Limitations of traditional Methods

- Some of the two-class problems may fall into a load imbalance situation because the size of each class may be very imbalance in some problems. (eg. Forest CoverType).
- Using the one-versus-one, some of the two-class problems may still be too large to learn.

Min-Max Modular SVM

- Dividing a K-class problem into K(K-1)/2 two-class problems.
- These two-class problems can be further be decomposed into a number of relatively smaller and simplifier subproblems.
- These subproblems are independent from each other in learning phase, so they can be easily trained in a parallel way.

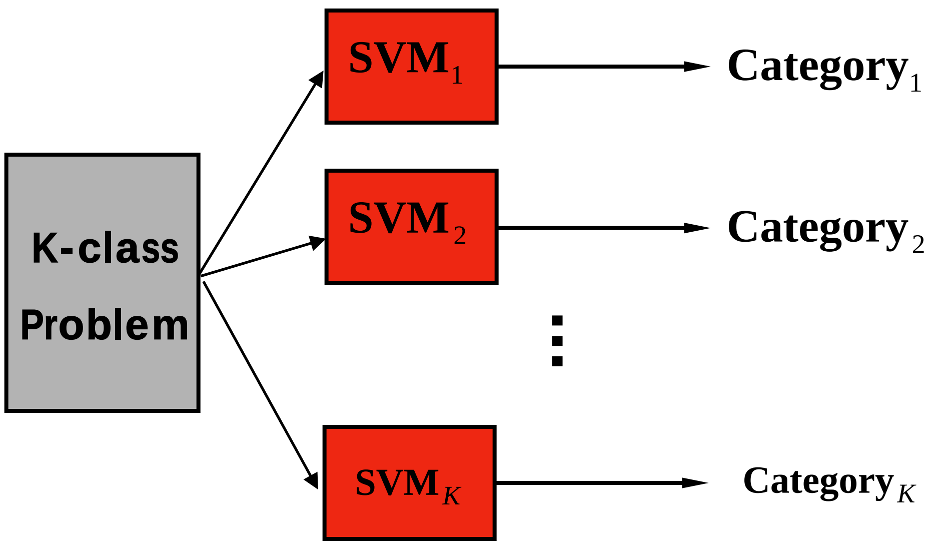
SVMs for Multi-class Classification Problems

Three task decomposition methods:

- One-versus-rest
- One-versus-one
- Part-versus-part

One-Versus-Rest method

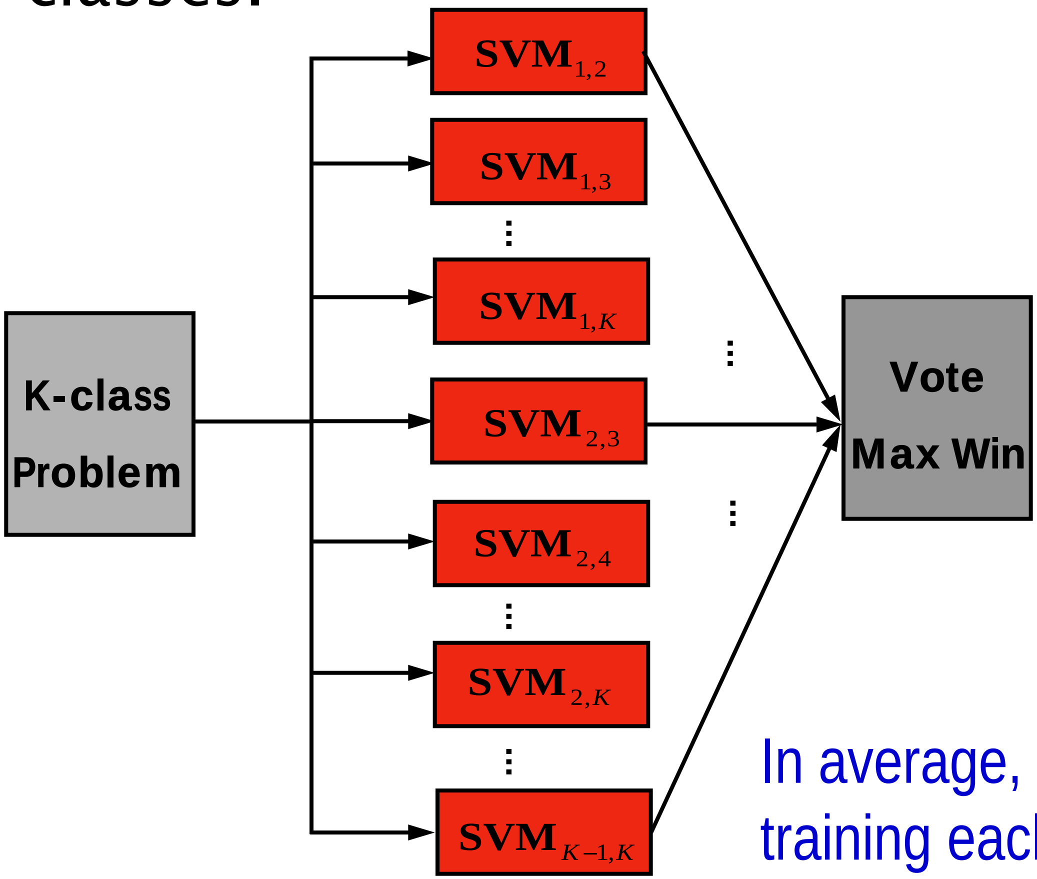
- This method requires one classifier per category. The ith SVM will be trained with all of the examples in the ith class with positive labels, and all other examples with negative labels.



The Number of training data for each classifier is <u>*N*</u>

One–Versus–One Method

- This method constructs K(K–1)/2 classifiers where each one is trained on data from two out of K classes.



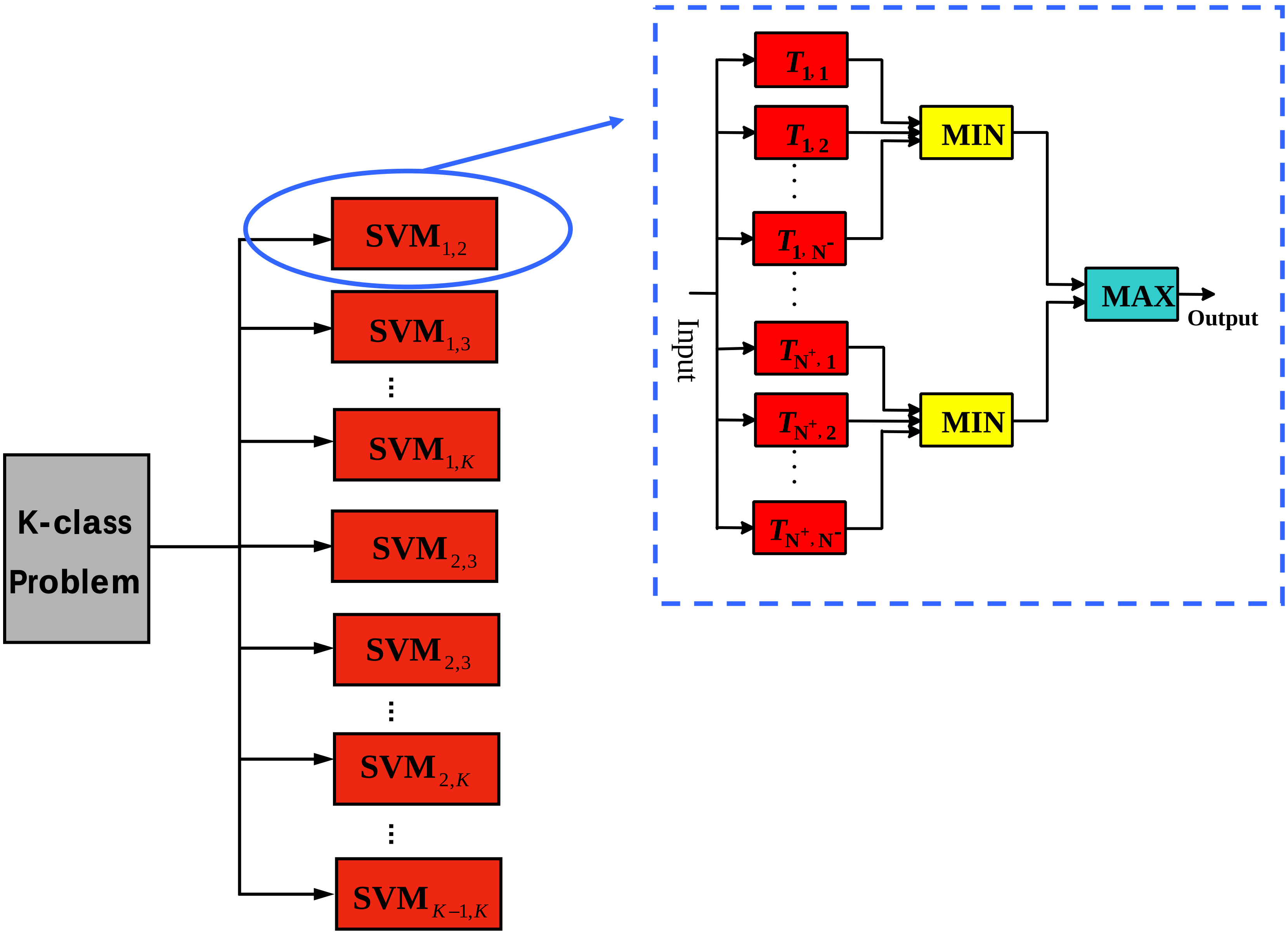
In average, number of data for training each classifier is <u>*2N/K*</u>

Limitations of traditional Methods

- Some of the two-class problems may fall into a load imbalance situation for the size of each class may be very imbalance in some problems.
- Using the one-versus-one, some of the two-class problems may still be too large to learn.

Part-versus-part

- Part-vs-part: Any two-class problem can be further decomposed into a number of two-class sub-problems as small as needed.

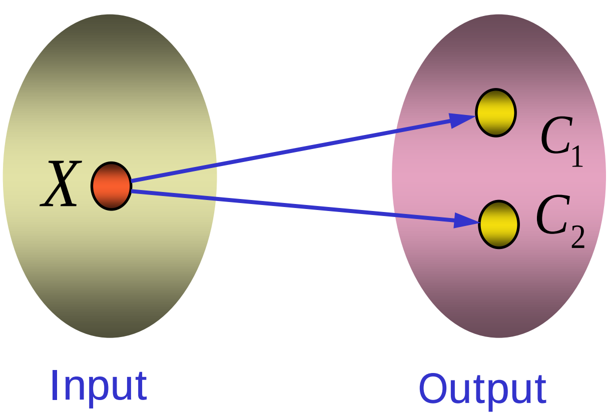

K-class Problem
$SVM_{1,2}$
$SVM_{1,3}$
$SVM_{1,K}$
$SVM_{2,3}$
$SVM_{2,3}$
$SVM_{2,K}$
$SVM_{K-1,K}$
Input
$T_{1,1}$
$T_{1,2}$
T_{1,N^-}
$T_{N^+,1}$
$T_{N^+,2}$
T_{N^+,N^-}
MIN
MIN
MAX
Output

Advantages of part-versus-part method

- A large-scale two-class problem can be divided into a number of relatively smaller two-class problems
- A serious imbalance two-class problem can be divided into a number of balance two-class problems
- Massively parallel learning can be easily implemented

What is a multi-label problem ?

- For a given training input *x*, there are *n* (*n*>1) labels, *yi* (*i*=*1*,…,*n*), corresponding to the training input *x*
- Multi-label problems can not be directly solved by using conventional learning frameworks because a one-to-many mapping should be created



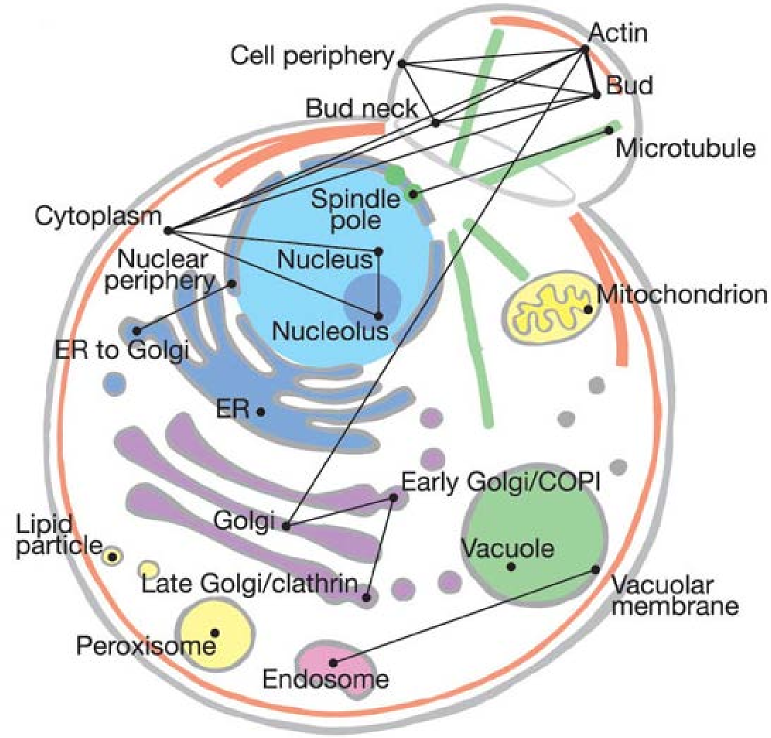
Multi-label problems DO Exit !

- Text categorization

  There are 1.7 labels for each document in average at Yomiuri News corpus

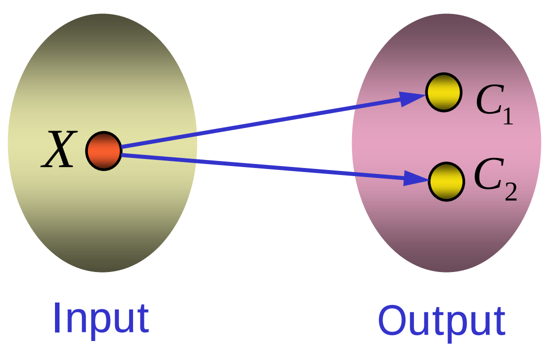
- Subcellular localization of protein subsequence

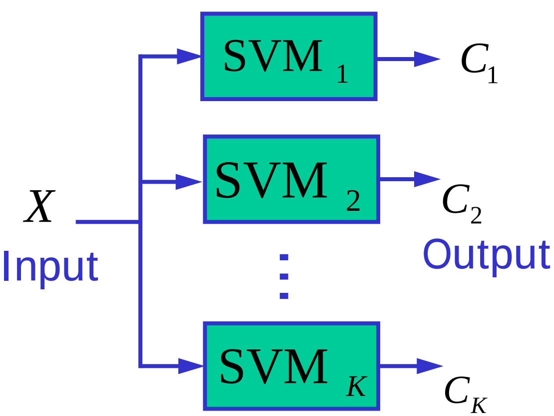
  One protein sequence has at most 5 locations in budding yeast



Existing Method for Multi-Label Problem

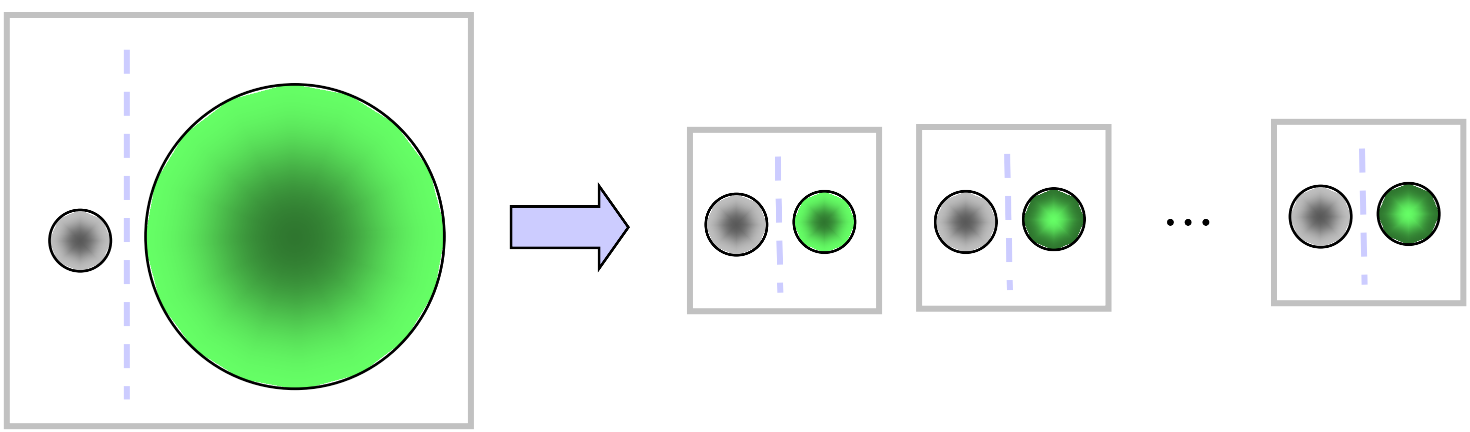
- Divide a K-class multi-label problem into K two-class problems using one-versus-rest method.
- Shortcoming: each of the two class problems will be a serious imbalance and large-scale one.





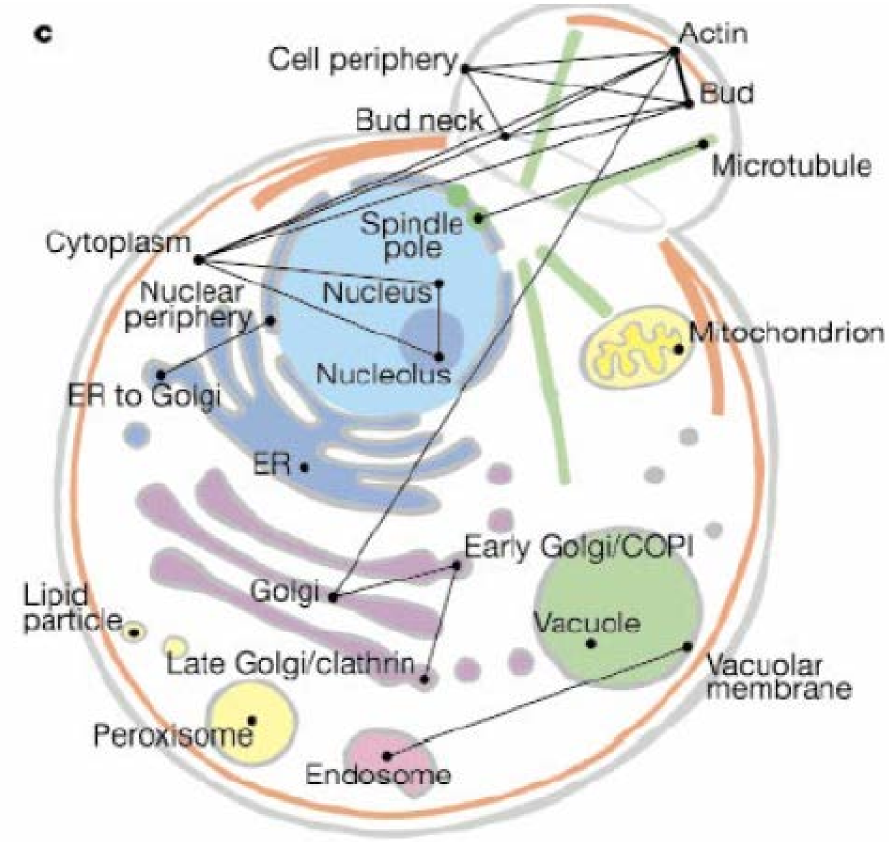
Part–versus–part method

- Divide a K–class multi–label problem into K two–class problems using one–versus–rest method
- Divide each of the imbalance or large–scale two–class problems into a number of relatively more balance and smaller two–class subproblems.



Protein Subcellular Localization

- The function of a protein is closely correlated with its subcellular location.
- Since more and more protein sequences enter into public database, extracting the sequence information for predicting protein subcellular location becomes very important.
- Multi-location problem: One protein sequence has at most 5 locations in yeast cells.



Data set

3555 proteins in budding yeast in 22 subcellular locations (Y. D. Cai and K. C. Chou, BBRC, 2004), taken from the experimental classification results by Huh et al. [17] at http://yeastgfp.ucsf.edu

Subcellular location	Number of proteins
Actin	29
Bud	23
Bud neck	60
Cell periphery	106
Cytoplasm	1576
Early Golgi	51
Endosome	43
ER	272
ER to Golgi	6
Golgi	40
Late Golgi	37
Lipid particle	19
Microtubule	20
Mitochondrion	494
Nuclear periphery	59
Nucleolus	157
Nucleus	1333
Peroxisome	20
Punctate composite	123
Spindle pole	58
Vacuolar membrane	54
vacuole	129

Experimental Result

- 22-label classification Problem
  - One-vs-rest
  - Build 22 SVM classifiers corresponding to 22 subcellular locations.
- Divide big class to smaller parts
  - Module =1000
- 10-fold cross-validation

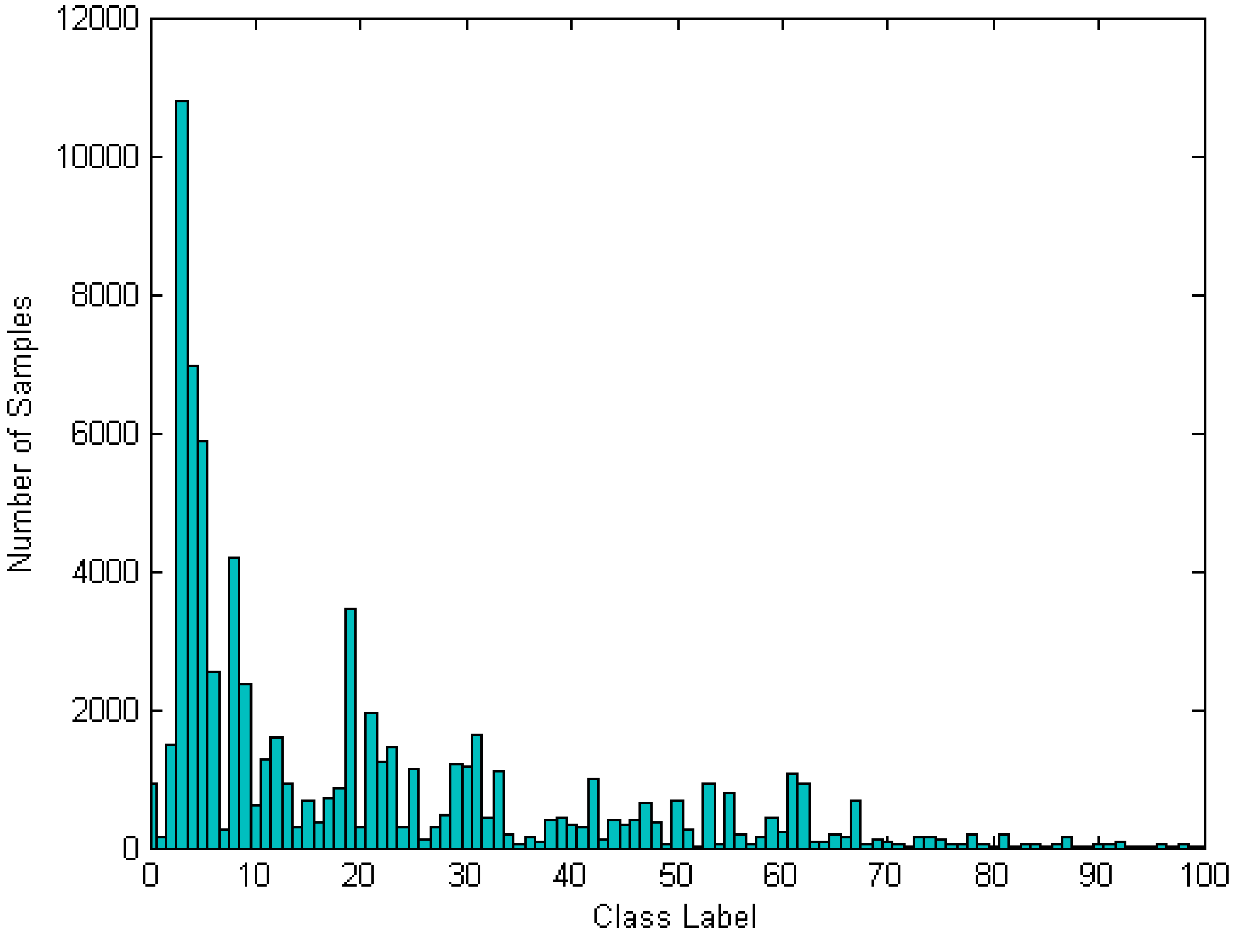
Comparison of Classification accuracy

Measure	Min-max modular SVM	Traditional SVM
Total Accuracy (%)	73.24	45.95
Location Accuracy (%)	34.55	16.66

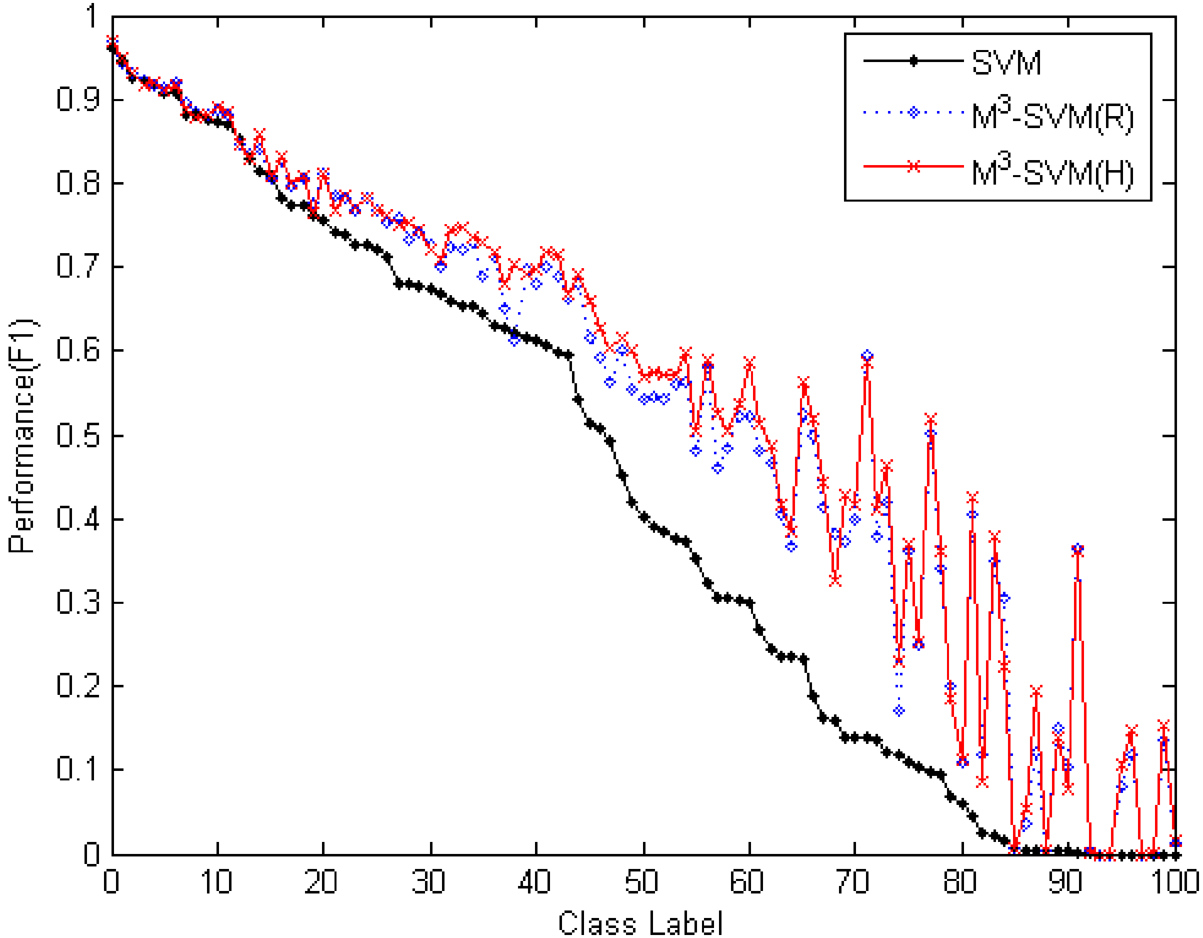
Text Categorization

(F. Y. Liu & B. L. Lu, 2005)

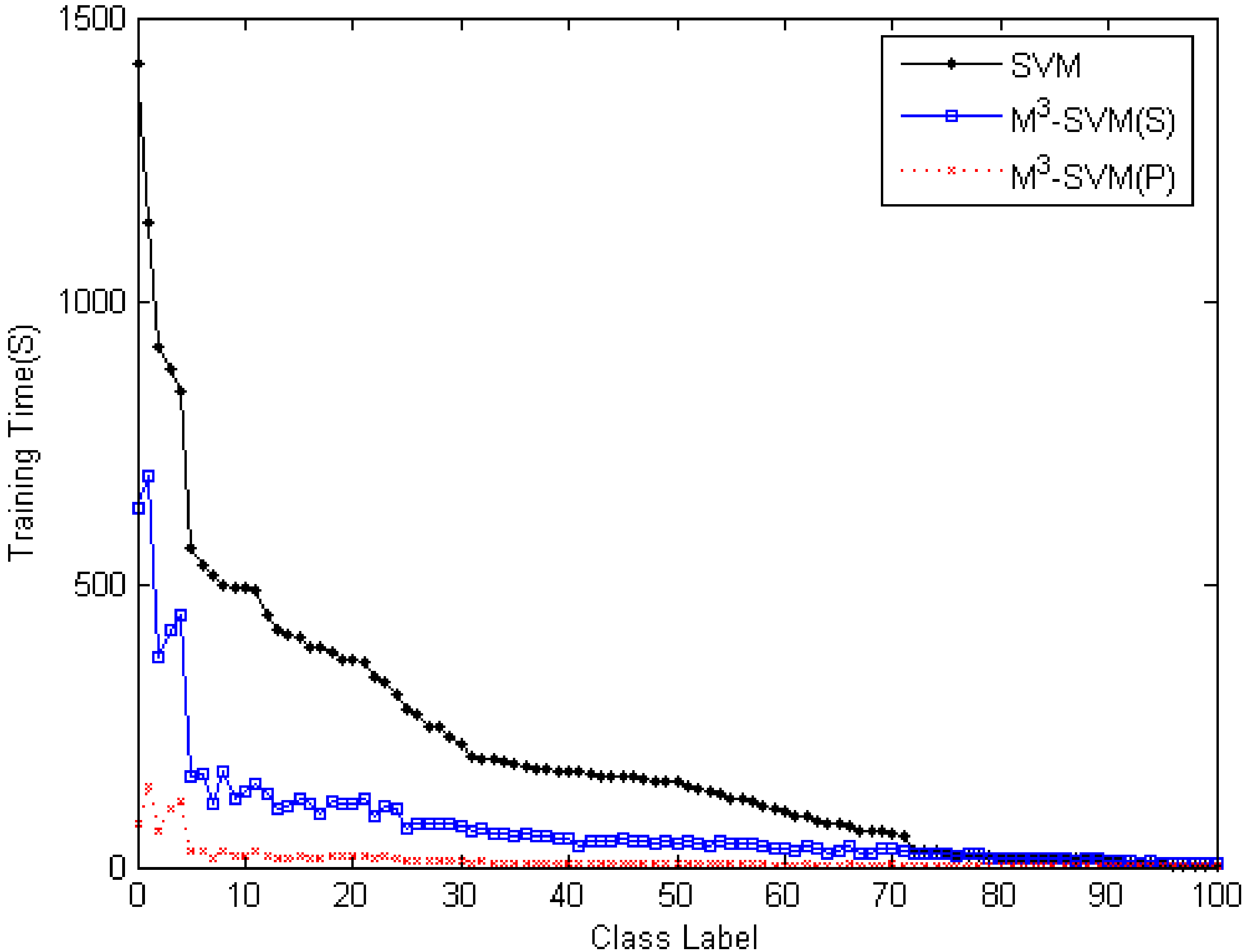
RCV1–V2: Data Distribution



RCV1–V2: Generalization Performance



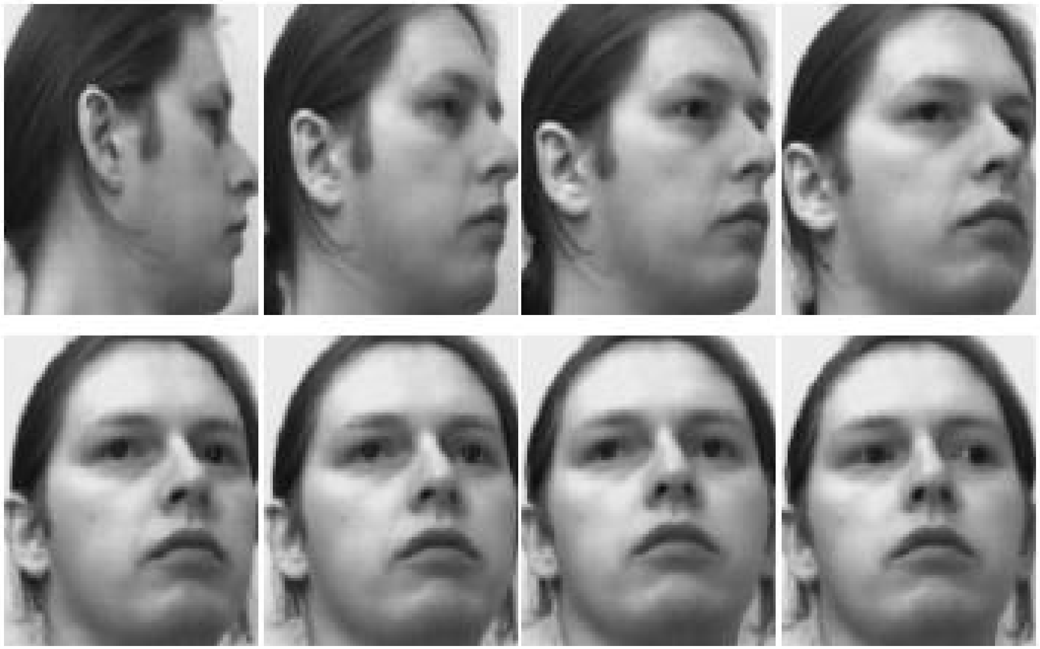
RCV1−V2: Training Times

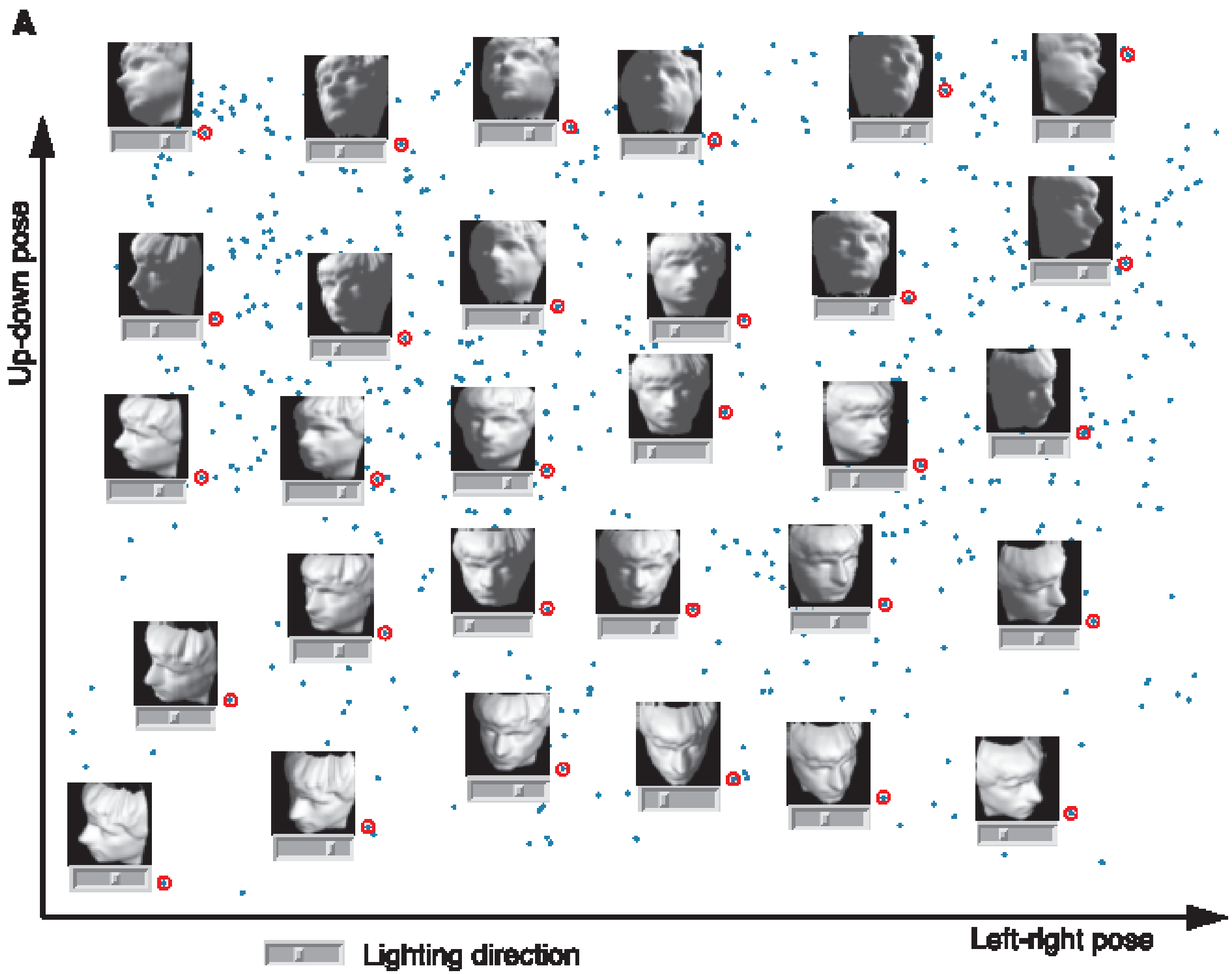


Gender Recognition

(H. C. Lian & B. L. Lu, 2005)

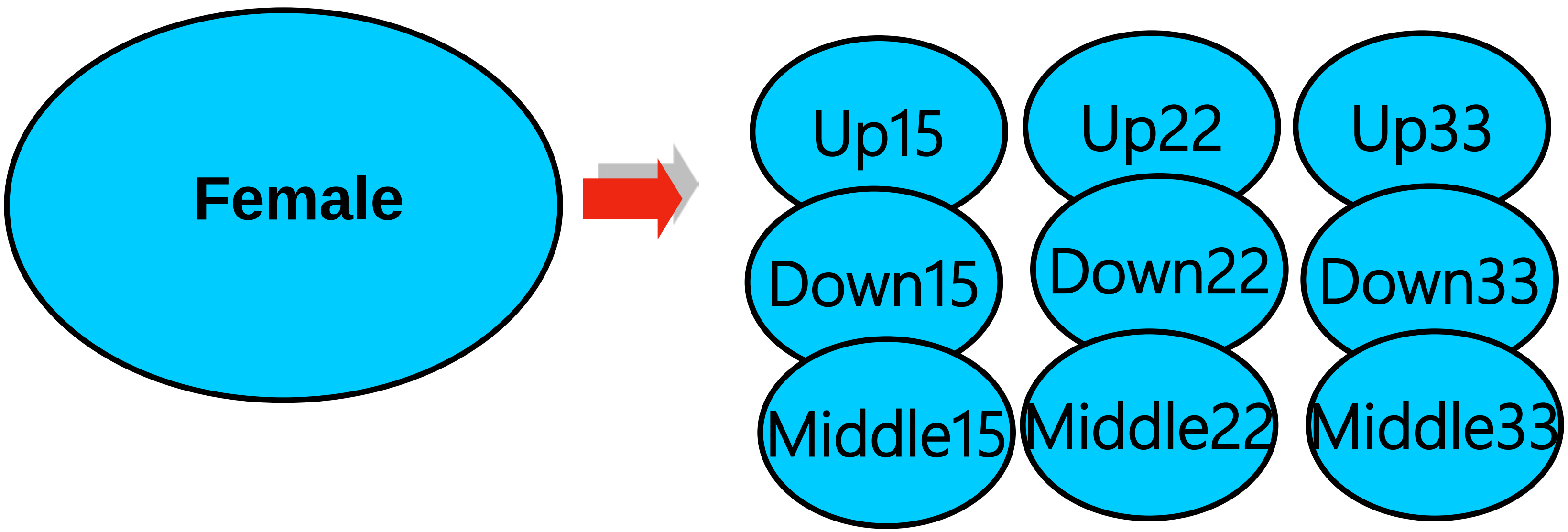
Multi-view Face Recognition

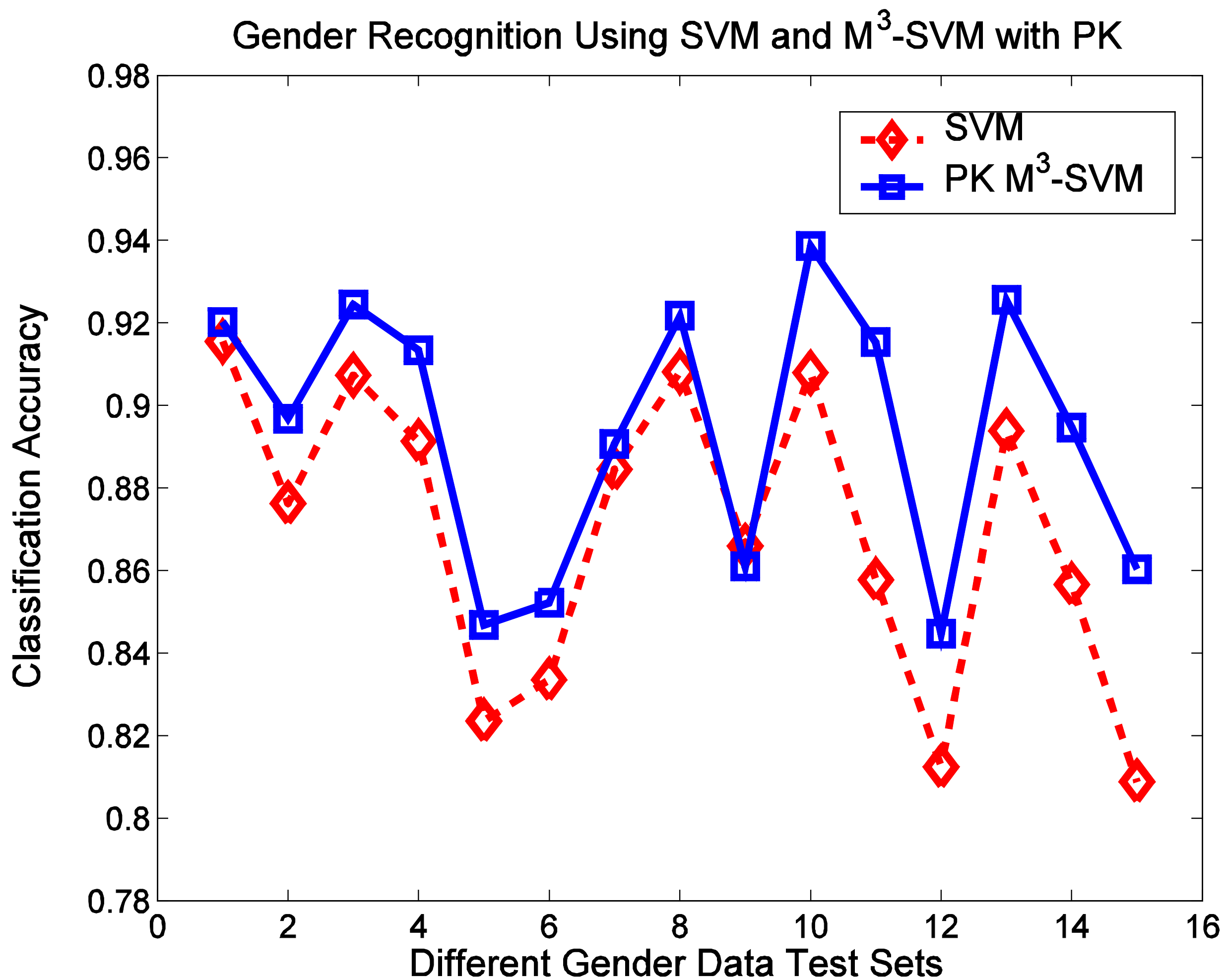



A
Up-down pose
Left-right pose
Lighting direction

Task Decomposition

View information is used for task decomposition



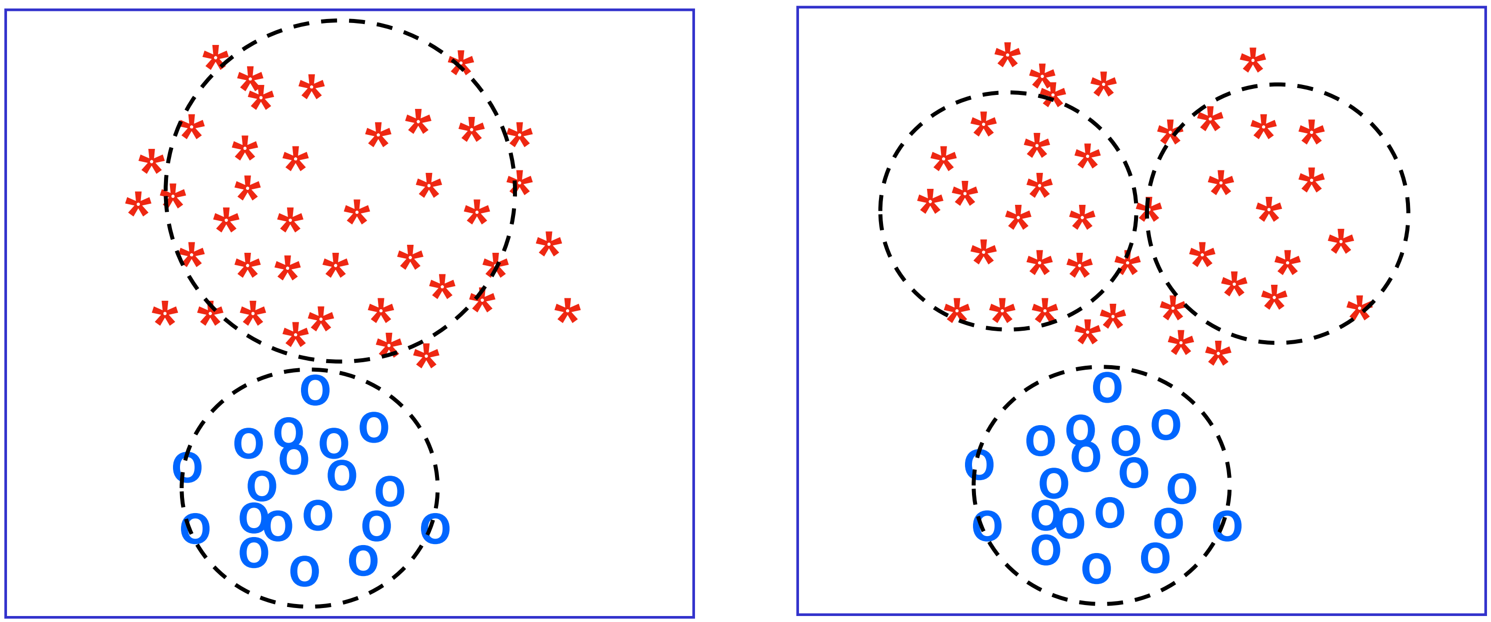

Gender Recognition Using SVM and M^3-SVM with PK
SVM
PK M^3-SVM
Classification Accuracy
0.98
0.96
0.94
0.92
0.9
0.88
0.86
0.84
0.82
0.8
0.78
0
2
4
6
8
10
12
14
16
Different Gender Data Test Sets

Gender Recognition Using a SVM With Equal Clustering

(J. Luo & B. L. Lu, 2005)

Equal Clustering

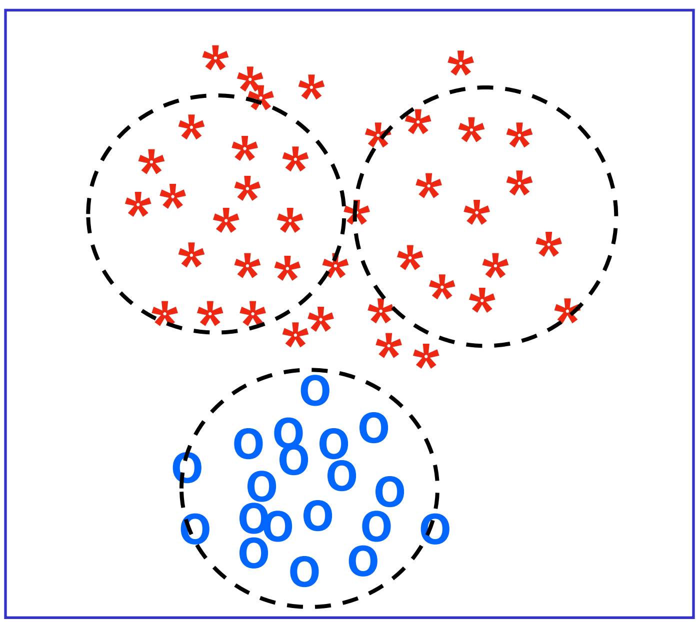
- Based on the algorithm "GeoClust" (Choudhury, Nair and Keane, 2002)
- To generate spatially localized clusters that contain (nearly) equal number of samples to keep load balance.



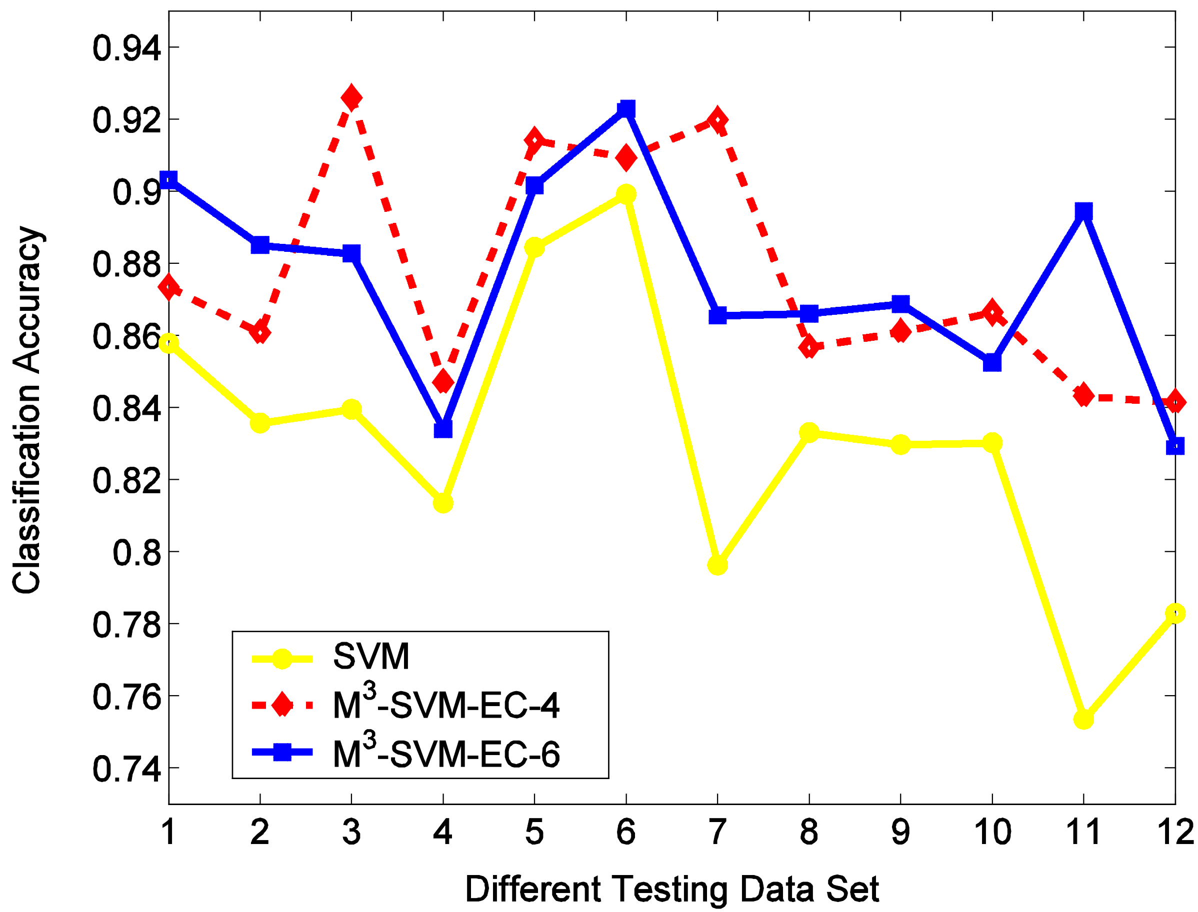
Basic Idea of Equal Clustering

- Solve an unconstrained nonlinear programming problem as follows:

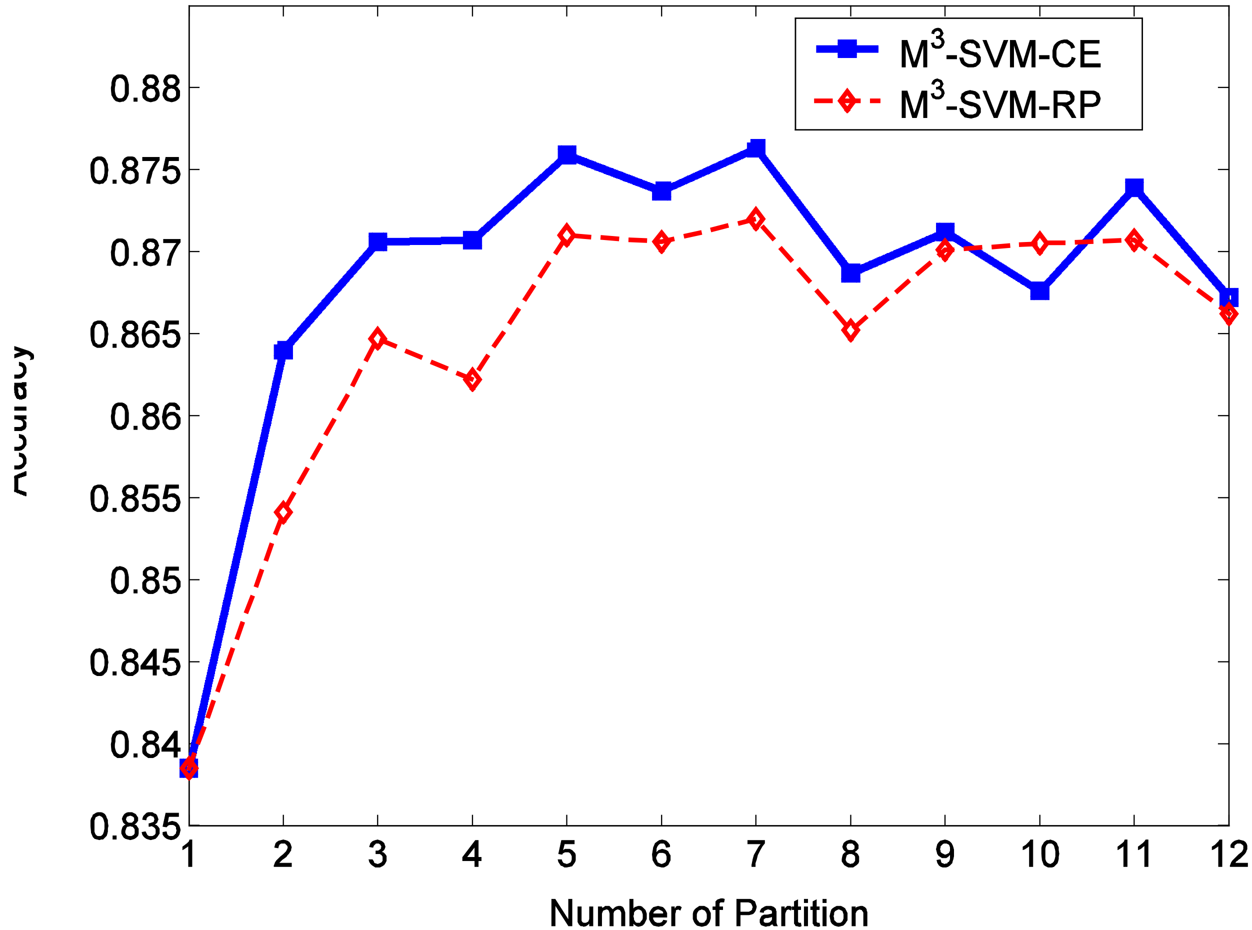
$$\underset{c_1, c_2, \ldots, c_m}{\text{Minimize}} \; h = \max_{i=1}^{m} \left| W_i - \overline{W} \right|$$



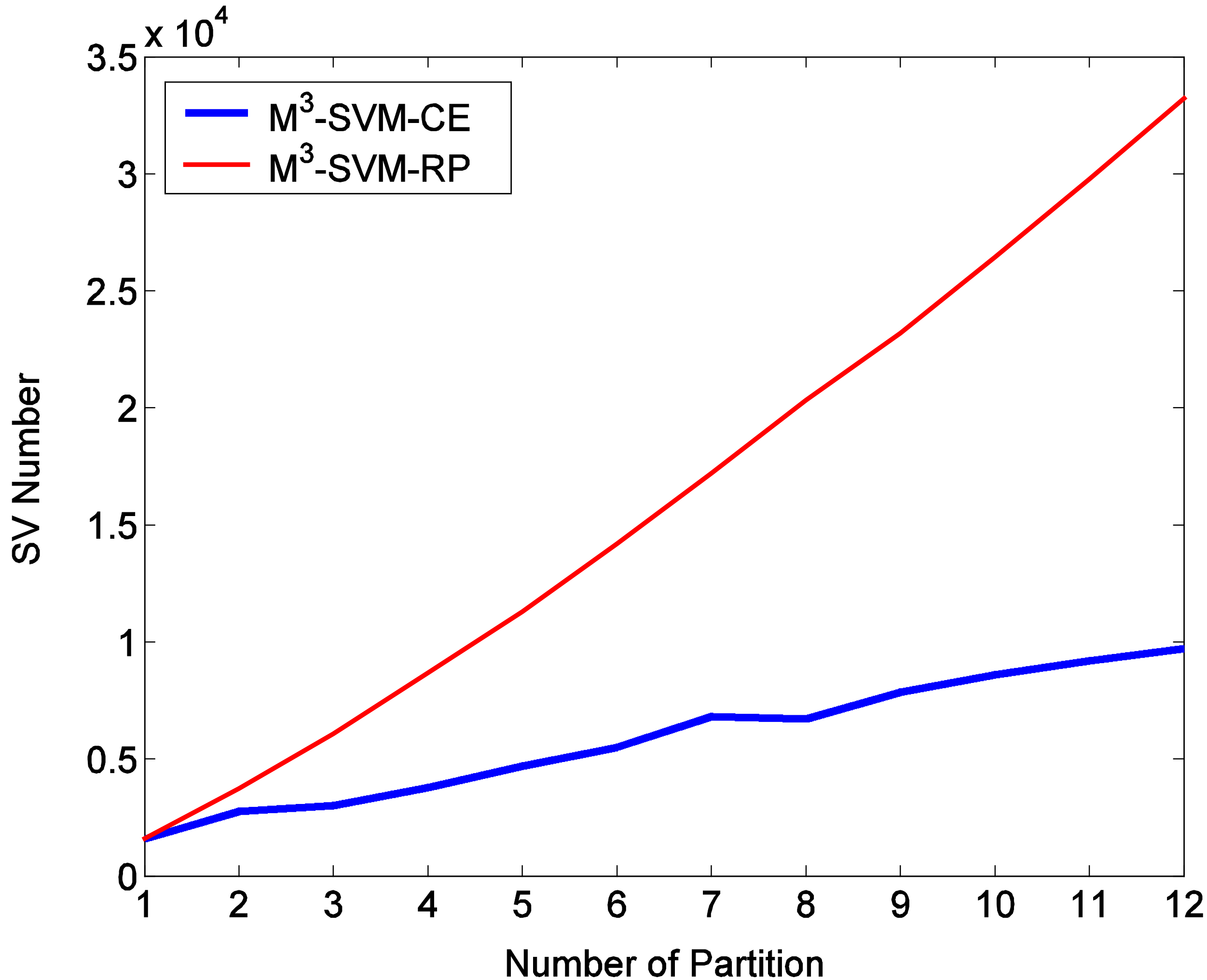
Gender Estimation on Peal dataset



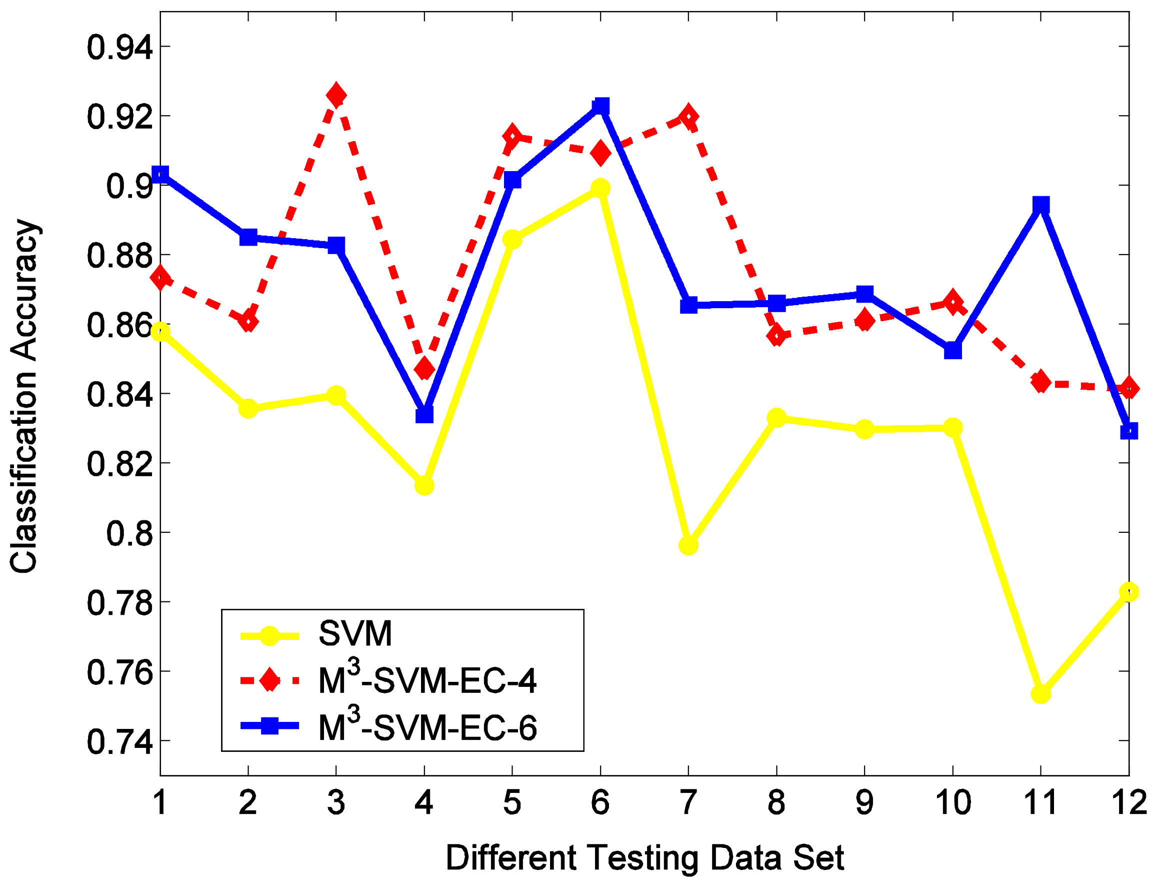
Comparison of Generalization Accuracy



Comparison of Number of SVs



Results of Gender Recognition



SVM software packages

- LibSVM
  - Http://www.csie.ntu.edu.tw/~cjlin/libsvm/
  - Chih-Chung Chang and Chin-Jen Lin
- SVM^{light}
  - http://svmlight.joachims.org/
  - Thorsten Joachims

LibSVM

- Various language versions
  - C++, C#, java, MatLab, etc.
  - Recommend C++ version
- The source code is readable
- The interface is clear

LibSVM

- Two executable files
  - Train.exe
    - Compiled by svm.cpp, svm.h and svm-train.c
  - Test.exe
    - Complied by svm.cpp, svm.h and svm-predict.c

LibSVM

- Description of svmtrain.exe
  - “one versus one” is implemented a solution to multi-class problem
  - Several frequently used parameters
    - -s : svm type (0 for classification)
    - -t : kernel type (2 for RBF kernel)
    - -g : gamma value
    - -c : panelized cost
    - e. g.,

    svmtrain -s 0 -t 2 -g 0.5 –c 2 train_file model_file

LibSVM

- Description of svmpredict.exe
  - e. g.,

    svmpredict test_file model_file result_file

LibSVM

- If you want to directly modify the source code and do your homework…
  - The source code has several interface functions. You can write codes to call these functions.
    - svm_train(), svm_predict_values(), svm_save_model(),…
  - Not recommended unless you have strong understanding to SVMs