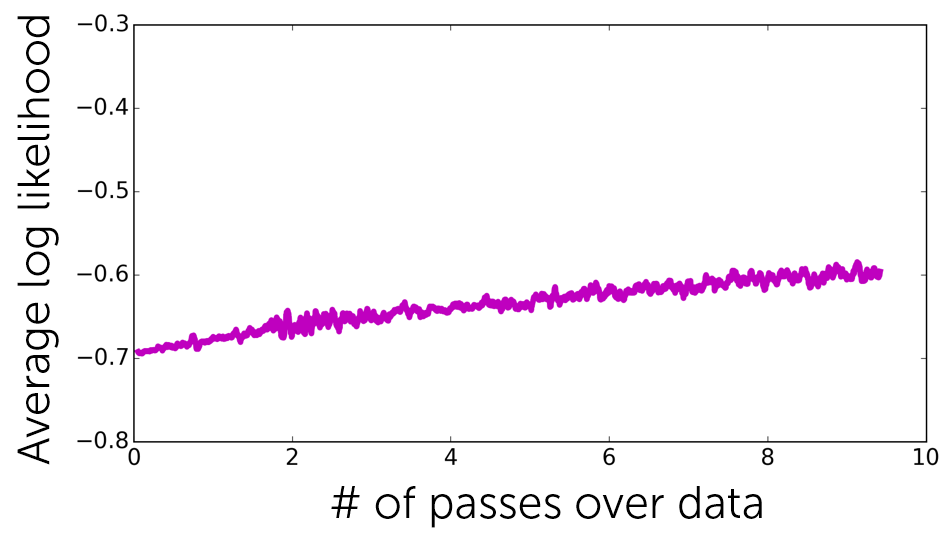
## Week 7 – Scaling to Huge Datasets and Online Learning

1. Stochastic gradient ascent often requires fewer passes over the dataset than batch gradient ascent to achieve a similar log likelihood.
   1. True.
2. Choosing a large batch size results in less noisy gradients.
   1. True.
3. The set of coefficients obtained at the last iteration represents the best coefficients found so far.
   1. False.
4. Suppose you obtained the following log likelihood plot after running stochastic gradient ascent. What would help the most to improve the rate of convergence?



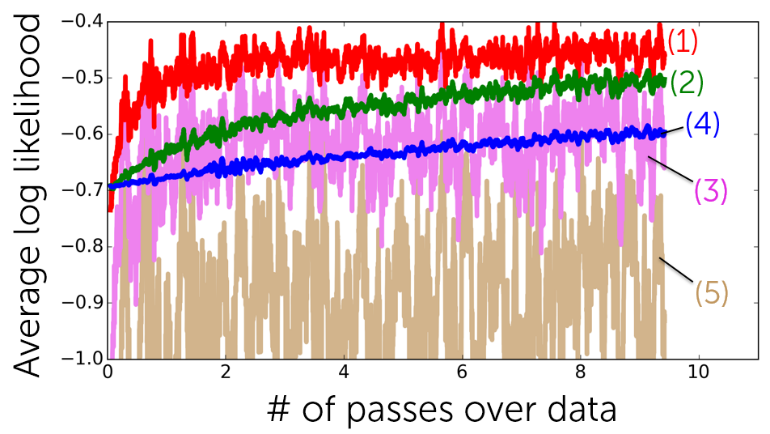
* 1. Decrease step size.

1. Suppose you obtained the following log likelihood plot after running stochastic gradient ascent. What would help the most to improve the rate of convergence?



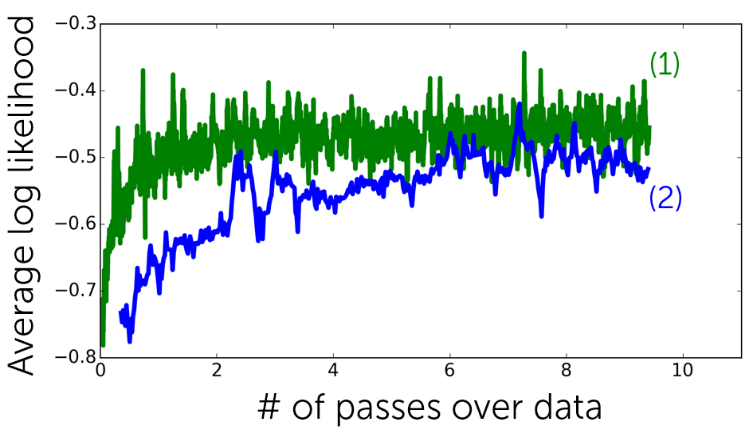
* 1. Increase step size.

1. Suppose it takes about 1 millisecond to compute a gradient descent for a single example. You run an online advertising company and would like to do online learning via mini-batch stochastic gradient ascent. If you aim to update the coefficients once every 5 minutes, how many examples can you cover in each update? Overhead and other operations take up to 2 minutes, so you only have 3 minutes for the coefficient update.
   1. 180,000 (there are 60,000 milliseconds in every minute).
2. In the search for an optimal step size, you experiment with multiple step sizes and obtain the following convergence plot. Which line corresponds to the best step size?



* 1. (1)

1. Suppose you run stochastic gradient ascent with two different batch sizes. Which of the two lines corresponds to the smaller batch size (assuming both are using the same step size)?



* 1. (1)

1. What are NOT benefits of stochastic gradient ascent over batch gradient ascent?
   1. Log likelihood of data improves monotonically.
   2. Stochastic gradient ascent is highly robust with respect to parameter choices.
2. Suppose we run the stochastic gradient ascent algorithm with batch size of 100. To make 10 passes over a dataset consisting of 15,400 examples, how many iterations does it need to run?
   1. 15,400 / 100 = 1540