Advice for applying machine learning

Deciding what to try next
Suppose you have implemented regularized linear regression to predict housing prices.

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

However, when you test your hypothesis on a new set of houses, you find that it makes unacceptably large errors in its predictions. What should you try next?

- Get more training examples
- Try smaller sets of features
- Try getting additional features
- Try adding polynomial features \((x_1^2, x_2^2, x_1x_2, \text{etc.})\)
- Try decreasing \(\lambda\)
- Try increasing \(\lambda\)
Machine learning diagnostic:

Diagnostic: A test that you can run to gain insight what is/isn’t working with a learning algorithm, and gain guidance as to how best to improve its performance.

Diagnostics can take time to implement, but doing so can be a very good use of your time.
Advice for applying machine learning

Evaluating a hypothesis
Evaluating your hypothesis

Fails to generalize to new examples not in training set.

\[ h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4 \]

\[
\begin{align*}
x_1 & = \text{size of house} \\
x_2 & = \text{no. of bedrooms} \\
x_3 & = \text{no. of floors} \\
x_4 & = \text{age of house} \\
x_5 & = \text{average income in neighborhood} \\
x_6 & = \text{kitchen size} \\
\vdots \\
x_{100} & \\
\end{align*}
\]
Evaluating your hypothesis

Dataset:

<table>
<thead>
<tr>
<th>Size</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>2104</td>
<td>400</td>
</tr>
<tr>
<td>1600</td>
<td>330</td>
</tr>
<tr>
<td>2400</td>
<td>369</td>
</tr>
<tr>
<td>1416</td>
<td>232</td>
</tr>
<tr>
<td>3000</td>
<td>540</td>
</tr>
<tr>
<td>1985</td>
<td>300</td>
</tr>
<tr>
<td>1534</td>
<td>315</td>
</tr>
<tr>
<td>1427</td>
<td>199</td>
</tr>
<tr>
<td>1380</td>
<td>212</td>
</tr>
<tr>
<td>1494</td>
<td>243</td>
</tr>
</tbody>
</table>

20% \(\rightarrow\) Training set

30% \(\rightarrow\) Test set

\(\begin{align*}
(x^{(1)}, y^{(1)}) \\
(x^{(2)}, y^{(2)}) \\
&\vdots \\
(x^{(m)}, y^{(m)})
\end{align*}\)

\(m_{\text{test}} = \text{no. of test examples} (x_{\text{test}}^{(i)}, y_{\text{test}}^{(i)})\)
Training/testing procedure for linear regression

- Learn parameter $\theta$ from training data (minimizing $J(\theta)$)
  - $\approx 70\%$

- Compute test set error:

$$J_{\text{test}}(\theta) = \frac{1}{2m_{\text{test}}} \sum_{i=1}^{m_{\text{test}}} (h_\theta(x^{(i)\text{test}}) - y^{(i)\text{test}})^2$$
Training/testing procedure for logistic regression

- Learn parameter $\theta$ from training data
- Compute test set error:

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

- Misclassification error (0/1 misclassification error):
Advice for applying machine learning

Model selection and training/validation/test sets
Once parameters $\theta_0, \theta_1, \ldots, \theta_4$ were fit to some set of data (training set), the error of the parameters as measured on that data (the training error $J(\theta)$) is likely to be lower than the actual generalization error.
Model selection

1. $h_{\theta}(x) = \theta_0 + \theta_1 x$
2. $h_{\theta}(x) = \theta_0 + \theta_1 x + \theta_2 x^2$
3. $h_{\theta}(x) = \theta_0 + \theta_1 x + \cdots + \theta_3 x^3$

\[ \vdots \]
10. $h_{\theta}(x) = \theta_0 + \theta_1 x + \cdots + \theta_{10} x^{10}$

\[ \begin{array}{c}
\theta_0 + \ldots \theta_5 x^5 \\
\end{array} \]

Choose $\theta_0 + \ldots \theta_5 x^5$

How well does the model generalize? Report test set error $J_{test}(\theta^{(5)})$.

Problem: $J_{test}(\theta^{(5)})$ is likely to be an optimistic estimate of generalization error. I.e. our extra parameter ($d = \text{degree of polynomial}$) is fit to test set.
Evaluating your hypothesis

Dataset:

<table>
<thead>
<tr>
<th>Size</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>2104</td>
<td>400</td>
</tr>
<tr>
<td>1600</td>
<td>330</td>
</tr>
<tr>
<td>2400</td>
<td>369</td>
</tr>
<tr>
<td>1416</td>
<td>232</td>
</tr>
<tr>
<td>3000</td>
<td>540</td>
</tr>
<tr>
<td>1985</td>
<td>300</td>
</tr>
<tr>
<td>1534</td>
<td>315</td>
</tr>
<tr>
<td>1427</td>
<td>199</td>
</tr>
<tr>
<td>1380</td>
<td>212</td>
</tr>
<tr>
<td>1494</td>
<td>243</td>
</tr>
</tbody>
</table>

Dataset split:
- **60%** for training set
- **20%** for cross-validation set
- **20%** for test set

Mathematical representation:
\[
(x^{(1)}, y^{(1)}) \quad (x^{(2)}, y^{(2)}) \quad \ldots \quad (x^{(m)}, y^{(m)})
\]
\[
(x_{cv}^{(1)}, y_{cv}^{(1)}) \quad (x_{cv}^{(2)}, y_{cv}^{(2)}) \quad \ldots \quad (x_{cv}^{(m_{cv})}, y_{cv}^{(m_{cv})})
\]
\[
(x_{test}^{(1)}, y_{test}^{(1)}) \quad (x_{test}^{(2)}, y_{test}^{(2)}) \quad \ldots \quad (x_{test}^{(m_{test})}, y_{test}^{(m_{test})})
\]

\[M_{cv} = \text{no. of CV examples} (x_{cv}, y_{cv})\]

\[M_{test} = \text{no. of test examples} (x_{test}, y_{test})\]
Train/validation/test error

Training error:

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

Cross Validation error:

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

Test error:

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

1. \( h_\theta(x) = \theta_0 + \theta_1 x \)
2. \( h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 \)
3. \( h_\theta(x) = \theta_0 + \theta_1 x + \cdots + \theta_3 x^3 \)

\vdots

10. \( h_\theta(x) = \theta_0 + \theta_1 x + \cdots + \theta_{10} x^{10} \)

Pick \( \theta_0 + \theta_1 x_1 + \cdots + \theta_4 x^4 \)

Estimate generalization error for test set \( J_{test}(\theta^{(4)}) \)
Advice for applying machine learning

Diagnosing bias vs. variance
Bias/variance

- High bias (underfit) \( d=1 \)
- "Just right" \( d=2 \)
- High variance (overfit) \( d=4 \)
Bias/variance

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

Cross validation error: \( J_{\text{cv}}(\theta) = \frac{1}{2m_{\text{cv}}} \sum_{i=1}^{m_{\text{cv}}} (h_{\theta}(x_{\text{cv}}^{(i)}) - y_{\text{cv}}^{(i)})^2 \)

(or \( J_{\text{tot}}(\theta) \))
Diagnosing bias vs. variance

Suppose your learning algorithm is performing less well than you were hoping. \( J_{cv}(\theta) \) or \( J_{test}(\theta) \) is high.) Is it a bias problem or a variance problem?

Bias (underfit): \[ J_{train}(\theta) \text{ will be high} \]

\[ J_{cv}(\theta) \approx J_{train}(\theta) \]

Variance (overfit): \[ J_{train}(\theta) \text{ will be low} \]

\[ J_{cv}(\theta) \gg J_{train}(\theta) \]
Advice for applying machine learning

Regularization and bias/variance
Linear regression with regularization

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

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

- Large \( \lambda \) \( \Rightarrow \) High bias (underfit)
  \[ \lambda = 10000, \ \theta_1 \approx 0, \ \theta_2 \approx 0, \ldots \]
  \[ h_\theta(x) \approx \theta_0 \]

- Intermediate \( \lambda \) \( \Rightarrow \) "Just right"

- Small \( \lambda \) \( \Rightarrow \) High variance (overfit)
  \[ \lambda = 0 \]
Choosing the regularization parameter $\lambda$

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

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

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

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

$$J_{\text{test}}(\theta) = \frac{1}{2m_{\text{test}}} \sum_{i=1}^{m_{\text{test}}} (h_\theta(x_{\text{test}}^{(i)}) - y_{\text{test}}^{(i)})^2$$
Choosing the regularization parameter $\lambda$

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

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

1. Try $\lambda = 0 \leftarrow \min J(\theta) \rightarrow \Theta^{(1)} \rightarrow J_{\text{cv}}(\Theta^{(1)})$
2. Try $\lambda = 0.01 \rightarrow \Theta^{(2)} \rightarrow J_{\text{cv}}(\Theta^{(2)})$
3. Try $\lambda = 0.02$
4. Try $\lambda = 0.04$
5. Try $\lambda = 0.08$

$\vdots$

12. Try $\lambda = 10 \leftarrow \Theta^{(12)} \rightarrow J_{\text{cv}}(\Theta^{(12)})$

Pick (say) $\Theta^{(5)}$. Test error: $J_{\text{test}}(\Theta^{(5)})$

Andrew Ng
Bias/variance as a function of the regularization parameter $\lambda$

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

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

$$J_{\text{train}}(\theta)$$
Advice for applying machine learning

Learning curves

Machine Learning
Learning curves

\[ J_{\text{train}}(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 \]

\[ J_{\text{cv}}(\theta) = \frac{1}{2m_{\text{cv}}} \sum_{i=1}^{m_{\text{cv}}} (h_\theta(x^{(i)}_{\text{cv}}) - y^{(i)}_{\text{cv}})^2 \]

\[ h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 \]
If a learning algorithm is suffering from high bias, getting more training data will not (by itself) help much.
High variance

If a learning algorithm is suffering from high variance, getting more training data is likely to help.
Advice for applying machine learning

Deciding what to try next (revisited)
Debugging a learning algorithm:

Suppose you have implemented regularized linear regression to predict housing prices. However, when you test your hypothesis in a new set of houses, you find that it makes unacceptably large errors in its prediction. What should you try 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 \((x_1^2, x_2^2, x_1 x_2, \text{etc})\) → fix high bias
- Try decreasing \(\lambda\) → fix high bias
- Try increasing \(\lambda\) → fix high variance
Neural networks and overfitting

“Small” neural network (fewer parameters; more prone to underfitting)

Computationally cheaper

“Large” neural network (more parameters; more prone to overfitting)

Computationally more expensive.

Use regularization ($\lambda$) to address overfitting.