# Explain how SVM algorithms works and explain why the result comes out (explain the comparing result of onebyone function and onebyall function)

## SVM 동작 원리

SVM에서 두 클래스에 속하는 점들을 나누는 경계인 hyperplane *H*는 다음과 같이 정의할 수 있다.

여기서 *w*와 w0은 weight이며, *x*는 m차원의 데이터를 말한다.

분류기의 안정성을 위해서는 hyperplane과 support vector간의 거리 (margin)을 최대화해야 하는데, 그러한 hyperplane을 구하기 위해서는 다음 최적화 문제를 풀어야 한다.

는 i번째 데이터를 말한다.

와 사이의 거리 (margin)는 로 계산할 수 있으므로, 를 최소화하면 margin을 최대화할 수 있다. 제약 조건인 은 클래스 레이블 가 1인 데이터들은 hyperplane의 위에 위치하고, 가 -1인 데이터들은 hyperplane의 아래에 위치하도록 만들기 위해 필요하다.

## onebyone VS. onebyall

***onebyone*** 함수는 hyperplane을 계산할 때 총 세 종류의 클래스 중 두 클래스에 속하는 데이터들만을 고려하고, 나머지 한 클래스에 속하는 데이터는 없는 걸로 간주한다. 때문에 결과물로 나온 그래프를 보면 hyperplane이 고려되지 않는 클래스에 속하는 데이터들을 가로지르는 것을 확인할 수 있다.

반면 ***onebyall*** 함수는 선택된 클래스에 속하는 데이터들을 클래스 1, 그렇지 않은 나머지 데이터들은 클래스 2에 속하는 걸로 간주하고 hyperplane을 그린다. 그 결과로 나온 그래프에서는 hyperplane이 어떠한 데이터도 가로지르지 않는 것을 확인할 수 있다.

# Suppose you want to classify an instance into one of N classes. How can you build a single multi-class classifier based on your binary discriminants?

onebyall 함수에서 취한 접근법을 사용하면 된다. 한 클래스와 나머지 모든 클래스를 구분 짓는 hyperplane을 그리는 binary SVM 분류기를 N개 훈련시킨 후, 그 분류기들을 이용해서 N개의 hyperplane을 그리면 된다.

# Graphs that you can get when you run the svmTest.m script

## onebyone

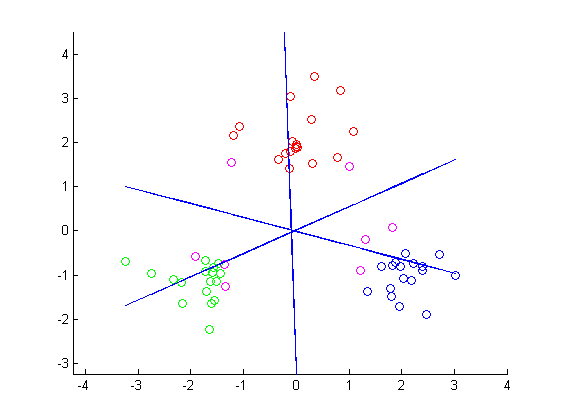


Figure onebyone

## onebyall

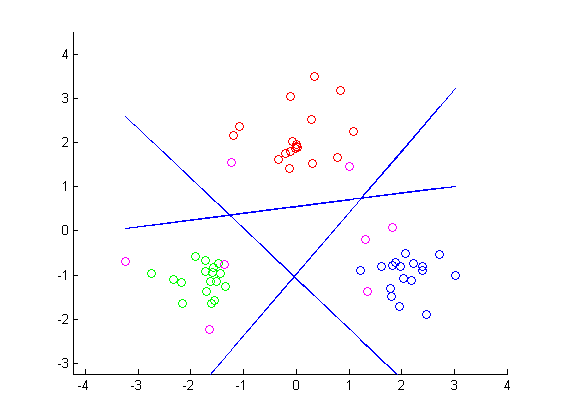


Figure onebyall