ML w6 Evaluating a learning algorithm

Deciding a learning algorithm

large error.

more examples

smaller sets of features

try getting additional features.

Try add polynomial features.

Machine learning diagnostic.

take time but very fruitful.

Evaluating a Hypothesis

overfit $\Longleftrightarrow$ Training set and Test set

70% 30%

Linear Regression

Training error $J(\theta)$

→ Compute test set error

$$J_{test}(\theta) = \frac{1}{2m_{test}} \sum_{i=1}^{m_{test}} \left(h_\theta(x_{test}^{(i)}) - y_{test}^{(i)} \right)$$

→ Logistic Regression

$$J_{test}(\theta) = -\frac{1}{m_{test}} \sum_{i=1}^{m_{test}} y_{test}^{(i)} \log h_\theta(x_{test}^{(i)})$$

$$+(1-y_{test}^{(i)}) \log h_\theta(x_{test}^{(i)})$$

$$\text{err}(h_\theta(x), y) = \begin{cases} 1 & \text{if } h_\theta(x) \geq 0.5 \quad y = 0 \\ 0 & \text{or if } h_\theta(x) < 0.5 \quad y = 1 \end{cases} \text{error}$$

$$\text{Test error} = \frac{1}{m_{test}} \sum_{i=1}^{m_{test}} \text{err}(h_\theta(x_{test}^{(i)}), y^{(i)}).$$

Model Selection and Train/Validation/Test Sets.

overfit → too small error.

$h_\theta(x) = \theta_0 + \theta_1 x$ $\quad \theta^{(1)} \rightarrow J_{test}(\theta^{(1)})$

$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2$ $\quad \theta^{(2)} \rightarrow J_{test}(\theta^{(2)})$

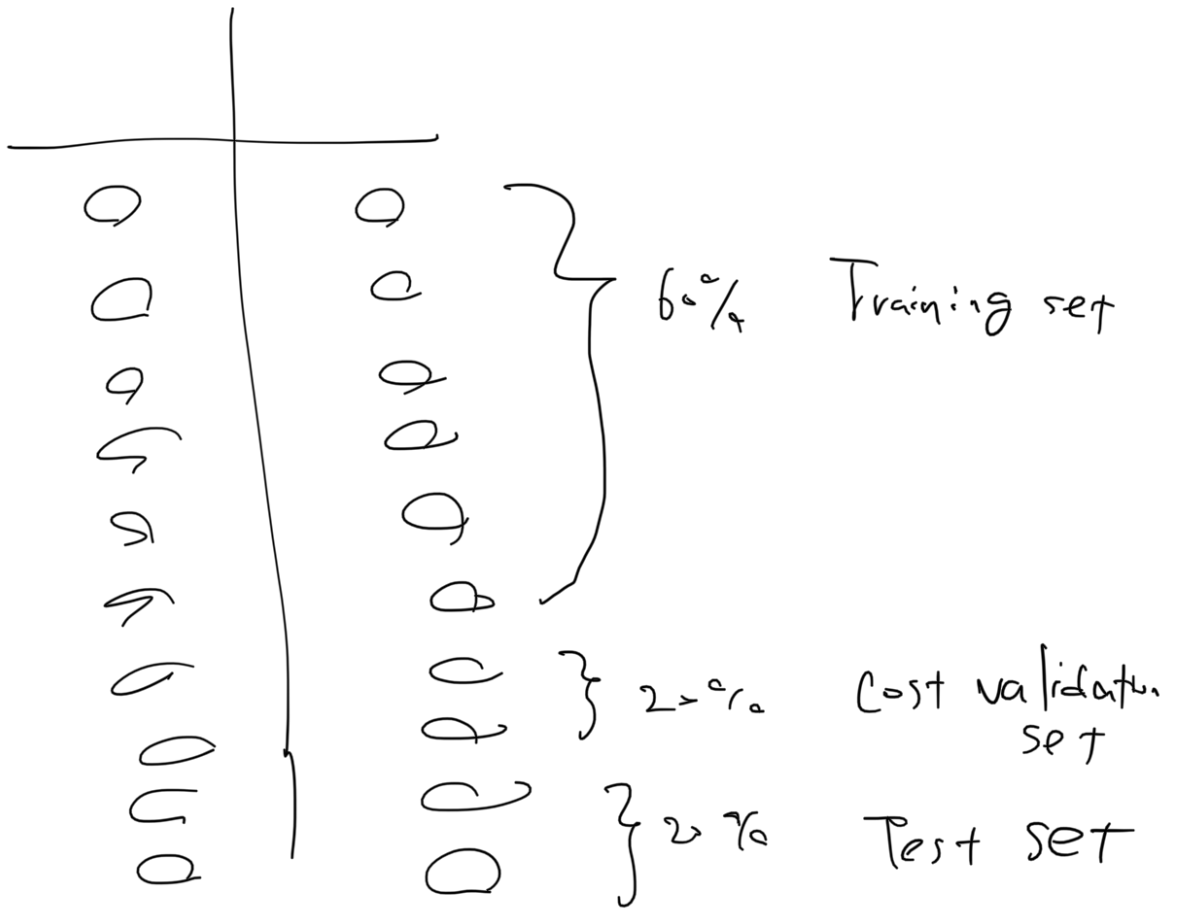
$\vdots$

$h_\theta(x) = \theta_0 + \theta_1 x + \cdots + \theta_{10} x^{10}$ $\quad \theta^{(10)} \rightarrow J_{test}(\theta^{(10)})$

⇩

Choose $\theta_0 + \cdots \theta_5 x^5$ ∈ less error

Problem: $J_{test}(\theta^{(5)})$ is optimistic estimate of generalization.



Training error $J_{train}(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$

Cross validation error ← use cross validation data

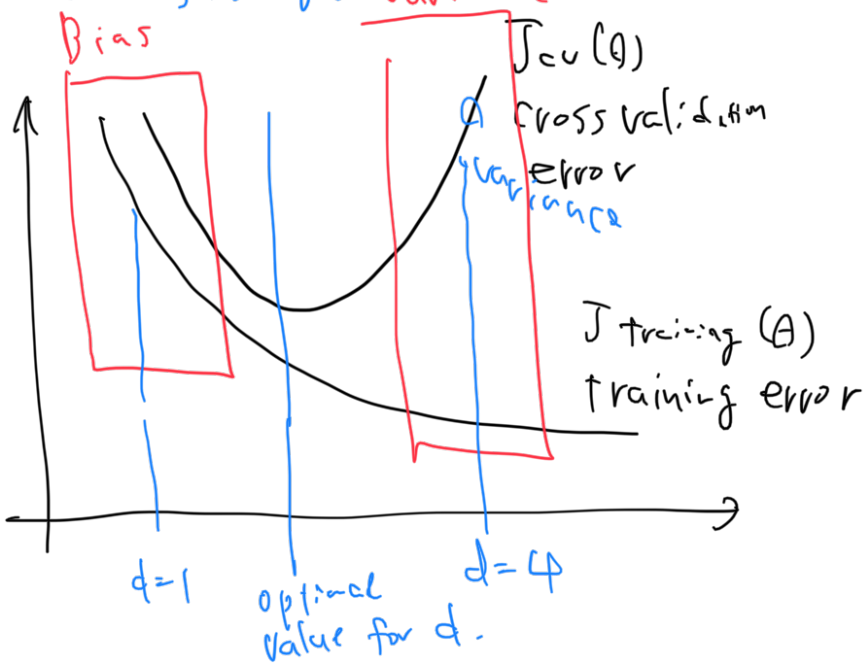
Test set error

1. Optimize the parameter in θ using the training set for each polynomial degree
2. Find the polynomial degree d with the least error using cross validation set
3. Estimate the generalization error using the

test set with $J_{test}(\theta^{(d)})$

Bias vs. Variance

High bias $\longleftrightarrow$ right $\longleftrightarrow$ high variance
(underfitting) Variance (over fitting)



Bias (under fit)

- $J_{train}(\theta)$ will be high
- $J_{cv}(\theta) \approx J_{train}(\theta)$

Variance (over fit)

- $J_{train}(\theta)$ will be low
- $J_{cv}(\theta) \gg J_{train}(\theta)$

Choosing the regularization parameter λ

Model: $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4 \cdots$

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 + \frac{\lambda}{2m}\sum_{j=1}^{m}\theta_j^2$$

1 try $\lambda = 0$

2 $= 0.01$

3 $= 0.02$

4 $= 0.04$

5 $= 0.08$

$\lambda = 10$ ...

$\min_\theta J(\theta) \rightarrow \theta^{(1)} \rightarrow J_{cv}(\theta^{(1)})$

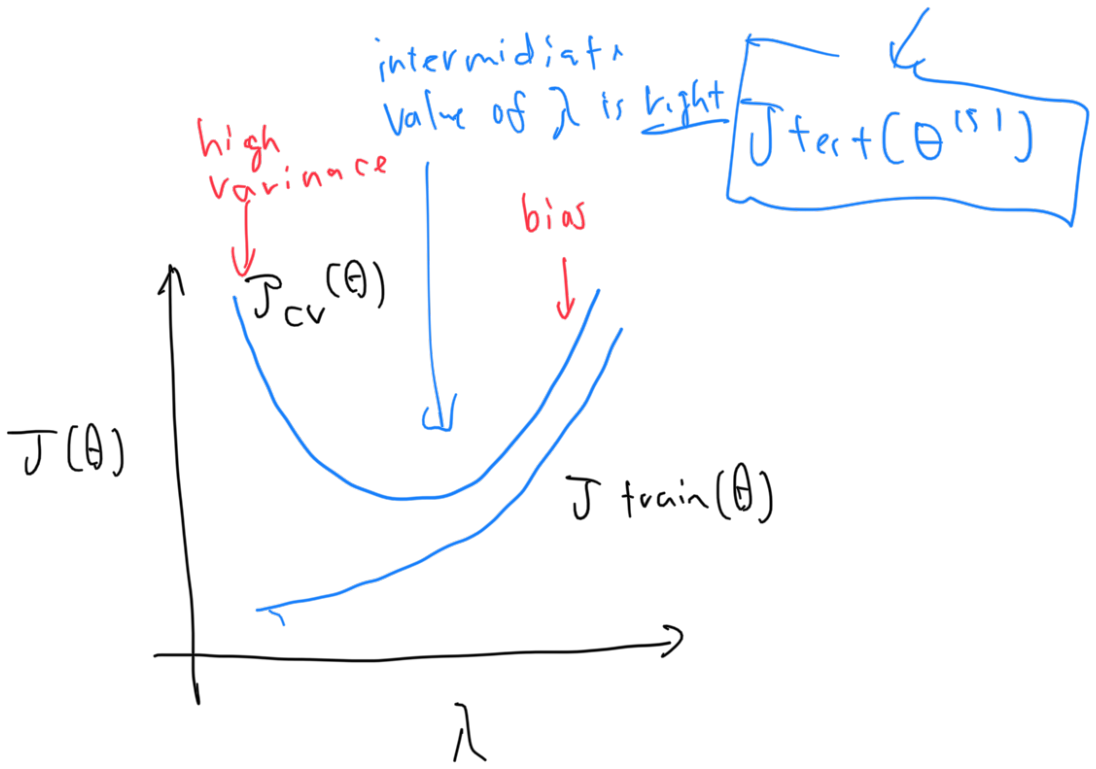
$\min_\theta J(\theta) \rightarrow \theta^{(2)} \rightarrow J_{cv}(\theta^{(2)})$

$\theta^{(3)}$

$J_{test}(\theta^{(5)}) \rightarrow J_{cv}(\theta^{(5)})$

lowest cross validation polynomial curve.

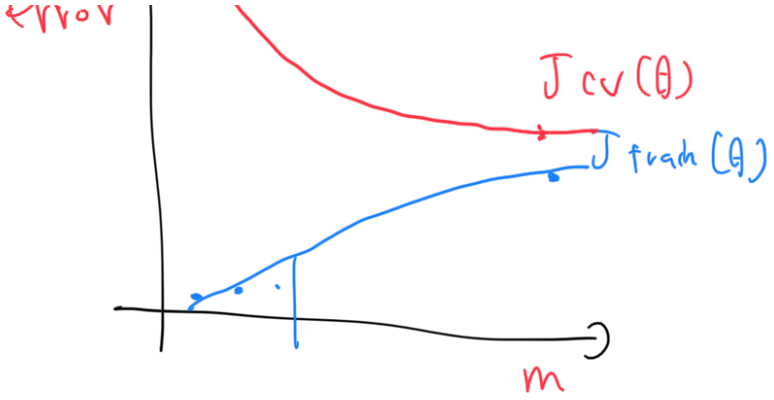
pick θ^5



Learning Curve

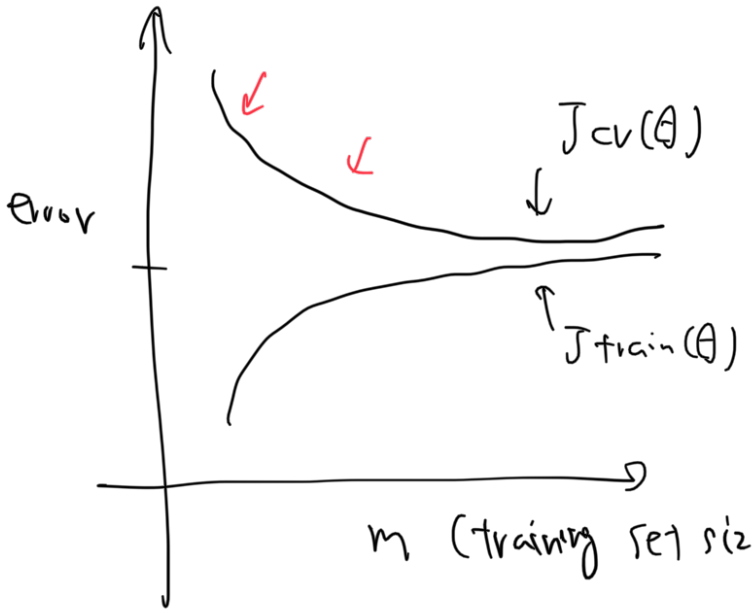
$$J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

$$J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}\left(h_\theta(x_{cv}^{(i)}) - y_{cv}^{(i)}\right)^2$$

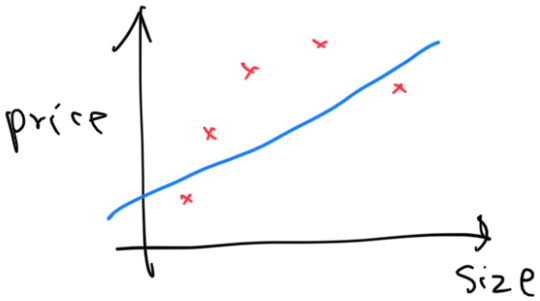


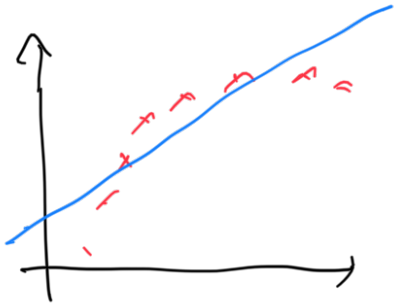
↑
Training size is low,
error is small. (easy to fit)

High bias



$h_\theta(x) = \theta_0 + \theta_1 x$



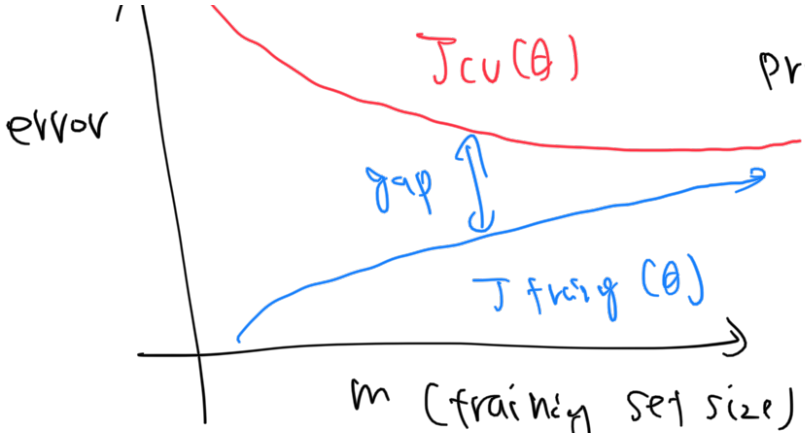


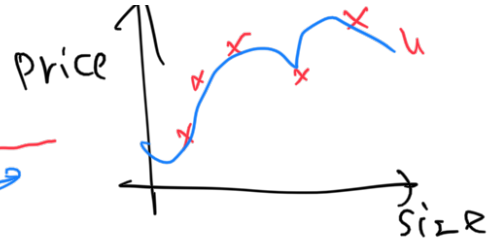
If learning algorithm is suffering from high bias, getting more training data will not (by itself) help much

High variance

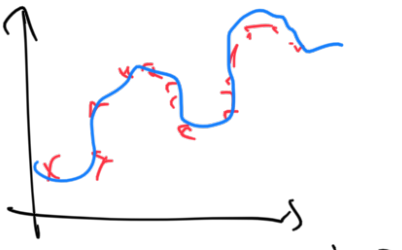
$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_{100} x^{100}$
(and small λ)

↑





If a learning algorithm is suffering from high variance, getting more training data is likely to help



Deciding what to do next

- Get more training examples : fix : high variance.
- Try smaller sets of features : fix : high variance.
- Try getting additional features : fix : high bias
- Try adding polynomial features : fix : high bias.
- Try decrease λ : fix : high bias
  increase λ : fix : high : variance.

(λ ↑ regularization parameter.)

Neural Networks and Overfitting.

computationally cheaper

↑

likely underfitting

(high bias)

:

decrease λ

computationally more expensive

↑

likely overfitting

↑

(high variance)

:

increase λ

Building a Spam Classifier.

Prioritizing what to work on

$$x = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} \begin{matrix} \text{a} \\ \text{buy} \\ \text{deal} \\ \\ \\ \\ \\ \end{matrix} \quad x \in \mathbb{R}^{100}$$

$$x_j = \begin{cases} 1 & \text{if word } j \text{ appears in email} \\ 0 & \text{otherwise} \end{cases}$$

- Collect lots of data
- Develop sophisticated feature
- Develop algorithm to process your input in different ways.

Error Analysis

Recommended approach

- Start with simple algorithm quickly, with cross validation error
- Plot learning curve.

- Error Analysis: manually examine the examples.

- Numerical evaluation

if we develop new features by examin test set, then we end up choosing features that will work specifically with test set.

m_cv = 500 examples.

Algorithm misclassify 100 exaples.

↓ manually examine.

- what type?
- what cues (features) you think would have help?

↓

no longer good example

Numerical evaluation

Can use "steming" idea (software)

with steming ⟷ without steming

5% error 3% error

Distinguish upper/lower cases ... etc.

Handling skewed data.

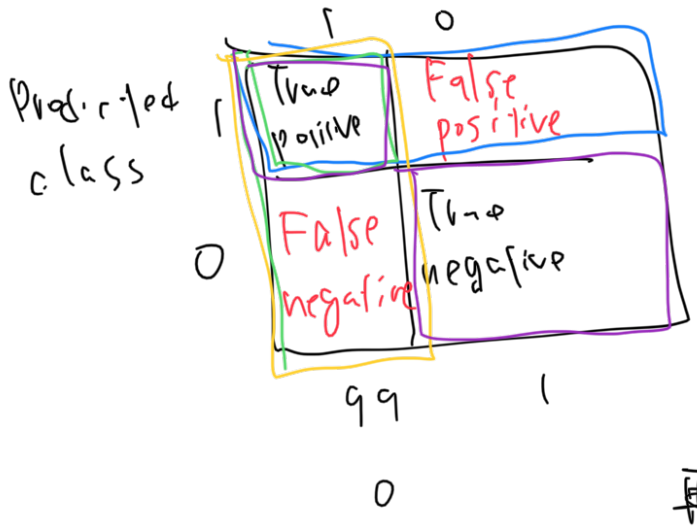
Error metrics for skewed classes.

Find 1% error ...

99.2% accuracy : 0.8% error
99.5% accuracy : 0.5% error ↓ 50% error down

Precision / Recall gives us insight about how algorithm work

Actual class



$$\text{Precision} = \frac{\text{True Positive}}{\#\text{ predicted as positive}} = \frac{\text{True Positive}}{\text{True positive + False positive}}$$

精準率

$$\text{Recall} = \frac{\text{True Positive}}{\#\text{ actual positive}} = \frac{\text{True Positive}}{\text{True positive + False negative}}$$

再現率

$$\text{accuracy} = \frac{\text{true positive + true negative}}{\text{total number}}$$

Trading off precision and recall

- avoid false negative ...

Predict 1 if $h(x) \geq 0.5$, 0.7 ~ 0.9, 0.3

We want to predict $y=1$ (cancer) only if confident

→ High precision, lower recall

We want to avoid missing too many cases of cancer

→ High recall, lower precision

F1 Score (F Score)

	Precision	Recall	Average	F1 Score
Algorithm 1	0.5	0.4	0.45	0.444
2	0.7	0.1	0.4	0.175
3	0.02	1.0	0.51	0.0392

Predict $y=1$ all the time

$$F_1 \text{ Score} = 2\frac{PR}{P+R}$$

$P=0$ or $R=0$ → F_1 Score $=0$

$P=1$ or $R=1$ → F_1 Score $=1$

Data for machine learning

more data, more accuracy of any algorithm

Large data rationale.

algorithm with many feature :
neural network hidden unit

$J_{train}(\theta)$ will be small

$J_{train}(\theta) \approx J_{test}(\theta)$

J_{test} will be small.

1. You are working on a spam classification system using regularized logistic regression. "Spam" is a positive class (y = 1) and "not spam" is the negative class (y = 0). You have trained your classifier and there are m = 1000 examples in the cross-validation set. The chart of predicted class vs. actual class is:

	Actual Class: 1	Actual Class: 0
Predicted Class: 1	85	890
Predicted Class: 0	15	10

For reference:

- Accuracy = (true positives + true negatives) / (total examples)
- Precision = (true positives) / (true positives + false positives)
- Recall = (true positives) / (true positives + false negatives)
- F_1 score = (2 * precision * recall) / (precision + recall)

What is the classifier's precision (as a value from 0 to 1)?

Enter your answer in the box below. If necessary, provide at least two values after the decimal point.

0.08

1. You are working on a spam classification system using regularized logistic regression. "Spam" is a positive class (y = 1) and "not spam" is the negative class (y = 0). You have trained your classifier and there are m = 1000 examples in the cross-validation set. The chart of predicted class vs. actual class is:

1 / 1点

	Actual Class: 1	Actual Class: 0
Predicted Class: 1	85	890
Predicted Class: 0	15	10

For reference:

- Accuracy = (true positives + true negatives) / (total examples)
- Precision = (true positives) / (true positives + false positives)
- Recall = (true positives) / (true positives + false negatives)
- F_1 score = (2 * precision * recall) / (precision + recall)

What is the classifier's precision (as a value from 0 to 1)?

Enter your answer in the box below. If necessary, provide at least two values after the decimal point.

0.08

正解
There are 85 true positives and 890 false positives, so precision is 85 / (85 + 890) = 0.087.

2. Suppose a massive dataset is available for training a learning algorithm. Training on a lot of data is likely to give good performance when two of the following conditions hold true. 1点

Which are the two?

- [ ] We train a learning algorithm with a small number of parameters (that is thus unlikely to overfit).
- [ ] We train a model that does not use regularization.
- [x] We train a learning algorithm with a large number of parameters (that is able to learn/represent fairly complex functions).
- [x] The features x contain sufficient information to predict y accurately. (For example, one way to verify this is if a human expert on the domain can confidently predict y when given only x).

2. Suppose a massive dataset is available for training a learning algorithm. Training on a lot of data is likely to give good performance when two of the following conditions hold true.

Which are the two?

- [ ] We train a learning algorithm with a small number of parameters (that is thus unlikely to overfit).
- [ ] We train a model that does not use regularization.
- [x] We train a learning algorithm with a large number of parameters (that is able to learn/represent fairly complex functions).

正解
You should use a "low bias" algorithm with many parameters, as it will be able to make use of the large dataset provided. If the model has too few parameters, it will underfit the large training set.

- [x] The features x contain sufficient information to predict y accurately. (For example, one way to verify this is if a human expert on the domain can confidently predict y when given only x).

正解
It is important that the features contain sufficient information, as otherwise no amount of data can solve a learning problem in which the features do not contain enough information to make an accurate prediction.

3. Suppose you have trained a logistic regression classifier which is outputting $h_\theta(x)$. 1点

Currently, you predict 1 if $h_\theta(x) \geq \text{threshold}$, and predict 0 if $h_\theta(x) < \text{threshold}$, where currently the threshold is set to 0.5.

Suppose you **decrease** the threshold to 0.3. Which of the following are true? Check all that apply.

- [x] The classifier is likely to now have lower precision.
- [ ] The classifier is likely to now have lower recall.
- [ ] The classifier is likely to have unchanged precision and recall, and thus the same F_1 score.
- [ ] The classifier is likely to have unchanged precision and recall, but higher accuracy.

3. Suppose you have trained a logistic regression classifier which is outputting $h_\theta(x)$. 1 / 1点

Currently, you predict 1 if $h_\theta(x) \geq \text{threshold}$, and predict 0 if $h_\theta(x) < \text{threshold}$, where currently the threshold is set to 0.5.

Suppose you **decrease** the threshold to 0.3. Which of the following are true? Check all that apply.

- [x] The classifier is likely to now have lower precision.

正解
Lowering the threshold means more y = 1 predictions. This will increase both true and false positives, so precision will decrease.

- [ ] The classifier is likely to now have lower recall.
- [ ] The classifier is likely to have unchanged precision and recall, and thus the same F_1 score.
- [ ] The classifier is likely to have unchanged precision and recall, but higher accuracy.

4. Suppose you are working on a spam classifier, where spam emails are positive examples ($y = 1$) and non-spam emails are negative examples ($y = 0$). You have a training set of emails in which 99% of the emails are non-spam and the other 1% is spam. Which of the following statements are true? Check all that apply.

- [ ] If you always predict non-spam (output $y = 0$), your classifier will have an accuracy of 99%.
- [x] If you always predict non-spam (output $y = 0$), your classifier will have 99% accuracy on the training set, but it will do much worse on the cross validation set because it has overfit the training data.
- [x] A good classifier should have both a high precision and high recall on the cross validation set.
- [ ] If you always predict non-spam (output $y = 0$), your classifier will have 99% accuracy on the training set, and it will likely perform similarly on the cross validation set.

4. Suppose you are working on a spam classifier, where spam emails are positive examples ($y = 1$) and non-spam emails are negative examples ($y = 0$). You have a training set of emails in which 99% of the emails are non-spam and the other 1% is spam. Which of the following statements are true? Check all that apply.

- [ ] If you always predict non-spam (output $y = 0$), your classifier will have an accuracy of 99%.

- [x] If you always predict non-spam (output $y = 0$), your classifier will have 99% accuracy on the training set, but it will do much worse on the cross validation set because it has overfit the training data.

  ⊗ **これを選択しないでください**
  The classifier achieves 99% accuracy because of the skewed classes in the data, not because it is overfitting the training set. Thus, it is likely to perform just as well on the cross validation set.

- [x] A good classifier should have both a high precision and high recall on the cross validation set.

  ✓ **正解**
  For data with skewed classes like these spam data, we want to achieve a high F_1 score, which requires high precision and high recall.

- [ ] If you always predict non-spam (output $y = 0$), your classifier will have 99% accuracy on the training set, and it will likely perform similarly on the cross validation set.

5. Which of the following statements are true? Check all that apply.

- [x] Using a **very large** training set makes it unlikely for model to overfit the training data.
- [ ] After training a logistic regression classifier, you **must** use 0.5 as your threshold for predicting whether an example is positive or negative.
- [x] It is a good idea to spend a lot of time collecting a **large** amount of data before building your first version of a learning algorithm.
- [x] If your model is underfitting the training set, then obtaining more data is likely to help.
- [x] On skewed datasets (e.g., when there are more positive examples than negative examples), accuracy is not a good measure of performance and you should instead use F_1 score based on the precision and recall.

5. Which of the following statements are true? Check all that apply. 0 / 1点

- [x] Using a **very large** training set makes it unlikely for model to overfit the training data.

  正解
  A sufficiently large training set will not be overfit, as the model cannot overfit some of the examples without doing poorly on the others.

- [ ] After training a logistic regression classifier, you **must** use 0.5 as your threshold for predicting whether an example is positive or negative.

- [x] It is a good idea to spend a lot of time collecting a **large** amount of data before building your first version of a learning algorithm.

  これを選択しないでください
  You cannot know whether a huge dataset will be important until you have built a first version and find that the algorithm has high variance.

- [x] If your model is underfitting the training set, then obtaining more data is likely to help.

  これを選択しないでください
  If the model is underfitting the training data, it has not captured the information in the examples you already have. Adding further examples will not help any more.

- [x] On skewed datasets (e.g., when there are more positive examples than negative examples), accuracy is not a good measure of performance and you should instead use F_1 score based on the precision and recall.

  正解
  You can always achieve high accuracy on skewed datasets by predicting the most the same output (the most common one) for every input. Thus the F_1 score is a better way to measure performance.

Take2

1. You are working on a spam classification system using regularized logistic regression. "Spam" is a positive class (y = 1) and "not spam" is the negative class (y = 0). You have trained your classifier and there are m = 1000 examples in the cross-validation set. The chart of predicted class vs. actual class is:

	Actual Class: 1	**Actual Class: 0**
Predicted Class: 1	85	890
Predicted Class: 0	15	10

For reference:

- Accuracy = (true positives + true negatives) / (total examples)
- Precision = (true positives) / (true positives + false positives)
- Recall = (true positives) / (true positives + false negatives)
- F_1 score = (2 * precision * recall) / (precision + recall)

What is the classifier's accuracy (as a value from 0 to 1)?

Enter your answer in the box below. If necessary, provide at least two values after the decimal point.

0.095

2. Suppose a massive dataset is available for training a learning algorithm. Training on a lot of data is likely to give good performance when two of the following conditions hold true. 1点

Which are the two?

- [ ] We train a learning algorithm with a small number of parameters (that is thus unlikely to overfit).
- [x] We train a learning algorithm with a large number of parameters (that is able to learn/represent fairly complex functions).
- [x] The features x contain sufficient information to predict y accurately. (For example, one way to verify this is if a human expert on the domain can confidently predict y when given only x).
- [ ] When we are willing to include high order polynomial features of x (such as x_1^2, x_2^2, x_1x_2, etc.).

3. Suppose you have trained a logistic regression classifier which is outputing $h_\theta(x)$. 1点

Currently, you predict 1 if $h_\theta(x) \geq \text{threshold}$, and predict 0 if $h_\theta(x) < \text{threshold}$, where currently the threshold is set to 0.5.

Suppose you **increase** the threshold to 0.9. Which of thefollowing are true? Check all that apply.

- [ ] The classifier is likely to have unchanged precision and recall, and thus the same F_1 score.
- [x] The classifier is likely to now have higher precision.
- [ ] The classifier is likely to have unchanged precision and recall, but higher accuracy.
- [ ] The classifier is likely to now have higher recall.

4. Suppose you are working on a spam classifier, where spam emails are positive examples ($y = 1$) and non-spam emails are negative examples ($y = 0$). You have a training set of emails in which 99% of the emails are non-spam and the other 1% is spam. Which of the following statements are true? Check all that apply.

- [ ] If you always predict non-spam (output $y = 0$), your classifier will have a recall of 0%.
- [ ] If you always predict spam (output $y = 1$), your classifier will have a recall of 0% and precision of 99%.
- [ ] If you always predict spam (output $y = 1$), your classifier will have a recall of 100% and precision of 1%.
- [x] If you always predict non-spam (output $y = 0$), your classifier will have an accuracy of 99%.

5. Which of the following statements are true? Check all that apply.

- [ ] The "error analysis" process of manually examining the examples which your algorithm got wrong can help suggest what are good steps to take (e.g., developing new features) to improve your algorithm's performance.
- [x] Using a **very large** training set makes it unlikely for model to overfit the training data.
- [x] It is a good idea to spend a lot of time collecting a **large** amount of data before building your first version of a learning algorithm.
- [ ] After training a logistic regression classifier, you **must** use 0.5 as your threshold for predicting whether an example is positive or negative.
- [ ] If your model is underfitting the training set, then obtaining more data is likely to help.

1. You are working on a spam classification system using regularized logistic regression. "Spam" is a positive class (y = 1) and "not spam" is the negative class (y = 0). You have trained your classifier and there are m = 1000 examples in the cross-validation set. The chart of predicted class vs. actual class is:

1 / 1点

	Actual Class: 1	**Actual Class: 0**
Predicted Class: 1	85	890
Predicted Class: 0	15	10

For reference:

- Accuracy = (true positives + true negatives) / (total examples)
- Precision = (true positives) / (true positives + false positives)
- Recall = (true positives) / (true positives + false negatives)
- F_1 score = (2 * precision * recall) / (precision + recall)

What is the classifier's accuracy (as a value from 0 to 1)?

Enter your answer in the box below. If necessary, provide at least two values after the decimal point.

0.095

正解
The classifier correctly predicted the true positives and the true negatives = 85 + 10, so the accuracy is 95/1000 = 0.095

2. Suppose a massive dataset is available for training a learning algorithm. Training on a lot of data is likely to give good performance when two of the following conditions hold true. 1 / 1点

Which are the two?

- [ ] We train a learning algorithm with a small number of parameters (that is thus unlikely to overfit).
- [x] We train a learning algorithm with a large number of parameters (that is able to learn/represent fairly complex functions).

正解
You should use a "low bias" algorithm with many parameters, as it will be able to make use of the large dataset provided. If the model has too few parameters, it will underfit the large training set.

- [x] The features x contain sufficient information to predict y accurately. (For example, one way to verify this is if a human expert on the domain can confidently predict y when given only x).

正解
It is important that the features contain sufficient information, as otherwise no amount of data can solve a learning problem in which the features do not contain enough information to make an accurate prediction.

- [ ] When we are willing to include high order polynomial features of x (such as x_1^2, x_2^2, x_1x_2, etc.).

3. Suppose you have trained a logistic regression classifier which is outputing $h_\theta(x)$. 1 / 1点

Currently, you predict 1 if $h_\theta(x) \geq \text{threshold}$, and predict 0 if $h_\theta(x) < \text{threshold}$, where currently the threshold is set to 0.5.

Suppose you **increase** the threshold to 0.9. Which of thefollowing are true? Check all that apply.

- [ ] The classifier is likely to have unchanged precision and recall, and

  thus the same F_1 score.

- [x] The classifier is likely to now have higher precision.

  正解
  Increasing the threshold means more y = 0 predictions. This will decrease both true and false positives, so precision will increase.

- [ ] The classifier is likely to have unchanged precision and recall, but

  higher accuracy.

- [ ] The classifier is likely to now have higher recall.

4. Suppose you are working on a spam classifier, where spam emails are positive examples ($y = 1$) and non-spam emails are negative examples ($y = 0$). You have a training set of emails in which 99% of the emails are non-spam and the other 1% is spam. Which of the following statements are true? Check all that apply.

0 / 1点

- [ ] If you always predict non-spam (output $y = 0$), your classifier will have a recall of 0%.
- [ ] If you always predict spam (output $y = 1$), your classifier will have a recall of 0% and precision of 99%.
- [ ] If you always predict spam (output $y = 1$), your classifier will have a recall of 100% and precision of 1%.
- [x] If you always predict non-spam (output $y = 0$), your classifier will have an accuracy of 99%.

正解
Since 99% of the examples are y = 0, always predicting 0 gives an accuracy of 99%. Note, however, that this is not a good spam system, as you will never catch any spam.

正しい回答をすべて選択しませんでした

5. Which of the following statements are true? Check all that apply. 0 / 1点

- [ ] The "error analysis" process of manually examining the examples which your algorithm got wrong can help suggest what are good steps to take (e.g., developing new features) to improve your algorithm's performance.
- [x] Using a **very large** training set makes it unlikely for model to overfit the training data.

正解
A sufficiently large training set will not be overfit, as the model cannot overfit some of the examples without doing poorly on the others.

- [x] It is a good idea to spend a lot of time collecting a **large** amount of data before building your first version of a learning algorithm.

これを選択しないでください
You cannot know whether a huge dataset will be important until you have built a first version and find that the algorithm has high variance.

- [ ] After training a logistic regression classifier, you **must** use 0.5 as your threshold for predicting whether an example is positive or negative.
- [ ] If your model is underfitting the training set, then obtaining more data is likely to help.

—

Take3

1. You are working on a spam classification system using regularized logistic regression. "Spam" is a positive class (y = 1) and "not spam" is the negative class (y = 0). You have trained your classifier and there are m = 1000 examples in the cross-validation set. The chart of predicted class vs. actual class is: 1 / 1点

	Actual Class: 1	**Actual Class: 0**
Predicted Class: 1	85	890
Predicted Class: 0	15	10

For reference:

- Accuracy = (true positives + true negatives) / (total examples)
- Precision = (true positives) / (true positives + false positives)
- Recall = (true positives) / (true positives + false negatives)
- F_1 score = (2 * precision * recall) / (precision + recall)

What is the classifier's recall (as a value from 0 to 1)?

Enter your answer in the box below. If necessary, provide at least two values after the decimal point.

0.85

正解
There are 85 true positives and 15 false negatives, so recall is 85 / (85 + 15) = 0.85.

2. Suppose a massive dataset is available for training a learning algorithm. Training on a lot of data is likely to give good performance when two of the following conditions hold true. **1 / 1点**

Which are the two?

- [ ] When we are willing to include high order polynomial features of x (such as x_1^2, x_2^2, $x_1 x_2$, etc.).
- [x] A human expert on the application domain can confidently predict y when given only the features x (or more generally, if we have some way to be confident that x contains sufficient information to predict y accurately).

正解
It is important that the features contain sufficient information, as otherwise no amount of data can solve a learning problem in which the features do not contain enough information to make an accurate prediction.

- [x] Our learning algorithm is able to represent fairly complex functions (for example, if we train a neural network or other model with a large number of parameters).

正解
You should use a complex, "low bias" algorithm, as it will be able to make use of the large dataset provided. If the model is too simple, it will underfit the large training set.

- [ ] The classes are not too skewed.

3. Suppose you have trained a logistic regression classifier which is outputing $h_\theta(x)$. 1 / 1点

Currently, you predict 1 if $h_\theta(x) \geq \text{threshold}$, and predict 0 if $h_\theta(x) lt \text{threshold}$, where currently the threshold is set to 0.5.

Suppose you **decrease** the threshold to 0.1. Which of the following are true? Check all that apply.

- [ ] The classifier is likely to now have higher precision.
- [ ] The classifier is likely to have unchanged precision and recall, but lower accuracy.
- [x] The classifier is likely to now have higher recall.

正解
Lowering the threshold means more y = 1 predictions. This will increase the number of true positives and decrease the number of false negatives, so recall will increase.

- [ ] The classifier is likely to have unchanged precision and recall, but higher accuracy.

4. Suppose you are working on a spam classifier, where spam emails are positive examples ($y = 1$) and non-spam emails are negative examples ($y = 0$). You have a training set of emails in which 99% of the emails are non-spam and the other 1% is spam. Which of the following statements are true? Check all that apply.

0 / 1

- [x] A good classifier should have both a high precision and high recall on the cross validation set.

  正解
  For data with skewed classes like these spam data, we want to achieve a high F_1 score, which requires high precision and high recall.

- [ ] If you always predict non-spam (output $y = 0$), your classifier will have 99% accuracy on the training set, and it will likely perform similarly on the cross validation set.

- [x] If you always predict non-spam (output $y = 0$), your classifier will have an accuracy of 99%.

  正解
  Since 99% of the examples are y = 0, always predicting 0 gives an accuracy of 99%. Note, however, that this is not a good spam system, as you will never catch any spam.

- [ ] If you always predict non-spam (output $y = 0$), your classifier will have 99% accuracy on the training set, but it will do much worse on the cross validation set because it has overfit the training data.

正しい回答をすべて選択しませんでした

5. Which of the following statements are true? Check all that apply.

- [ ] If your model is underfitting the training set, then obtaining more data is likely to help.
- [ ] On skewed datasets (e.g., when there are more positive examples than negative examples), accuracy is not a good measure of performance and you should instead use F_1 score based on the precision and recall.
- [ ] After training a logistic regression classifier, you **must** use 0.5 as your threshold for predicting whether an example is positive or negative.
- [x] It is a good idea to spend a lot of time collecting a **large** amount of data before building your first version of a learning algorithm.

⊗ **これを選択しないでください**
You cannot know whether a huge dataset will be important until you have built a first version and find that the algorithm has high variance.

- [x] Using a **very large** training set makes it unlikely for model to overfit the training data.

✓ **正解**
A sufficiently large training set will not be overfit, as the model cannot overfit some of the examples without doing poorly on the others.

Take4

4. Suppose you are working on a spam classifier, where spam emails are positive examples ($y = 1$) and non-spam emails are negative examples ($y = 0$). You have a training set of emails in which 99% of the emails are non-spam and the other 1% is spam. Which of the following statements are true? Check all that apply.

0 /

- [ ] If you always predict non-spam (output $y = 0$), your classifier will have 99% accuracy on the training set, and it will likely perform similarly on the cross validation set.

- [x] If you always predict non-spam (output $y = 0$), your classifier will have an accuracy of 99%.

正解
Since 99% of the examples are y = 0, always predicting 0 gives an accuracy of 99%. Note, however, that this is not a good spam system, as you will never catch any spam.

- [x] If you always predict non-spam (output $y = 0$), your classifier will have 99% accuracy on the training set, but it will do much worse on the cross validation set because it has overfit the training data.

これを選択しないでください
The classifier achieves 99% accuracy because of the skewed classes in the data, not because it is overfitting the training set. Thus, it is likely to perform just as well on the cross validation set.

- [x] A good classifier should have both a high precision and high recall on the cross validation set.

正解
For data with skewed classes like these spam data, we want to achieve a high F_1 score, which requires high precision and high recall.

5. Which of the following statements are true? Check all that apply. 1 / 1点

- [ ] If your model is underfitting the training set, then obtaining more data is likely to help.
- [ ] After training a logistic regression classifier, you **must** use 0.5 as your threshold for predicting whether an example is positive or negative.
- [x] Using a **very large** training set makes it unlikely for model to overfit the training data.

正解
A sufficiently large training set will not be overfit, as the model cannot overfit some of the examples without doing poorly on the others.

- [ ] It is a good idea to spend a lot of time collecting a **large** amount of data before building your first version of a learning algorithm.
- [x] The "error analysis" process of manually examining the examples which your algorithm got wrong can help suggest what are good steps to take (e.g., developing new features) to improve your algorithm's performance.

正解
This process of error analysis is crucial in developing high performance learning systems, as the space of possible improvements to your system is very large, and it gives you direction about what to work on next.