1 Logistic Regression

In [6]:

```
#code source: http://occam.olin.edu/sites/default/files/DataScienceMaterials/machine_lear
from sklearn.model_selection import train_test_split
from sklearn.grid_search import GridSearchCV
from sklearn.datasets import *
from sklearn.linear_model import LogisticRegression

data = load_breast_cancer() #refer: http://scikit-learn.org/stable/modules/generated/skle

tuned_parameters = [{'C': [10**-4, 10**-2, 10**0, 10**2, 10**4]}]
X_train, X_test, y_train, y_test = train_test_split(data.data, data.target, train_size=.9


#Using GridSearchCV
model = GridSearchCV(LogisticRegression(), tuned_parameters, scoring = 'f1', cv=5)
model.fit(X_train, y_train)

print(model.best_estimator_)
print(model.score(X_test, y_test))
```

```
C:\Users\DELL\Anaconda3\lib\site-packages\sklearn\model_selection\_split.p
y:2026: FutureWarning: From version 0.21, test_size will always complement
train_size unless both are specified.
  FutureWarning)

LogisticRegression(C=10000, class_weight=None, dual=False, fit_intercept=T
rue,
          intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
          penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
          verbose=0, warm_start=False)
0.9811320754716981
```

In [7]:

```
# More Sparsity (Fewer elements of W* being non-zero) by increasing Lambda (decreasing C)

import numpy as np

clf = LogisticRegression(C=0.1, penalty='l1');
clf.fit(X_train, y_train);
w = clf.coef_
print(np.count_nonzero(w))
```

```
8
```

In [8]:

```
clf = LogisticRegression(C=0.01, penalty='l1');
clf.fit(X_train, y_train);
w = clf.coef_
print(np.count_nonzero(w))
```

```
4
```

In [9]:

```
clf = LogisticRegression(C=0.001, penalty='l1');
clf.fit(X_train, y_train);
w = clf.coef_
print(np.count_nonzero(w))
```

```
3
```

In [10]:

```
clf = LogisticRegression(C=10, penalty='l1');
clf.fit(X_train, y_train);
w = clf.coef_
print(np.count_nonzero(w))
```

```
18
```

In [0]:

```

```