情報工学実験4: データマイニング班
(week 6) 機械学習概観の振り返り

1. (復習) 用語
2. (復習) 機械学習における問題設定
3. (復習) scikit-learnの使い方
4. (復習) モデルと仮説
5. (復習) 問題・アルゴリズム・モデル
6. (復習) 過学習
7. (復習) ペナルティ項（正則化）
8. (復習) 交差検定

実験ページ: http://ie.u-ryukyu.ac.jp/~tnal/2016/info4/dm/
## Terminology

<table>
<thead>
<tr>
<th>Supervised, Unsupervised Learning</th>
<th>Input, Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classification, Regression, Clustering</td>
<td>Training Data / Training Set</td>
</tr>
<tr>
<td>Sample</td>
<td>Test Data / Test Set</td>
</tr>
<tr>
<td>Features, Attributes</td>
<td>- Open Test</td>
</tr>
<tr>
<td>- Numerical Value</td>
<td>- Close Test</td>
</tr>
<tr>
<td>- Categorical Value</td>
<td>Model</td>
</tr>
<tr>
<td>- True or False</td>
<td>Parameters</td>
</tr>
<tr>
<td>Supervisory Signal, Teacher, Class, Label, Output Data, Target Variable</td>
<td>Predict, Estimate</td>
</tr>
<tr>
<td></td>
<td>Evaluation</td>
</tr>
</tbody>
</table>
Machine Learning: the problem setting


• In general, a learning problem considers a set of n samples of data and then tries to predict properties of unknown data. If each sample is more than a single number and, for instance, a multi-dimensional entry (aka multivariate data), is it said to have several attributes or features.
An introduction to machine learning with scikit-learn (1/3)

hg clone ssh://info3dm@shark//home/info3dm/HG/tnal less sklearn_intro.py

• Loading and an example dataset
  – workon py34
  – (py34) python
  – >>> from sklearn import datasets
  – >>> iris = datasets.load_iris() # datasets.load[tab]
  – >>> print(iris.DESCR)
  – >>> print(iris.data)
  – >>> print(iris.target)
  – >>> print(iris.target_names)

An introduction to machine learning with scikit-learn (2/3)

• Learning and predicting
  – >>> from sklearn import svm
  – >>> clf = svm.SVC(gamma=0.001, C=100.)
  – >>> clf.fit(iris.data[:-1], iris.target[:-1])
  – >>> clf.predict(iris.data[-1])
  – >>> print(iris.target[-1])
  – >>> clf.score(iris.data, iris.target)

Models

- Represent by any formulas with (sometimes one) parameters for the relationship between input X’s and output Y’s.
  - In machine learning, the formulas called as "hypothesis".
  - E.g., $h = ax + b$
    - $a, b$: parameters
  - Parameterized model.
  - Predictive model. (e.g., $a=1$, $b=2$)
Underfitting vs. Overfitting

This example demonstrates the problems of underfitting and overfitting and how we can use linear regression with polynomial features to approximate nonlinear functions. The plot shows the function that we want to approximate, which is a part of the cosine function. In addition, the samples from the real function and the approximations of different models are displayed. The models have polynomial features of different degrees. We can see that a linear function (polynomial with degree 1) is not sufficient to fit the training samples. This is called **underfitting**. A polynomial of degree 4 approximates the true function almost perfectly. However, for higher degrees the model will **overfit** the training data, i.e. it learns the noise of the training data.

a penalty for the parameters (generalization)

• introduce a penalty term to cost function $J(\theta)$.
  – sum of squared parameters (L2-norm)

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

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

  $\|\theta\|^2 = \theta_0^2 + \theta_1^2 + \ldots + \theta_M^2$
K-fold cross-validation (K=5)

(step 1) divides all the samples in K groups (folds) of equal sizes, if possible.

(step 2) The model.fit() is learned using K-1 folds (4 folds), and the fold left out is used for test.