

第三讲：支持向量机、迁移学习

吕宝粮

计算机科学与工程系

电信楼群3号楼431

Tel: 021-3420-5422

bllu@sjtu.edu.cn

<http://bcmi.sjtu.edu.cn/~blu>

Outline of Lecture Three

- Vapnik-Chervonenkis (VC) dimension
- Support Vector Machine
- Transfer Learning
- Performance evaluation Index

Vapnik-Chervonenkis (VC) dimension

Measures of Complexity

- “Complexity” is a measure of a set of classifiers, not any specific (fixed) classifier
- Many possible measures
 - degrees of freedom
 - description length
 - Vapnik-Chervonenkis dimension
 - etc.
- There are many reasons for introducing a measure of complexity
 - generalization error guarantees
 - selection among competing families of classifiers

VC-dimension: Preliminaries

- **A set of classifiers F :**

For example, this could be the set of all possible linear separators, where $h \in F$ means that

$$h(\mathbf{x}) = \text{sign} (w_0 + \mathbf{w}^T \mathbf{x})$$

for some values of the parameters \mathbf{w}, w_0 .

- **Complexity:** how many different ways can we label n training points $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ with classifiers $h \in F$?

In other words, how many distinct binary vectors

$$[h(\mathbf{x}_1) \ h(\mathbf{x}_2) \ \dots \ h(\mathbf{x}_n)]$$

do we get by trying each $h \in F$ in turn?

$$\begin{bmatrix} -1 & 1 & \dots & 1 \end{bmatrix} \ h_1$$
$$\begin{bmatrix} 1 & -1 & \dots & 1 \end{bmatrix} \ h_2$$

...

Shattering a Set of Instances

Definition: a *dichotomy* of a set S is a partition of S into two disjoint subsets. (二分)

Definition: a set of instances S is *shattered* by hypothesis space H if and only if for every dichotomy of S there exists some hypothesis in H consistent with this dichotomy. (打散)

打散：如果存在一个有 h 个样本的样本集能够被一个函数集中的函数按照所有可能的 2^h 种形式分为两类，则称函数集能够将样本数为 h 的样本集打散；

VC维：如果函数集能够打散 h 个样本的样本集，而不能打散 $h+1$ 个样本的样本集，则称函数集的VC维为 h 。

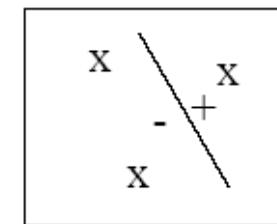
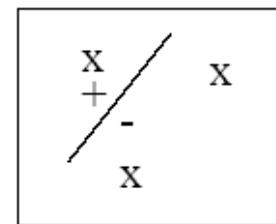
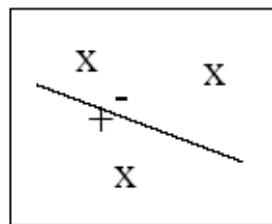
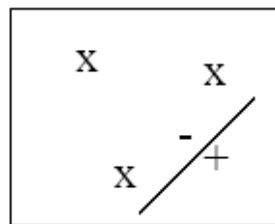
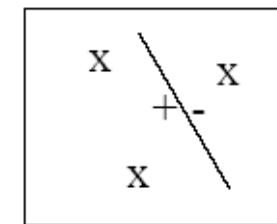
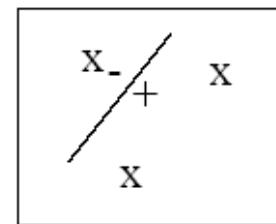
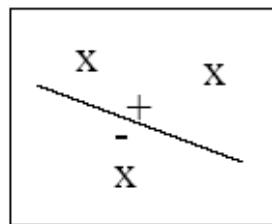
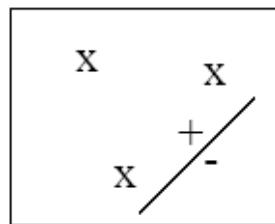
VC-dimension: Shattering

- A set of classifiers F *shatters* n points $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ if

$$[h(\mathbf{x}_1) \ h(\mathbf{x}_2) \ \dots \ h(\mathbf{x}_n)], \ h \in F$$

generates all 2^n distinct labelings.

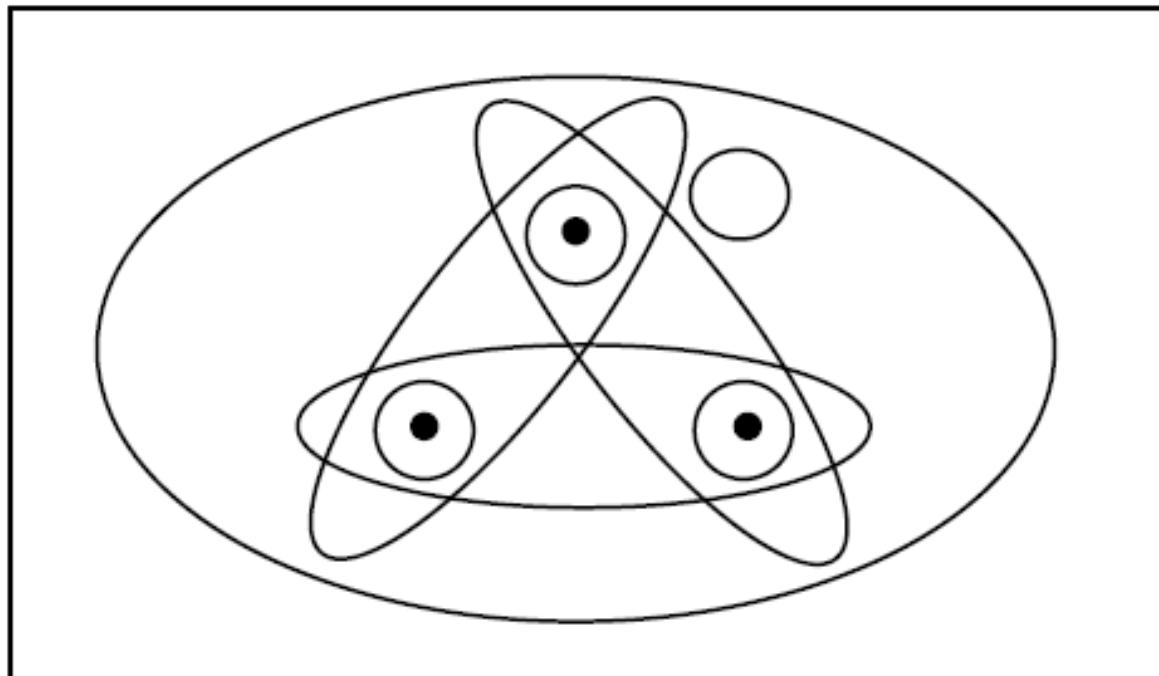
- Example: linear decision boundaries shatter (any) 3 points in 2D



but not any 4 points...

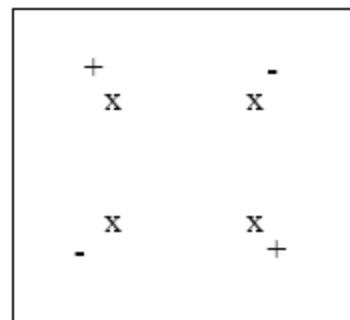
Three Instances Shattered

Instance space X



VC-dimension: Shattering -2

- We cannot shatter 4 points in 2D with linear separators
For example, the following labeling



cannot be produced with any linear separator

- More generally: the set of all d -dimensional linear separators
can shatter exactly $d + 1$ points

VC-dimension

- The VC-dimension d_{VC} of a set of classifiers F is the largest number of points that F can shatter
- This is a combinatorial concept and doesn't depend on what type of classifier we use, only how "flexible" the set of classifiers is

Example: Let F be a set of classifiers defined in terms of linear combinations of m **fixed** basis functions

$$h(\mathbf{x}) = \text{sign} (w_0 + w_1\phi_1(\mathbf{x}) + \dots + w_m\phi_m(\mathbf{x}))$$

d_{VC} is at most $m + 1$ regardless of the form of the fixed basis functions.

Bounds on the VC dimension of NNs

- The VC dimension of feedforward network with a threshold function is $O(W \log W)$
- The VC dimension of feedforward network with a sigmoid function is $O(W^2)$

Support Vector Machine

Introduction

□ What are benefits SV learning?

- Based on simple idea
- High performance in practical applications

□ Characteristics of SV method

- Can dealing with complex nonlinear problems
(pattern recognition, regression, feature extraction)
- But working with a simple linear algorithm
(by the use of kernels)

Empirical Risk

- We want to estimate a function using training data

$$T = \{(X_i, d_i)\}_{i=1}^N \rightarrow F(X, W)$$

- Loss between desired response and actual response

$$L(d, F(X, W)) = (d - F(X, W))^2$$

- Expected risk (风险泛函)

$$R(W) = \frac{1}{2} \int L(d, F(X; W)) dF_{X,D}(X, d)$$

- Empirical risk (经验风险泛函)

$$R_{emp}(W) = \frac{1}{N} \sum_{i=1}^N L(d_i, F(X_i, W))$$

Empirical risk minimization principle

- The true expected risk is approximated by empirical risk

$$R_{emp}(W) = \frac{1}{N} \sum_{i=1}^N L(d_i, F(X_i, W))$$

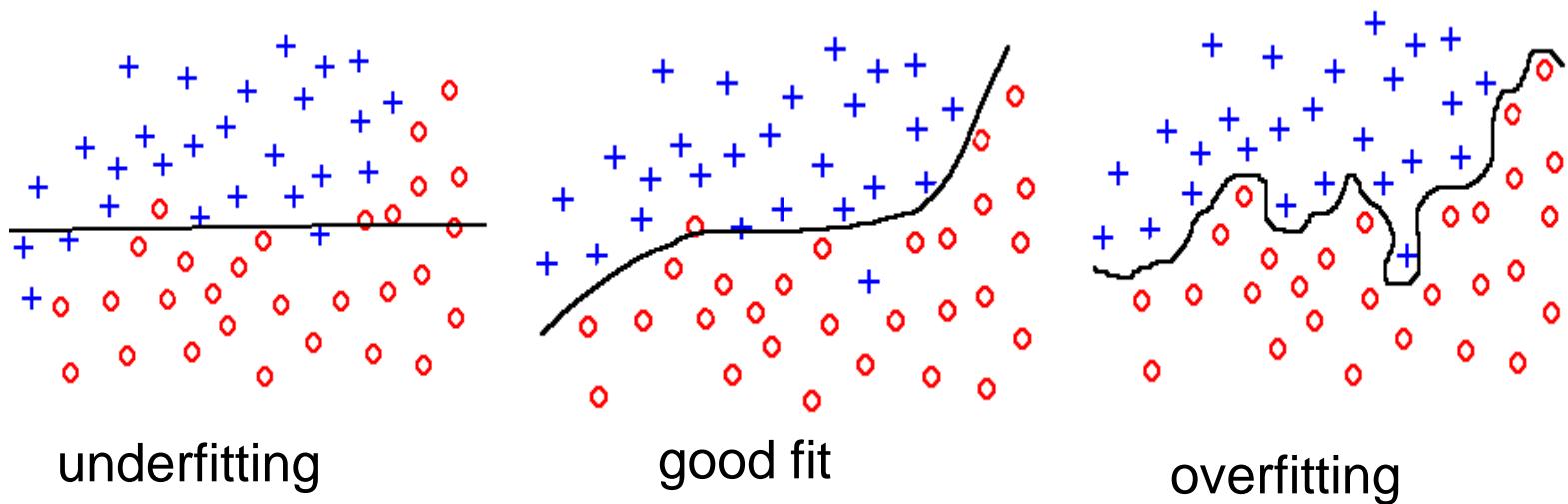
- The learning based on the empirical minimization principle is defined as

$$W^* = \arg \min_W R_{emp}(W)$$

Examples of algorithms: Perceptron, Back-propagation, etc.

Overfitting and underfitting

- Problem: How rich class of classifications $F(X, W)$ to use



- Problem of generalization: A small empirical risk $R_{emp}(W)$ does not imply small true expected risk $R(W)$

Structural Risk Minimization

- Statistical learning theory : Vapnik & Chervonenkis
- An upper bound on the expected risk of a classification rule

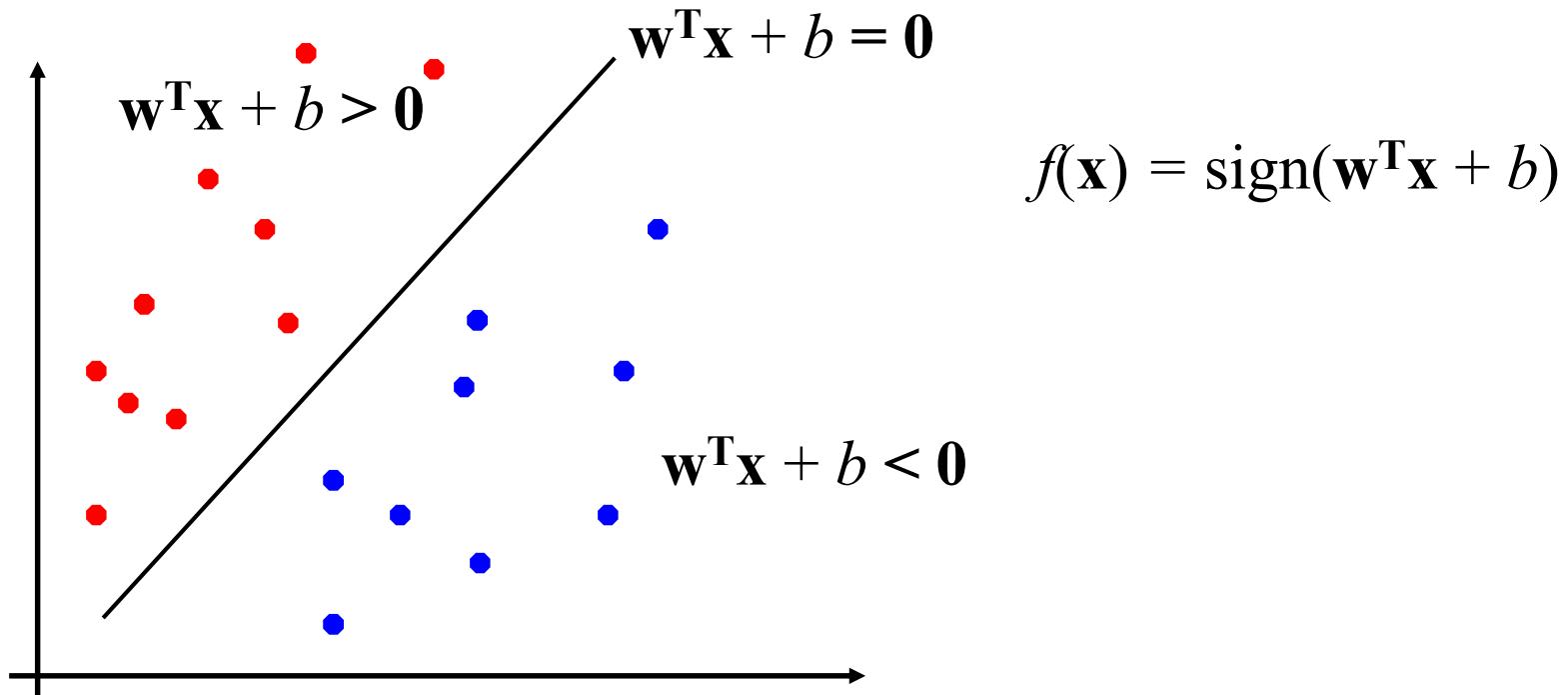
$$R(W) \leq R_{emp}(W) + \sqrt{\frac{h[\log(2N/h) + 1] - \log(\alpha)}{N}}$$

where N is the number of training data, h is VC-dimension of class of functions.

- SRM Principle: to find a network structure such that decreasing the VC dimension occurs at the expense of the smallest possible increase in training error

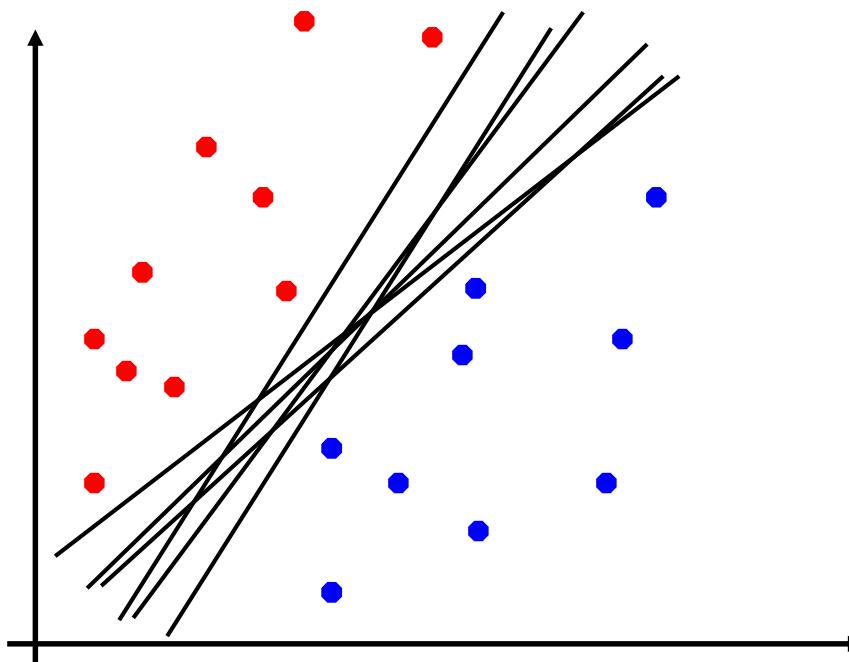
Perceptron Revisited: Linear Separators

- Binary classification can be viewed as the task of separating classes in feature space:



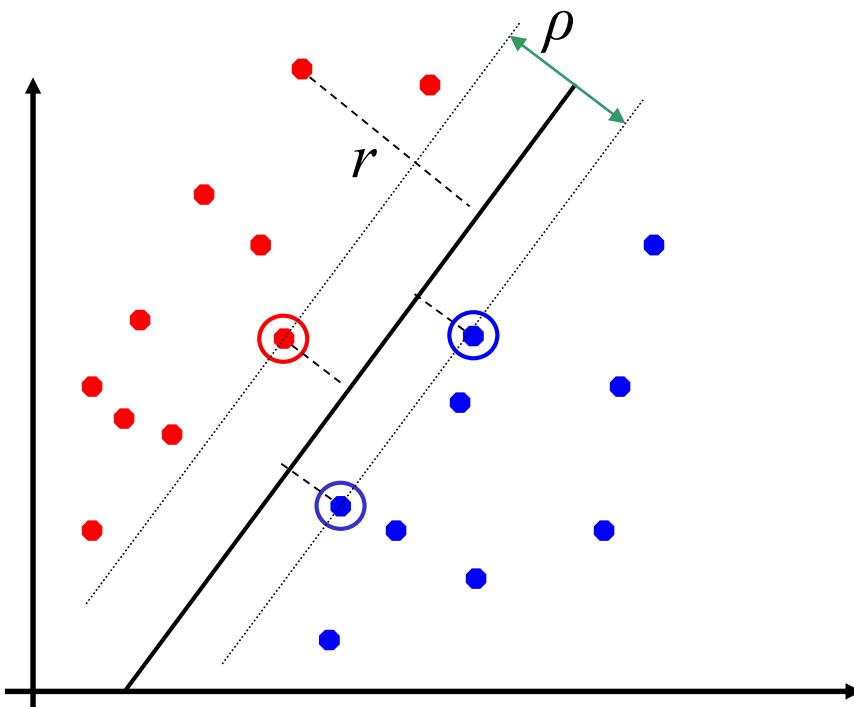
Linear Separators

- Which of the linear separators is optimal?



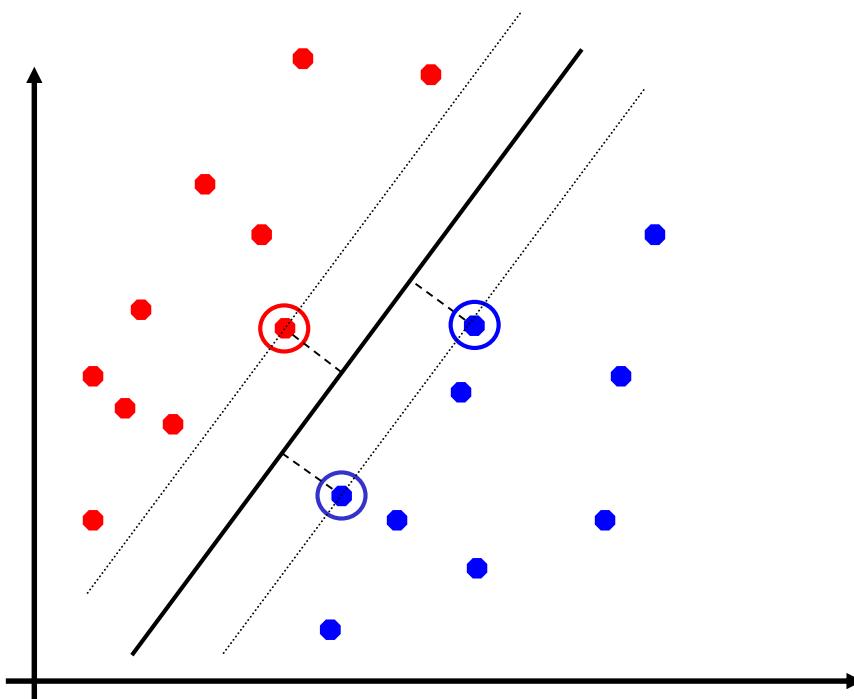
Classification Margin

- Distance from example x_i to the separator is $r = \frac{\mathbf{w}^T \mathbf{x}_i + b}{\|\mathbf{w}\|}$
- Examples closest to the hyperplane are *support vectors*.
- *Margin* ρ of the separator is the distance between support vectors.



Maximum Margin Classification

- Maximizing the margin is good according to intuition and PAC theory.
- Implies that only support vectors matter; other training examples are ignorable.



Linear SVM Mathematically

- Let training set $\{(\mathbf{x}_i, y_i)\}_{i=1..n}$, $\mathbf{x}_i \in \mathbb{R}^d$, $y_i \in \{-1, 1\}$ be separated by a hyperplane with margin ρ . Then for each training example (\mathbf{x}_i, y_i) :

$$\begin{aligned}\mathbf{w}^T \mathbf{x}_i + b &\leq -\rho/2 & \text{if } y_i = -1 \\ \mathbf{w}^T \mathbf{x}_i + b &\geq \rho/2 & \text{if } y_i = 1\end{aligned} \Leftrightarrow y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq \rho/2$$

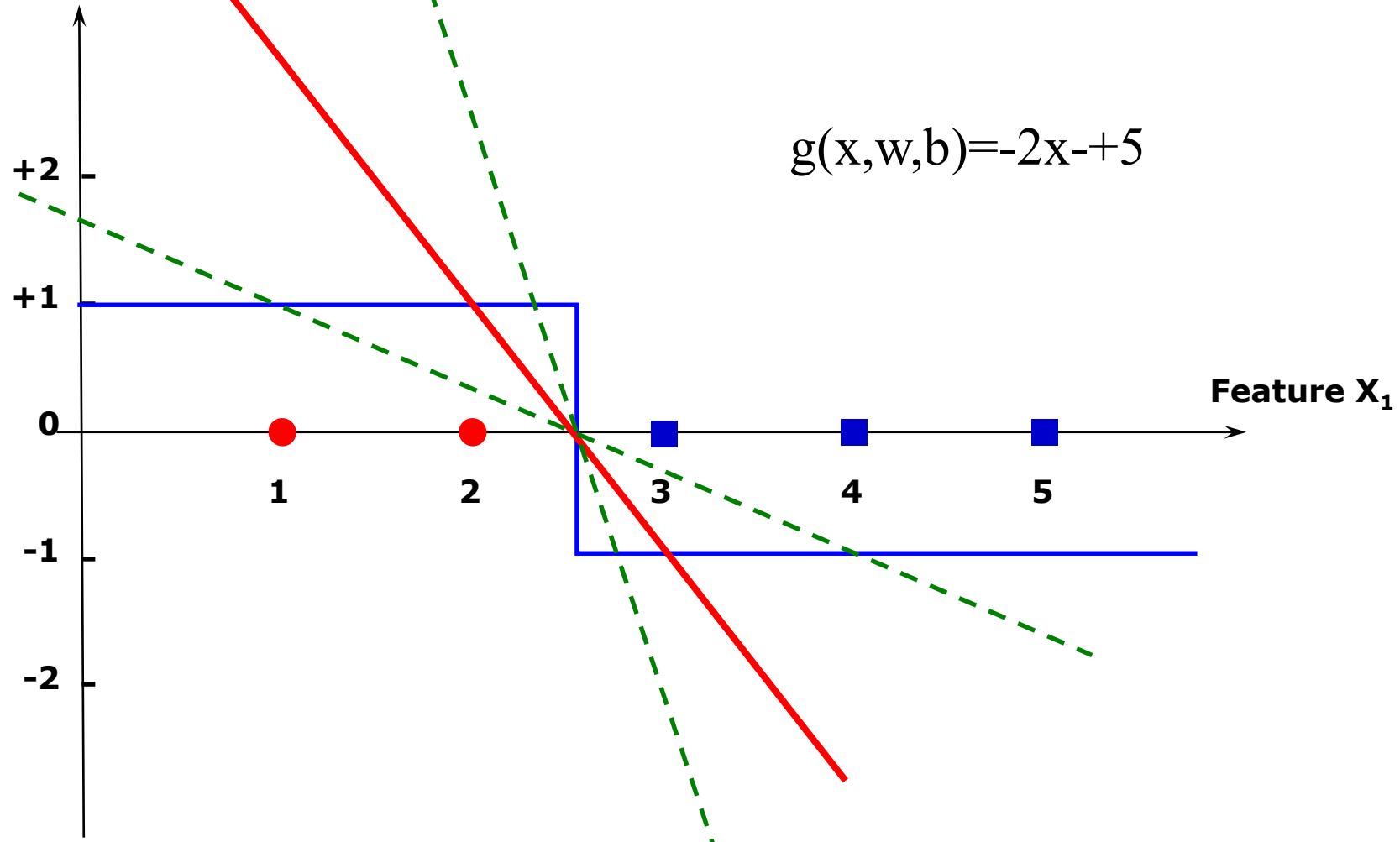
- For every support vector \mathbf{x}_s the above inequality is an equality. After rescaling w and b by $\rho/2$ in the equality, we obtain that distance between each \mathbf{x}_s and the hyperplane is

$$r = \frac{y_s (\mathbf{w}^T \mathbf{x}_s + b)}{\|\mathbf{w}\|} = \frac{1}{\|\mathbf{w}\|}$$

- Then the margin can be expressed through (rescaled) w and b as:

$$\rho = 2r = \frac{2}{\|\mathbf{w}\|}$$

Target $y(=d)$



Linear SVMs Mathematically (cont.)

- Then we can formulate the *quadratic optimization problem*:

Find \mathbf{w} and b such that

$$\rho = \frac{2}{\|\mathbf{w}\|} \text{ is maximized}$$

and for all (\mathbf{x}_i, y_i) , $i=1..n$: $y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1$

Which can be reformulated as:

Find \mathbf{w} and b such that

$$\Phi(\mathbf{w}) = \|\mathbf{w}\|^2 = \mathbf{w}^T \mathbf{w} \text{ is minimized}$$

and for all (\mathbf{x}_i, y_i) , $i=1..n$: $y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1$

Solving the Optimization Problem

Find \mathbf{w} and b such that

$\Phi(\mathbf{w}) = \mathbf{w}^T \mathbf{w}$ is minimized

and for all (\mathbf{x}_i, y_i) , $i=1..n$: $y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1$

- Need to optimize a quadratic function subject to linear constraints.
- Quadratic optimization problems are a well-known class of mathematical programming problems for which several (non-trivial) algorithms exist.
- The solution involves constructing a dual problem where a Lagrange multiplier α_i is associated with every inequality constraint in the primal (original) problem:

Find $\alpha_1 \dots \alpha_n$ such that

$Q(\alpha) = \sum \alpha_i - \frac{1}{2} \sum \sum \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$ is maximized and

$$(1) \quad \sum \alpha_i y_i = 0$$

$$(2) \quad \alpha_i \geq 0 \text{ for all } \alpha_i$$

The Optimization Problem Solution

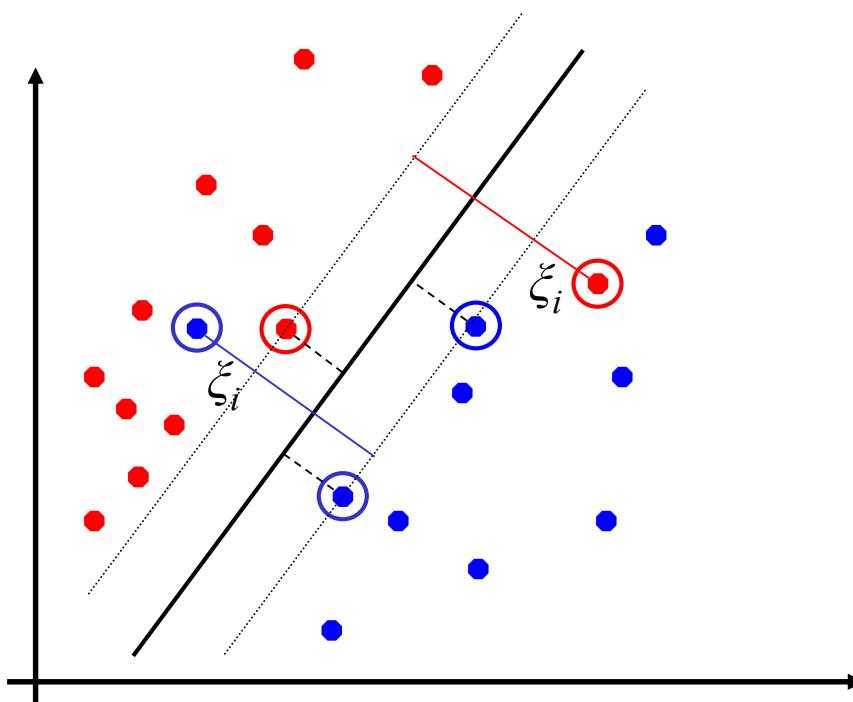
- Given a solution $\alpha_1 \dots \alpha_n$ to the dual problem, solution to the primal is:

$$w = \sum \alpha_i y_i x_i \quad b = y_k - \sum \alpha_i y_i x_i^T x_k \quad \text{for any } \alpha_k > 0$$

- Each non-zero α_i indicates that corresponding x_i is a support vector.
- Then the classifying function is (note that we don't need w explicitly):
$$f(x) = \sum \alpha_i y_i x_i^T x + b$$
- Notice that it relies on an inner product between the test point x and the support vectors x_i .
- Also keep in mind that solving the optimization problem involved computing the inner products $x_i^T x_j$ between all training points.

Soft Margin Classification

- What if the training set is not linearly separable?
- Slack variables ξ_i can be added to allow misclassification of difficult or noisy examples, resulting margin called soft.



Soft Margin Classification Mathematically

□ The old formulation:

Find \mathbf{w} and b such that

$\Phi(\mathbf{w}) = \mathbf{w}^T \mathbf{w}$ is minimized

and for all (\mathbf{x}_i, y_i) , $i=1..n$: $y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1$

□ Modified formulation incorporates slack variables:

Find \mathbf{w} and b such that

$\Phi(\mathbf{w}) = \mathbf{w}^T \mathbf{w} + C \sum \xi_i$ is minimized

and for all (\mathbf{x}_i, y_i) , $i=1..n$: $y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i$, $\xi_i \geq 0$

□ Parameter C can be viewed as a way to control overfitting: it “trades off” the relative importance of maximizing the margin and fitting the training data.

Soft Margin Classification – Solution

- Dual problem is identical to separable case (would not be identical if the 2-norm penalty for slack variables $C\sum \xi_i^2$ was used in primal objective, we would need additional Lagrange multipliers for slack variables):

Find $\alpha_1, \dots, \alpha_N$ such that

$Q(\alpha) = \sum \alpha_i - \frac{1}{2} \sum \sum \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$ is maximized and

$$(1) \quad \sum \alpha_i y_i = 0$$

$$(2) \quad 0 \leq \alpha_i \leq C \text{ for all } \alpha_i$$

- Again, \mathbf{x}_i with non-zero α_i will be support vectors.
- Solution to the dual problem is:

$$\mathbf{w} = \sum \alpha_i y_i \mathbf{x}_i$$

$$b = y_k(1 - \xi_k) - \sum \alpha_i y_i \mathbf{x}_i^T \mathbf{x}_k \quad \text{for any } k \text{ s.t. } \alpha_k > 0$$

Again, we don't need to compute \mathbf{w} explicitly for classification:

$$f(\mathbf{x}) = \sum \alpha_i y_i \mathbf{x}_i^T \mathbf{x} + b$$

SVM Boundaries with different C

Find \mathbf{w} and b such that

$$\Phi(\mathbf{w}) = \mathbf{w}^T \mathbf{w} + C \sum \xi_i \text{ is minimized}$$

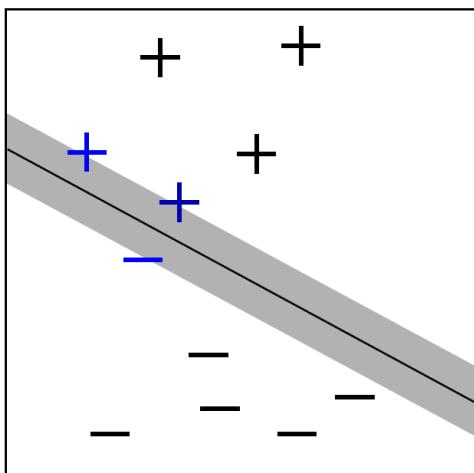
$$\text{and for all } (\mathbf{x}_i, y_i), i=1..n : y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0$$

Find $\alpha_1 \dots \alpha_N$ such that

$$Q(\mathbf{\alpha}) = \sum \alpha_i - \frac{1}{2} \sum \sum \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \text{ is maximized and}$$

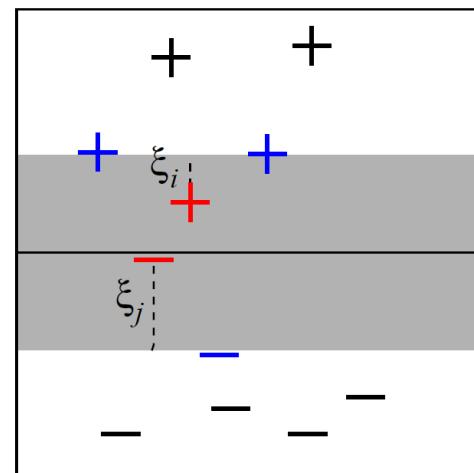
$$(1) \quad \sum \alpha_i y_i = 0$$

$$(2) \quad 0 \leq \alpha_i \leq C \text{ for all } \alpha_i$$



Large C

Small C



Theoretical Justification for Maximum Margins

- Vapnik has proved the following:

The class of optimal linear separators has VC dimension h bounded from above as

$$h \leq \min \left\{ \left\lceil \frac{D^2}{\rho^2} \right\rceil, m_0 \right\} + 1$$

where ρ is the margin, D is the diameter of the smallest sphere that can enclose all of the training examples, and m_0 is the dimensionality.

- Intuitively, this implies that regardless of dimensionality m_0 we can minimize the VC dimension by maximizing the margin ρ .
- Thus, complexity of the classifier is kept small regardless of dimensionality.

Linear SVMs: Overview

- The classifier is a separating hyperplane.
- Most “important” training points are support vectors; they define the hyperplane.
- Quadratic optimization algorithms can identify which training points x_i are support vectors with non-zero Lagrangian multipliers α_i .
- Both in the dual formulation of the problem and in the solution training points appear only inside inner products:

Find $\alpha_1 \dots \alpha_N$ such that

$Q(\alpha) = \sum \alpha_i - \frac{1}{2} \sum \sum \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$ is maximized and

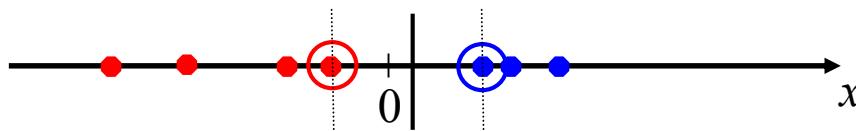
$$(1) \sum \alpha_i y_i = 0$$

$$(2) 0 \leq \alpha_i \leq C \text{ for all } \alpha_i$$

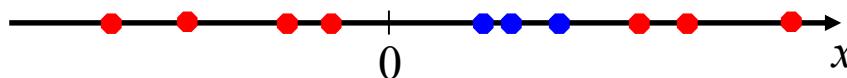
$$f(\mathbf{x}) = \sum \alpha_i y_i \mathbf{x}_i^T \mathbf{x} + b$$

Non-linear SVMs

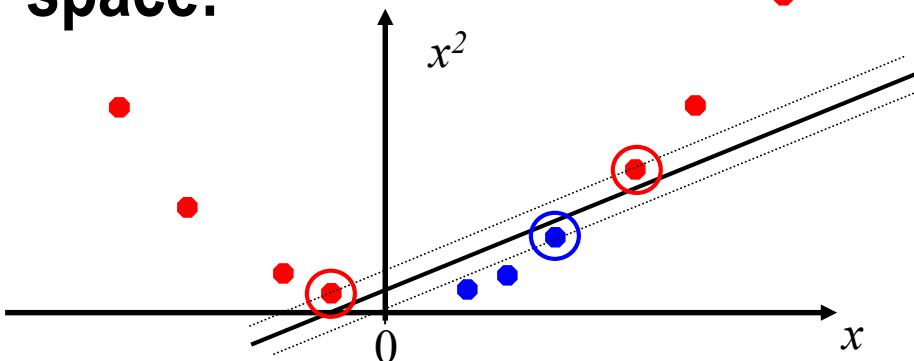
- Datasets that are linearly separable with some noise work out great:



- But what are we going to do if the dataset is just too hard?

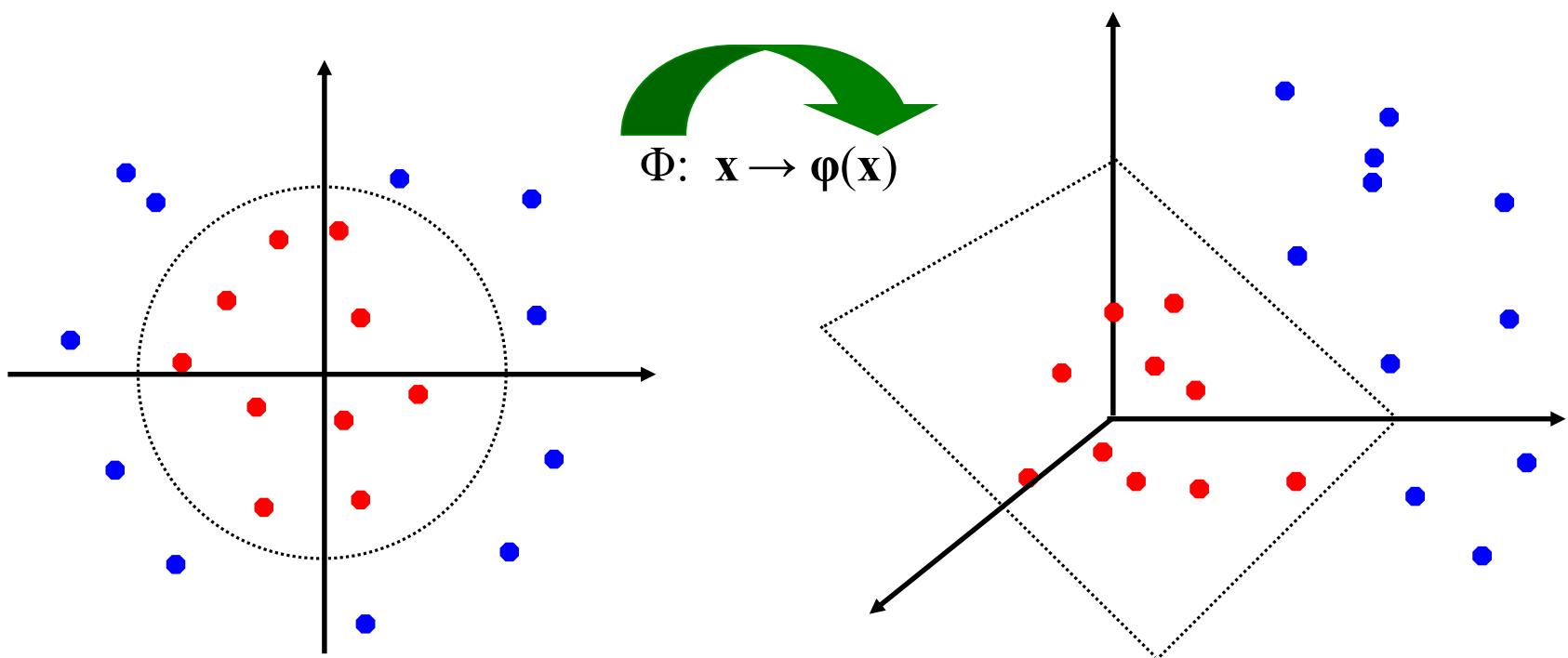


- How about... mapping data to a higher-dimensional space:



Non-linear SVMs: Feature spaces

- General idea: the original feature space can always be mapped to some higher-dimensional feature space where the training set is separable:



The “Kernel Trick”

- The linear classifier relies on inner product between vectors $K(x_i, x_j) = x_i^T x_j$
- If every datapoint is mapped into high-dimensional space via some transformation $\Phi: x \rightarrow \varphi(x)$, the inner product becomes:
$$K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$$
- A kernel function is a function that is equivalent to an inner product in some feature space.

Example:

2-dimensional vectors $x = [x_1 \ x_2]$; let $K(x_i, x_j) = (1 + x_i^T x_j)^2$,

Need to show that $K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$:

$$\begin{aligned} K(x_i, x_j) &= (1 + x_i^T x_j)^2 = 1 + x_{i1}^2 x_{j1}^2 + 2x_{i1}x_{j1}x_{i2}x_{j2} + x_{i2}^2 x_{j2}^2 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2} \\ &= [1 \ x_{i1}^2 \ \sqrt{2x_{i1}x_{i2}} \ x_{i2}^2 \ \sqrt{2x_{i1}} \ \sqrt{2x_{i2}}]^T [1 \ x_{j1}^2 \ \sqrt{2x_{j1}x_{j2}} \ x_{j2}^2 \ \sqrt{2x_{j1}} \ \sqrt{2x_{j2}}] \\ &= \varphi(x_i)^T \varphi(x_j), \end{aligned}$$

where $\varphi(x) = [1 \ x_1^2 \ \sqrt{2x_1x_2} \ x_2^2 \ \sqrt{2x_1} \ \sqrt{2x_2}]$

Thus, a kernel function implicitly maps data to a high-dimensional space (without the need to compute each $\varphi(x)$ explicitly).

What Functions are Kernels?

- For some functions $K(x_i, x_j)$ checking that $K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$ can be cumbersome.
- Mercer's theorem:
Every semi-positive definite symmetric function is a kernel
- Semi-positive definite symmetric functions correspond to a semi-positive definite symmetric Gram matrix:

K=

$K(\mathbf{x}_1, \mathbf{x}_1)$	$K(\mathbf{x}_1, \mathbf{x}_2)$	$K(\mathbf{x}_1, \mathbf{x}_3)$...	$K(\mathbf{x}_1, \mathbf{x}_n)$
$K(\mathbf{x}_2, \mathbf{x}_1)$	$K(\mathbf{x}_2, \mathbf{x}_2)$	$K(\mathbf{x}_2, \mathbf{x}_3)$		$K(\mathbf{x}_2, \mathbf{x}_n)$
...
$K(\mathbf{x}_n, \mathbf{x}_1)$	$K(\mathbf{x}_n, \mathbf{x}_2)$	$K(\mathbf{x}_n, \mathbf{x}_3)$...	$K(\mathbf{x}_n, \mathbf{x}_n)$

Examples of Kernel Functions

- Linear kernel

$$K(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^T \mathbf{x}')$$

- Polynomial kernel

$$K(\mathbf{x}, \mathbf{x}') = (1 + (\mathbf{x}^T \mathbf{x}'))^p$$

where $p = 2, 3, \dots$. To get the feature vectors we concatenate all p^{th} order polynomial terms of the components of \mathbf{x} (weighted appropriately)

- Radial basis kernel

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{1}{2}\|\mathbf{x} - \mathbf{x}'\|^2\right)$$

In this case the feature space consists of functions and results in a *non-parametric* classifier.

Non-linear SVMs Mathematically

- Dual problem formulation:

Find $\alpha_1 \dots \alpha_n$ such that

$Q(\alpha) = \sum \alpha_i - \frac{1}{2} \sum \sum \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$ is maximized and

$$(1) \quad \sum \alpha_i y_i = 0$$

$$(2) \quad \alpha_i \geq 0 \text{ for all } \alpha_i$$

- The solution is:

$$f(\mathbf{x}) = \sum \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b$$

- Optimization techniques for finding α_i 's remain the same!

Examples of Kernel Functions

- **Linear kernel**

$$K(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^T \mathbf{x}')$$

- **Polynomial kernel**

$$K(\mathbf{x}, \mathbf{x}') = (1 + (\mathbf{x}^T \mathbf{x}'))^p$$

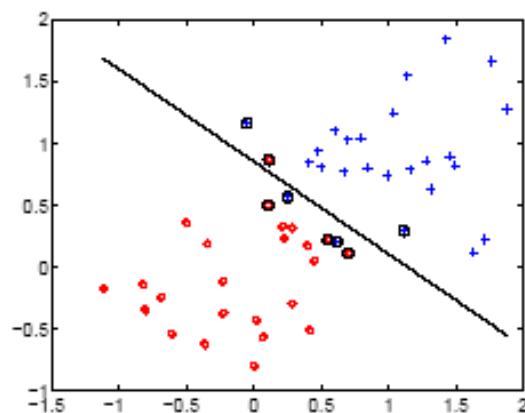
where $p = 2, 3, \dots$. To get the feature vectors we concatenate all p^{th} order polynomial terms of the components of \mathbf{x} (weighted appropriately)

- **Radial basis kernel**

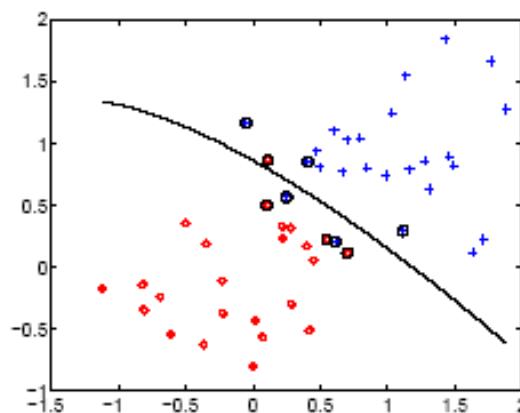
$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{1}{2}\|\mathbf{x} - \mathbf{x}'\|^2\right)$$

In this case the feature space consists of functions and results in a *non-parametric* classifier.

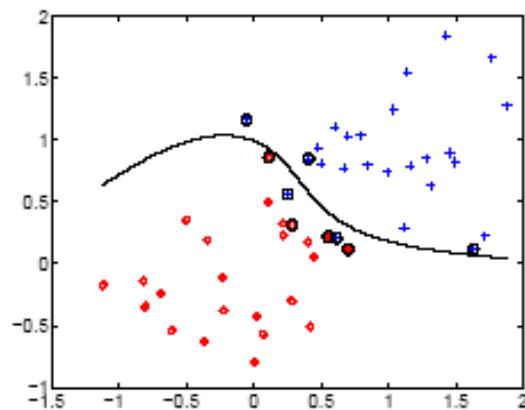
SVM Examples



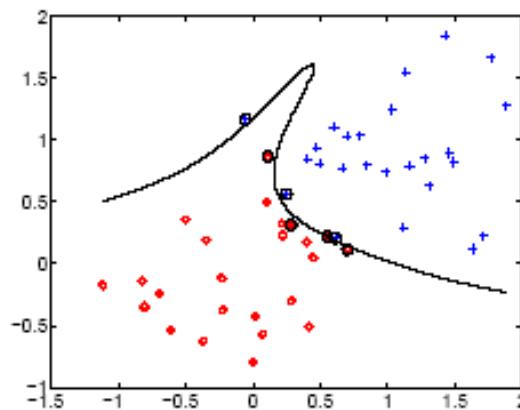
linear



2nd order polynomial



4th order polynomial



8th order polynomial

Key Points

- Learning depends only on dot products of sample pairs.
- Exclusive reliance on dot products enables approach to non-linearly separable problems.
- The classifier depends only on the support vectors, not on all the training points.
- Max margin lowers hypothesis variance.
- The optimal classifier is defined uniquely-there are no “local maxima” in the search space
- Polynomial in number of data points and dimensionality

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.
- 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.

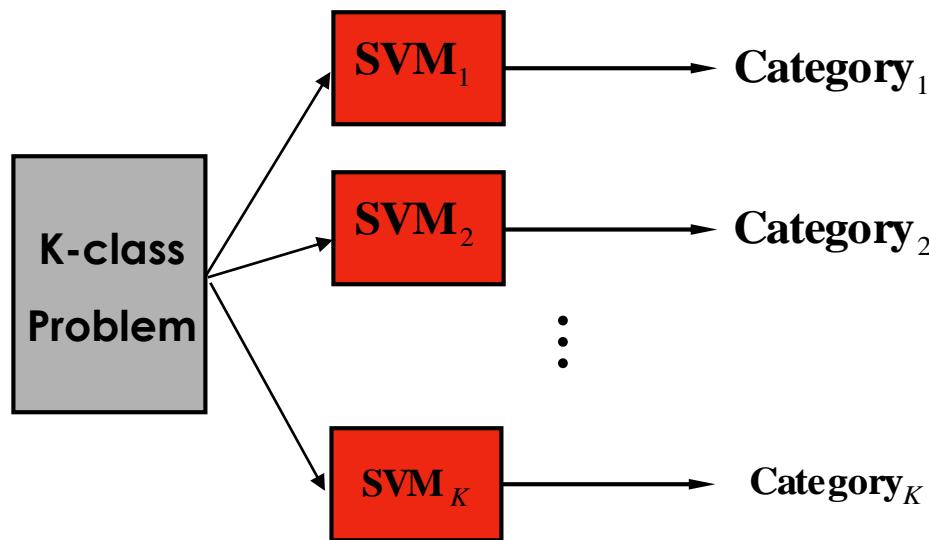
SVMs for Multi-class Classification Problems

Three task decomposition methods:

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

One-Versus-Rest method

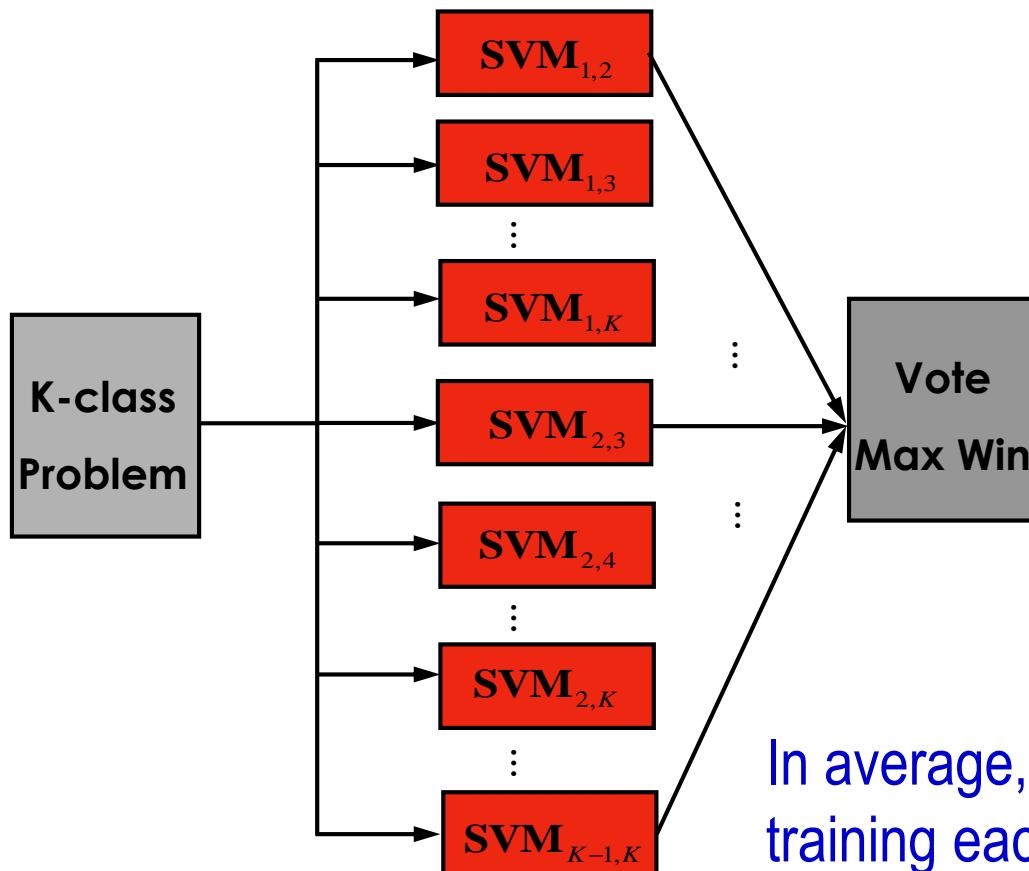
- This method requires one classifier per category. The i th SVM will be trained with all of the examples in the i th class with positive labels, and all other examples with negative labels.



The Number of training data for each classifier is N

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.



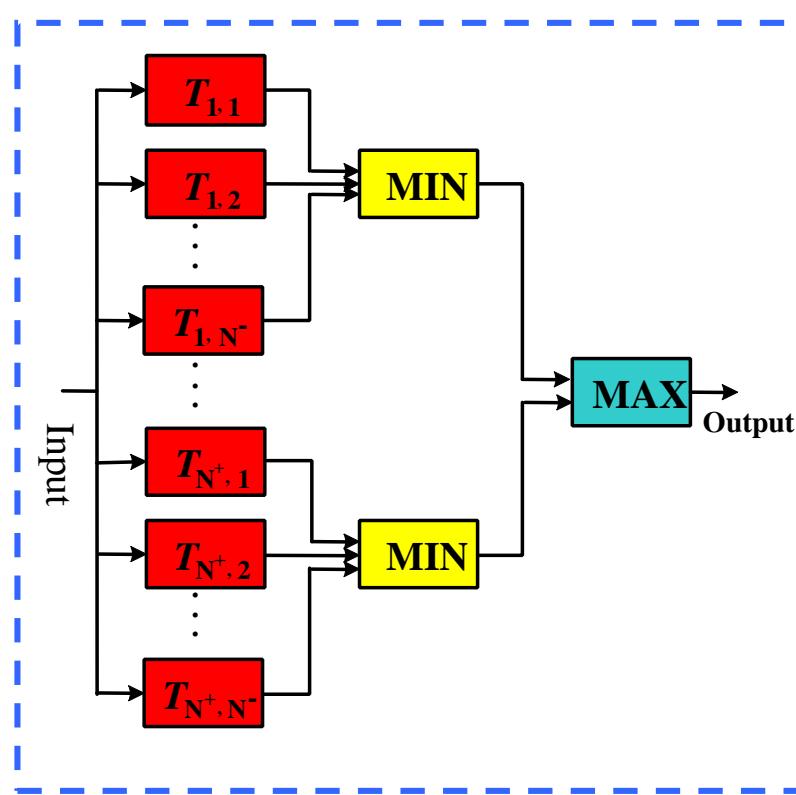
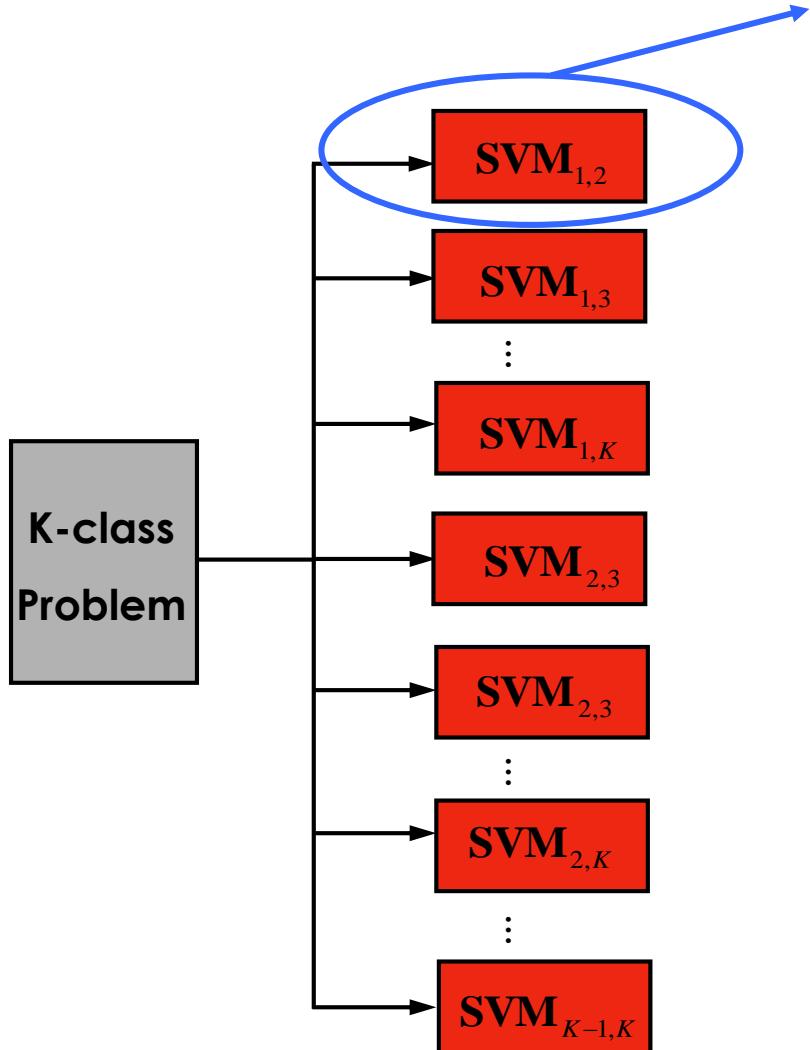
In average, number of data for training each classifier is $2N/K$

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.

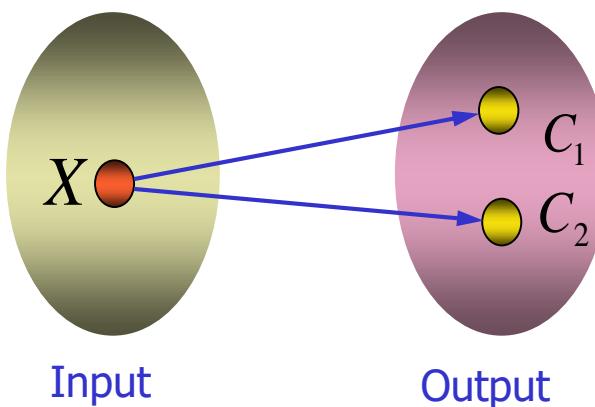


Advantages of part-versus-part method

- 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, y_i ($i = 1, \dots, n$), 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



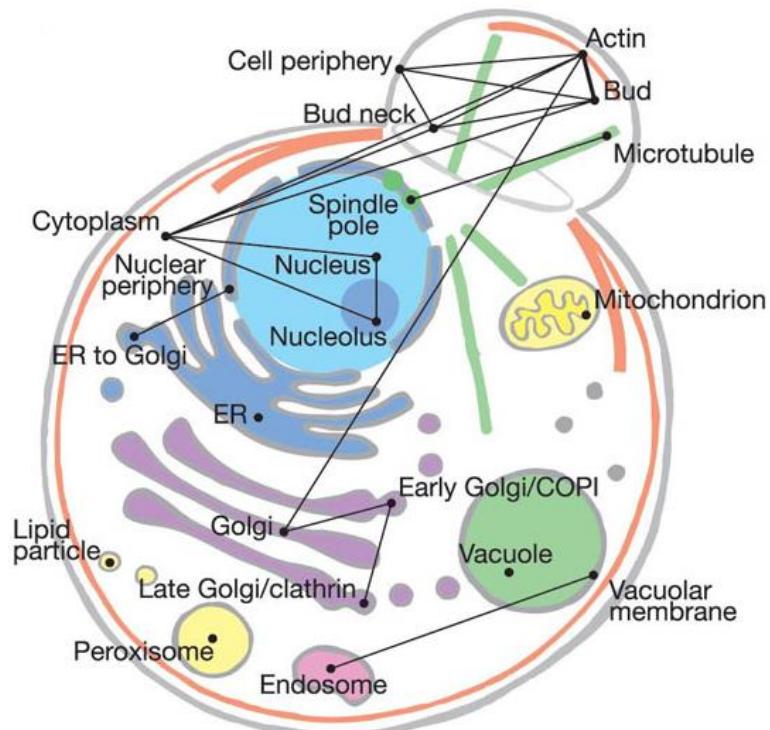
Multi-label problems DO Exist !

- Text categorization

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

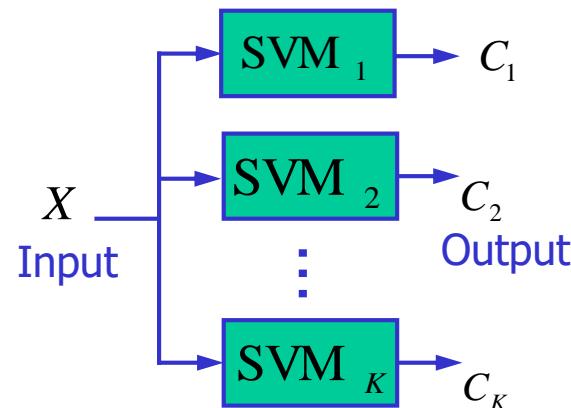
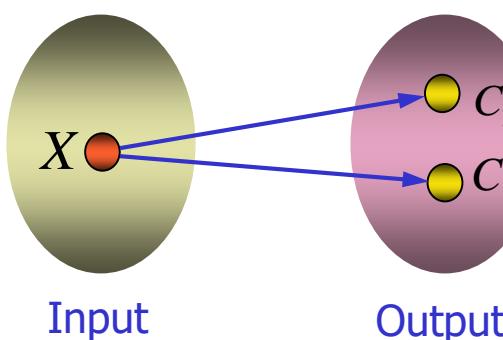
- Subcellular localization of protein subsequence

One protein sequence has at most 5 locations in budding yeast



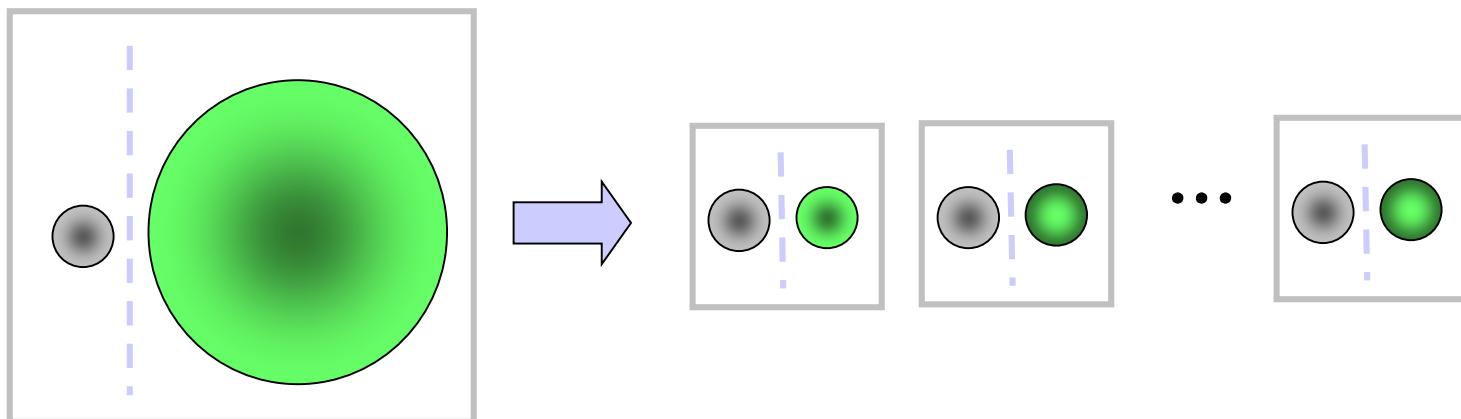
Existing Method for Multi-Label Problem

- 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.



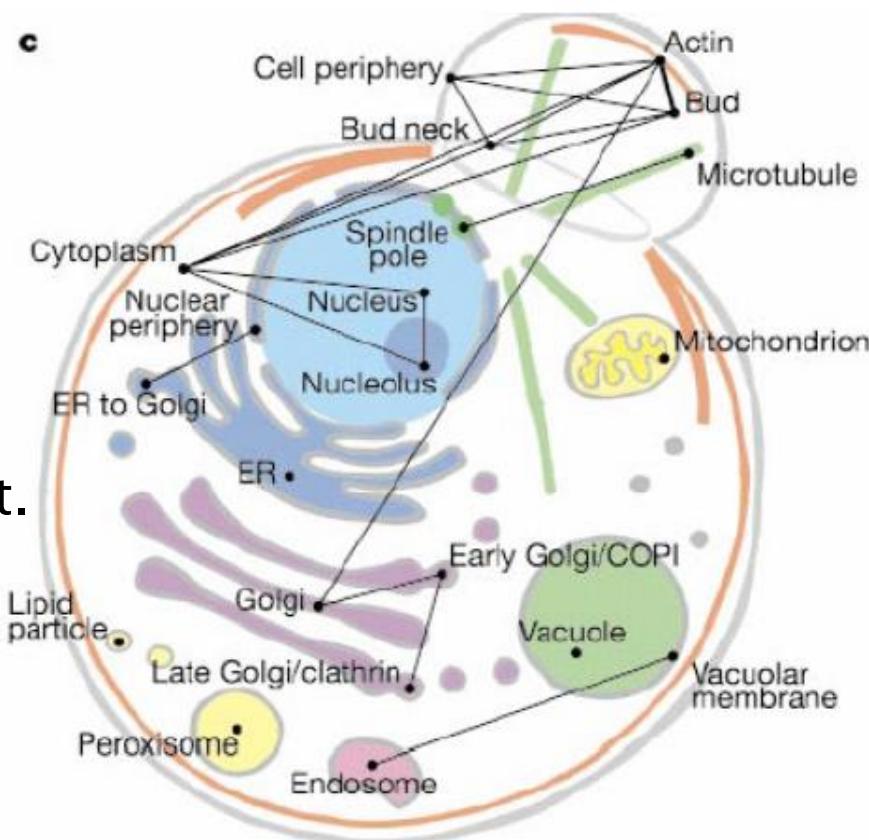
Part-versus-part method

- 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

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



酵母数据集分布

亚细胞位置	序列个数	亚细胞位置	序列个数
Actin	29	Lipid particle	19
Bud	23	Microtubule	20
Bud neck	60	Mitochondrion	494
Cell periphery	106	Nuclear periphery	59
Cytoplasm	1576	Nucleolus	157
Early Golgi	51	Nucleus	1333
Endosome	43	Peroxisome	20
ER	272	Punctate composite	123
ER to Golgi	6	Spindle pole	58
Golgi	40	Vacuolar membrane	54
Late Golgi	37	vacuole	129
总标号数			4709
蛋白质总数			3555

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 = 50, 100, 500, 1000
- 10-fold cross-validation

实验结果

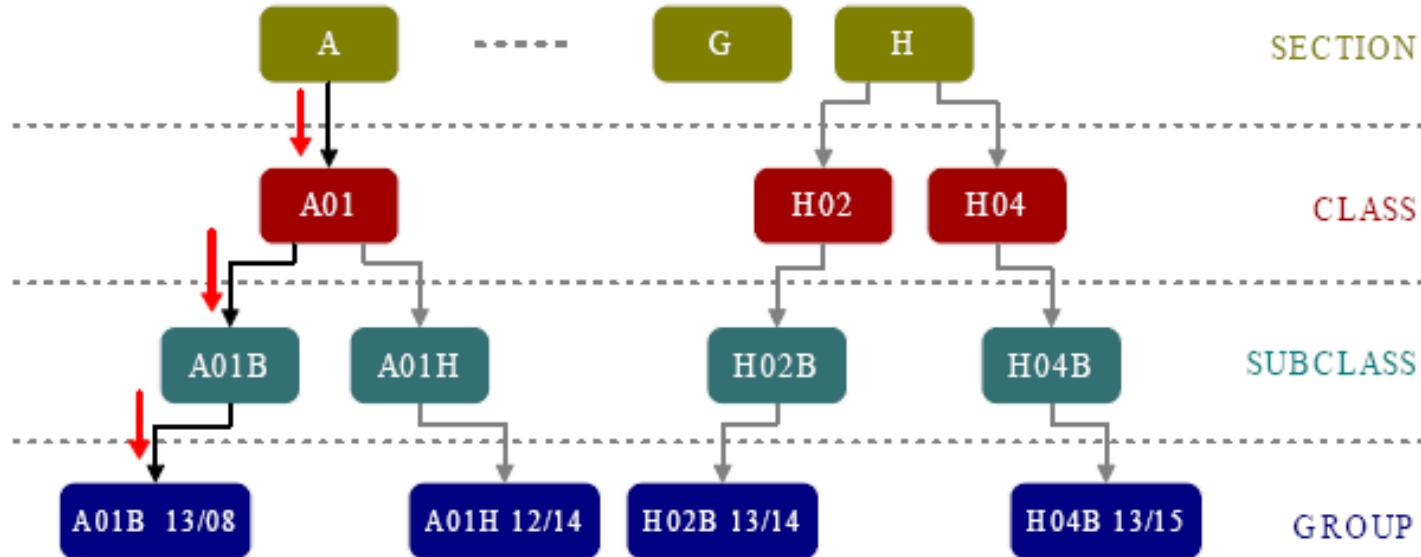
分类器	总体准确率(%)	位置准确率(%)	宏观平均(%)	微观平均(%)
M ³ -SVM(50)	74.3	59.4	54.0	68.8
M ³ -SVM(100)	74.4	58.7	55.5	69.7
M ³ -SVM(500)	73.4	53.6	56.7	73
M ³ -SVM(1000)	71.3	52.1	56.4	73.5
SVM	69.3	49.1	55.6	73.8
NN	67	48.3	49.5	67.5

分类器	串行运行时间 (秒)	最大模块运行时间 (秒)
SVM	33.3	5.7
M ³ SVM(50)	36.3	<0.1
M ³ SVM(100)	26.3	<0.1
M ³ -SVM(500)	23.5	0.6
M ³ SVM(1000)	25.2	2.3

An example of Japanese patent

	PATENT-JA-UPA-1998-000001
<Bibliography> [publication date] [title of invention] ...	(43)【公開日】平成10年(1998)1月6日 (54)【発明の名称】土壤改良方法とその作業機 ...
<Abstract> [purpose] [solution] ...	【課題】心土破碎、特に雪上心土破碎作業の際に積雪... 【解決手段】心土破碎を行うために用いるサブソイウの... ...
<Claims> [claim1] [claim2] ...	【請求項1】サブソイウ作業機を用いて心土破碎作業... 【請求項2】サブソイウ作業機において、そのナイフ... ...
<Description> [technique field] [prior art] [problem to be solved] [means of solving problems] [effects of invention] ...	【発明の属する技術分野】本発明は、土壤改良方法とそ... 【従来の技術】圃場の表面がまだ積雪に覆われている状... 【発明が解決しようとする課題】心土破碎は通常春先に... 【課題を解決するための手段】述のような目的達成す... 【発明の効果】以上の説明から明らかなように、本発明... ...
<Explanation of Drawing> [figure1] ...	【図1】本発明を施す圃場断面図である。 ...

Structure vs distribution

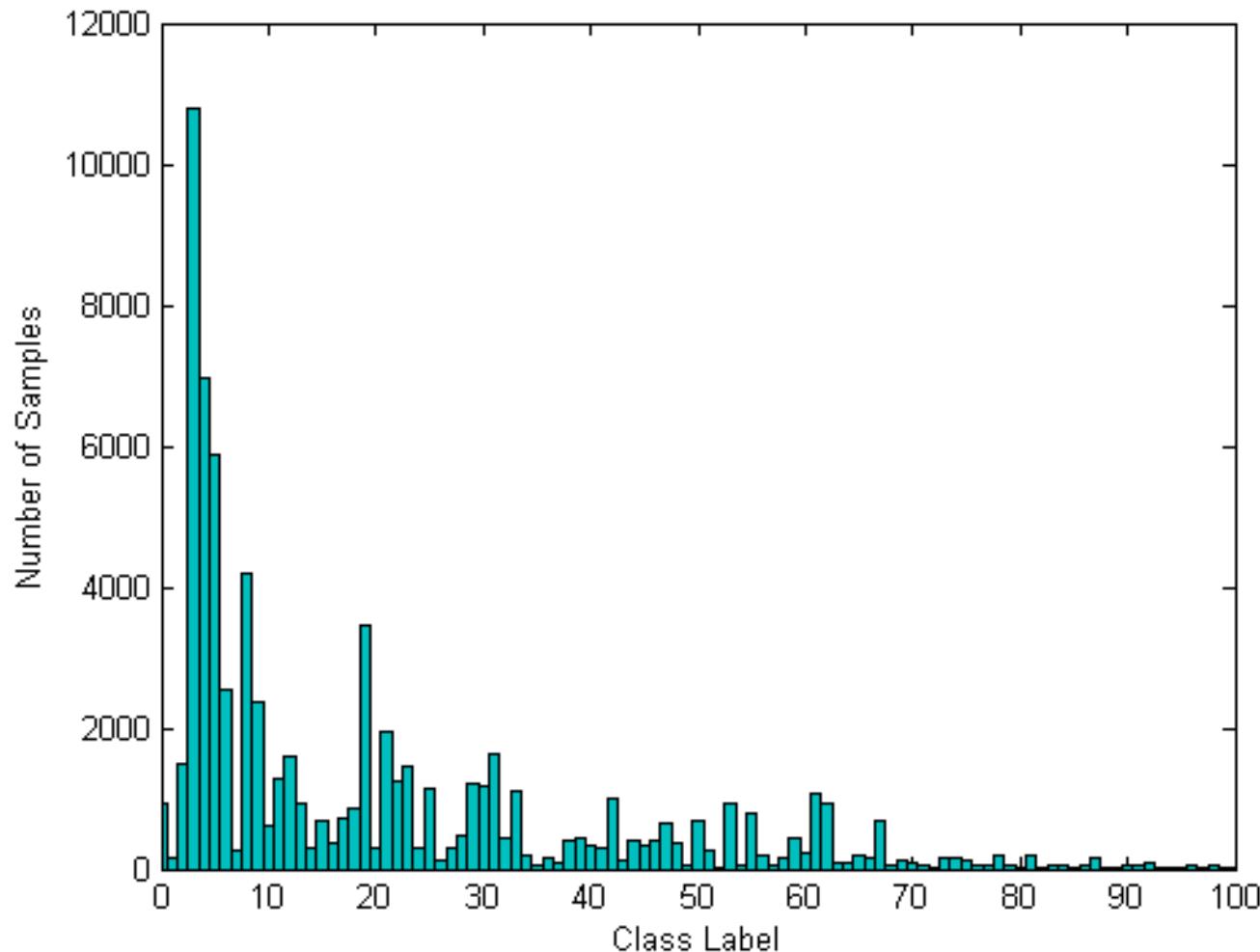


		Section	Class	Subclass	Group	Subgroup
No. Classes		8	120	630	7002	57913
No. Labels	Max	6	16	24	35	91
	Avg	1.3	1.5	1.7	2.2	2.7
No. Data	Max	857587	354104	176973	97008	23944
	Min	50540	38	1	1	1

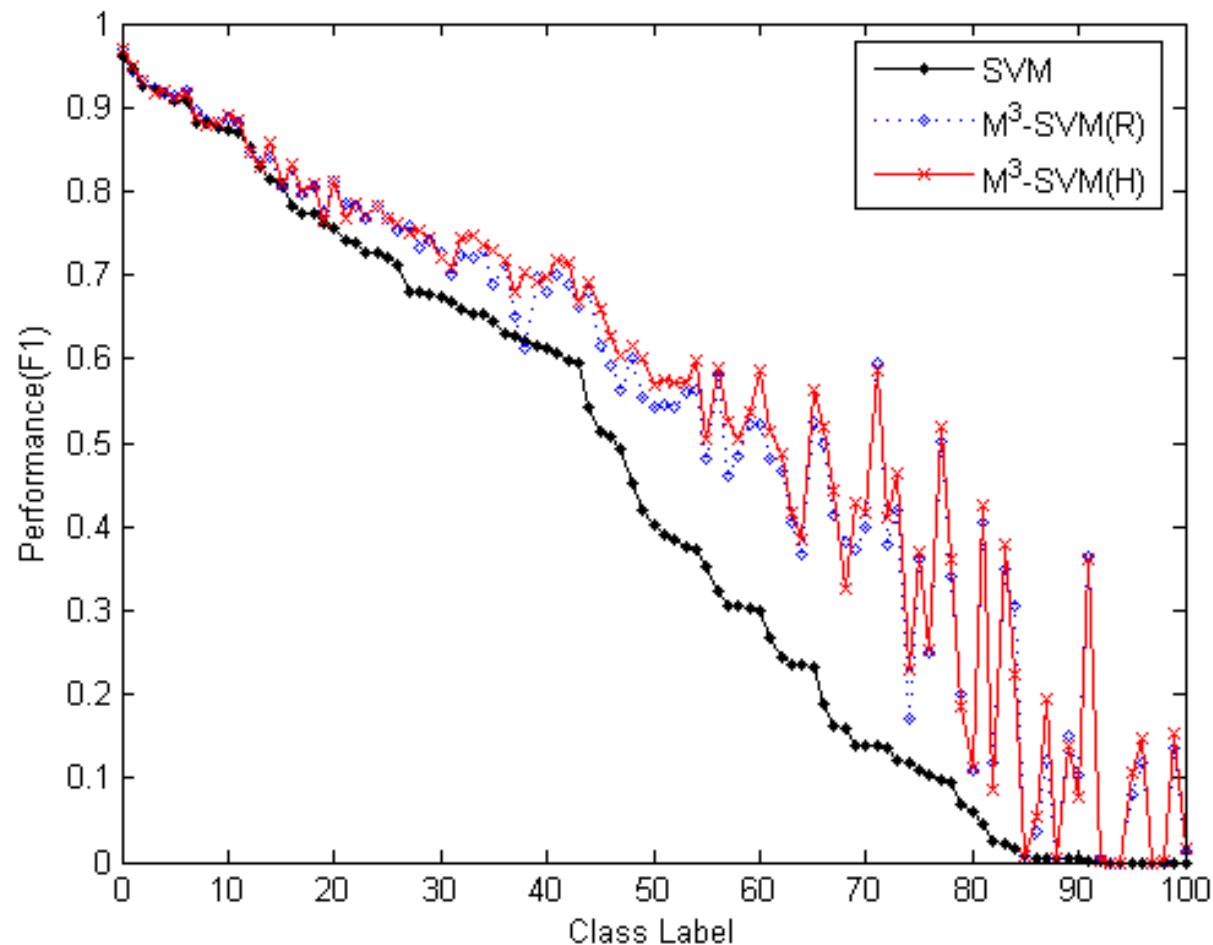
Text Categorization

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

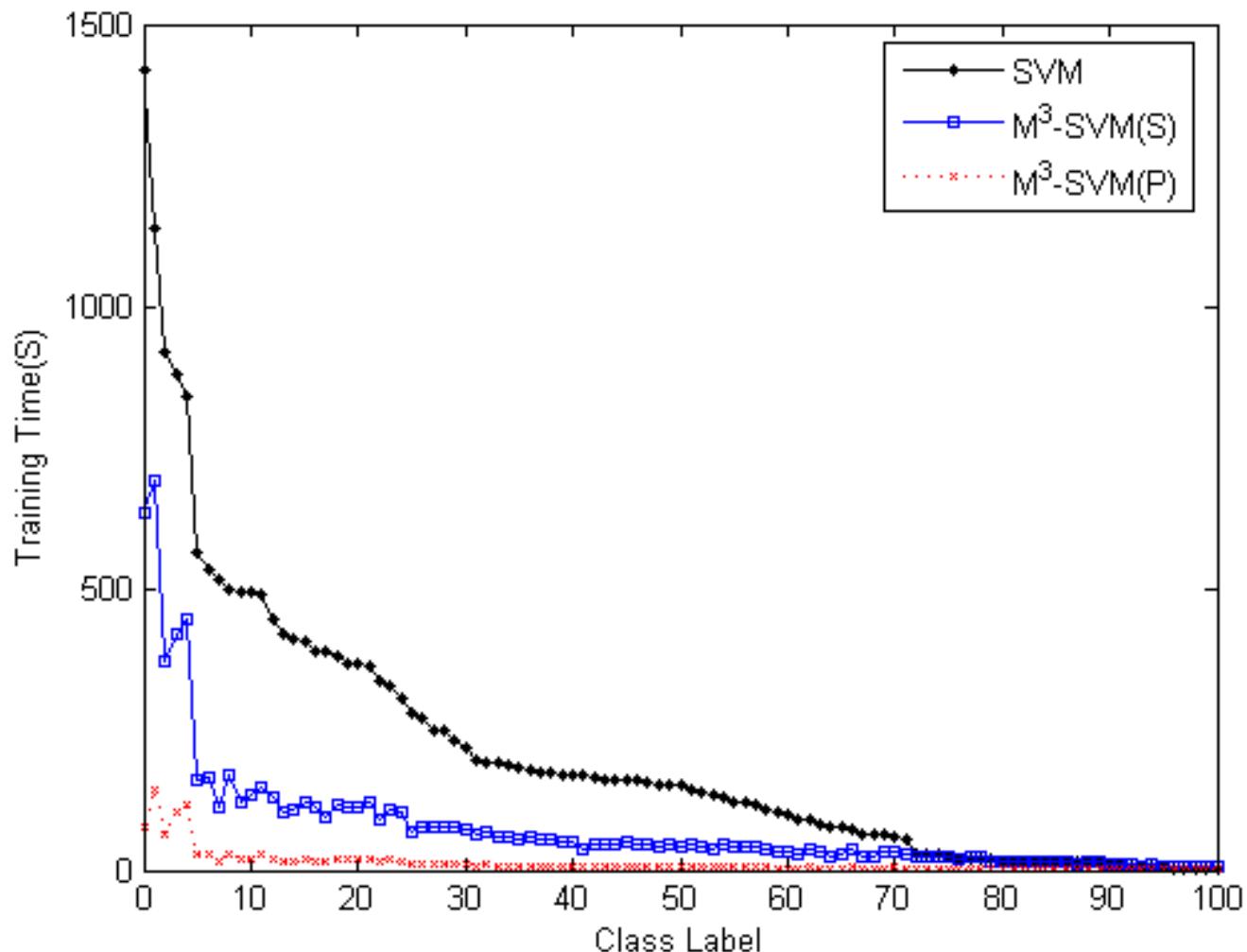
RCV1-V2: Data Distribution



RCV1-V2: Generalization Performance



RCV1-V2: Training Times



Gender Recognition

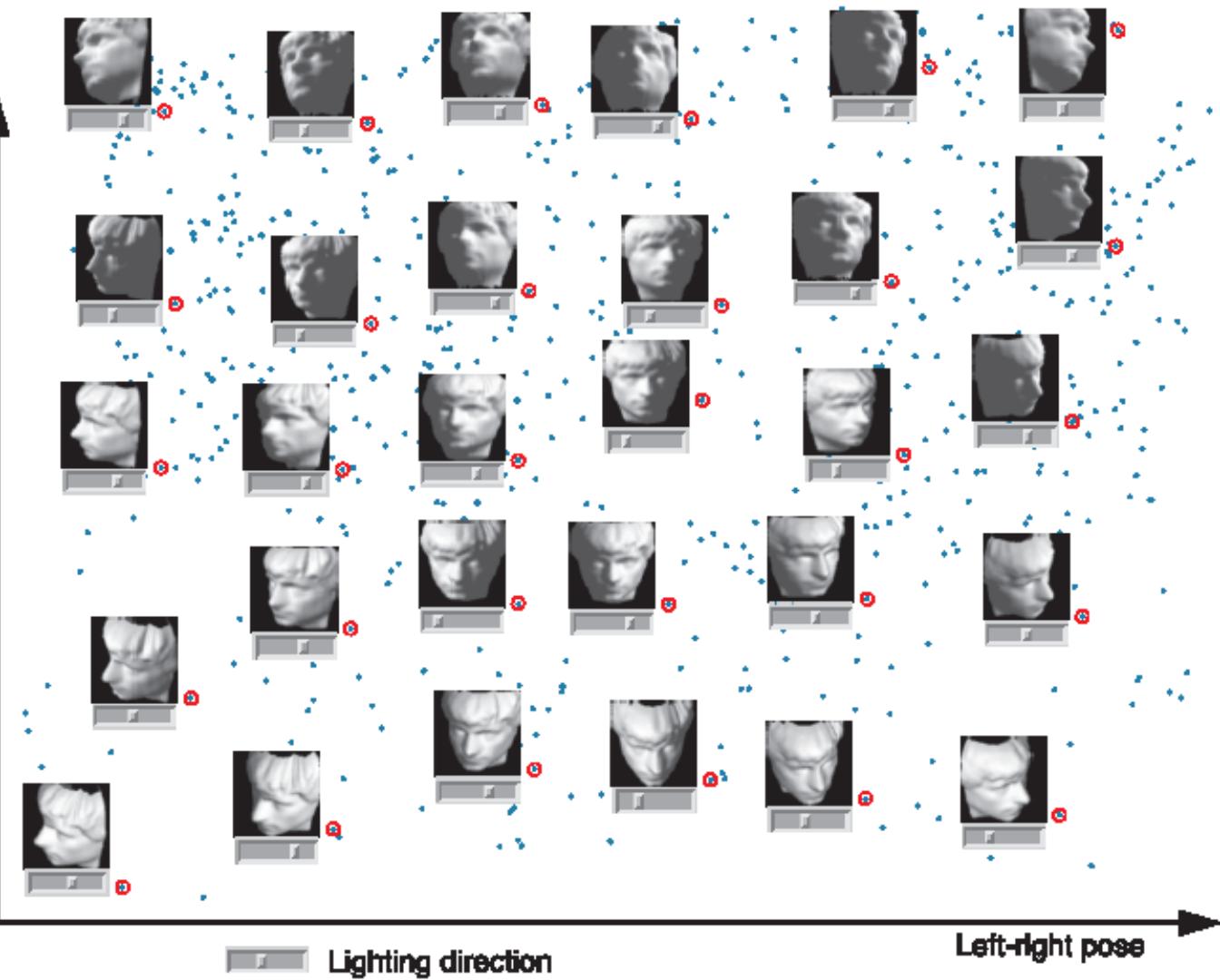
(H. C. Lian & B. L. Lu, 2005)

Multi-view Face Recognition



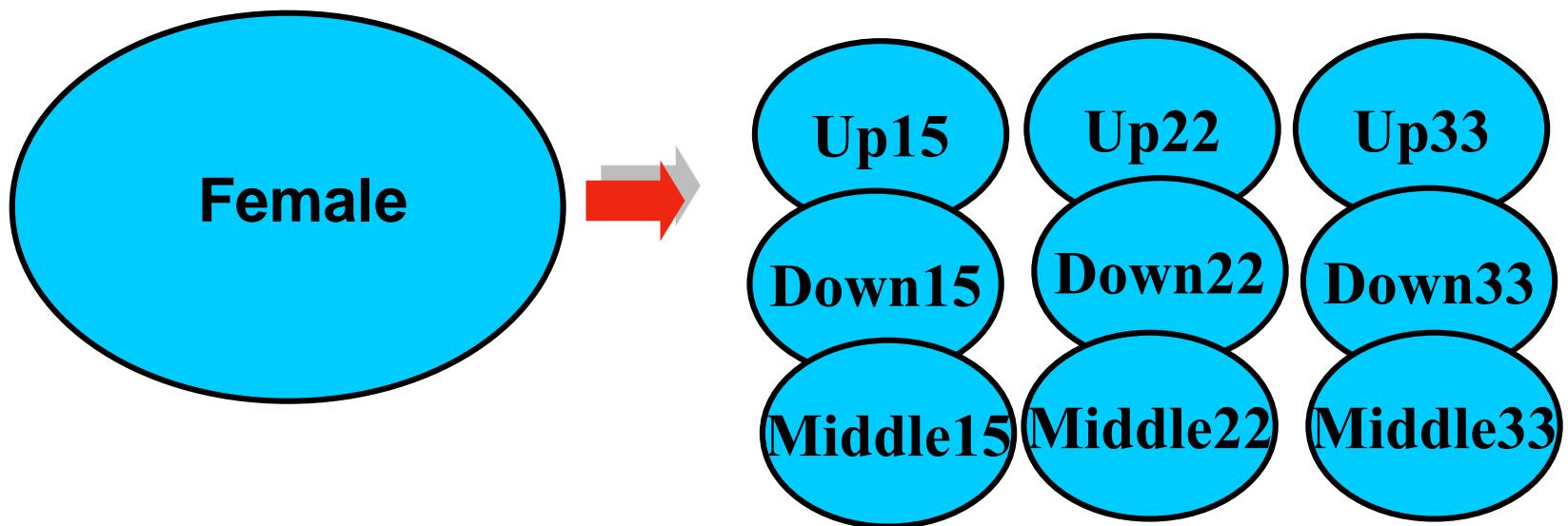
A

Up-down pose ↑

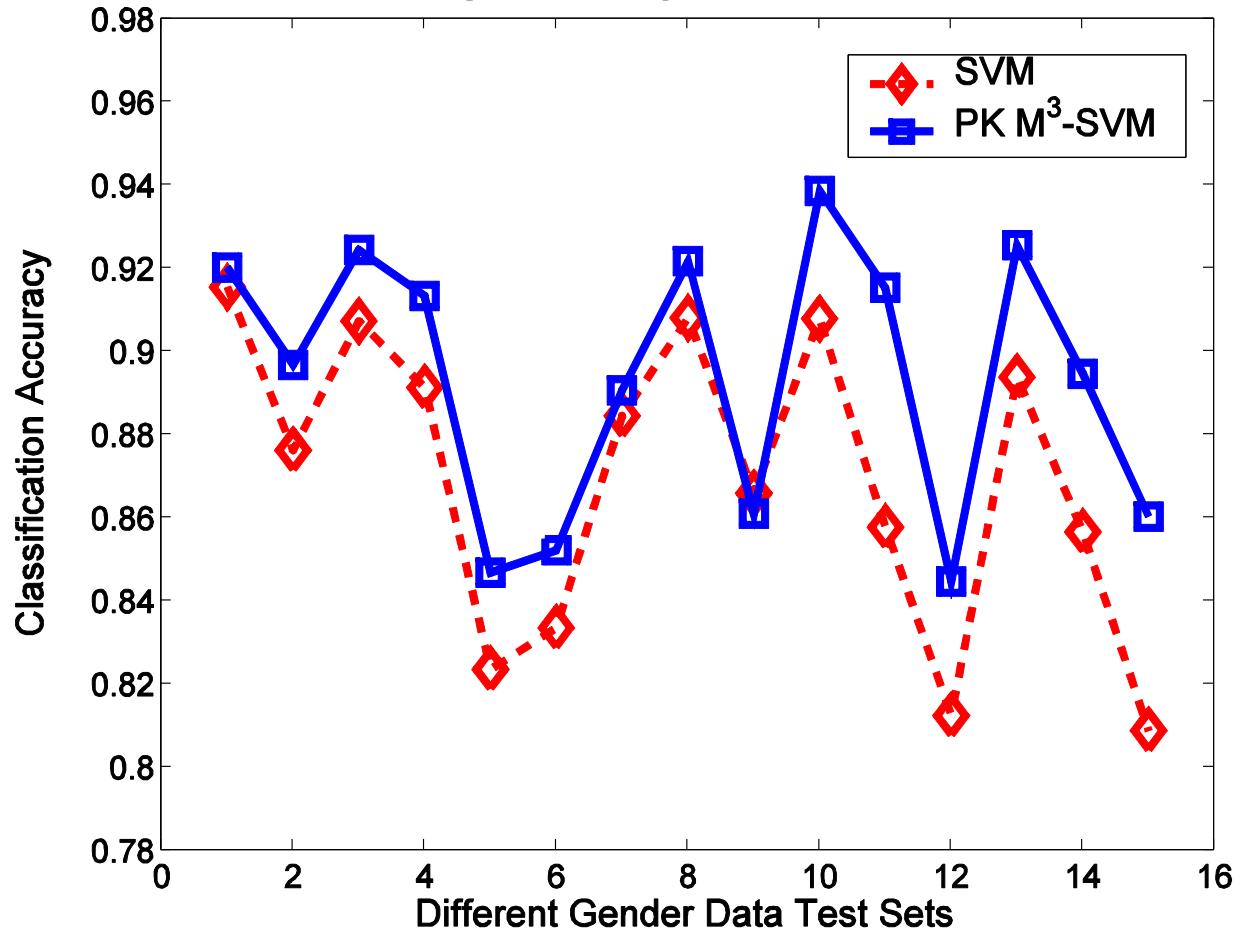


Task Decomposition

View information is used for task decomposition



Gender Recognition Using SVM and M³-SVM with PK

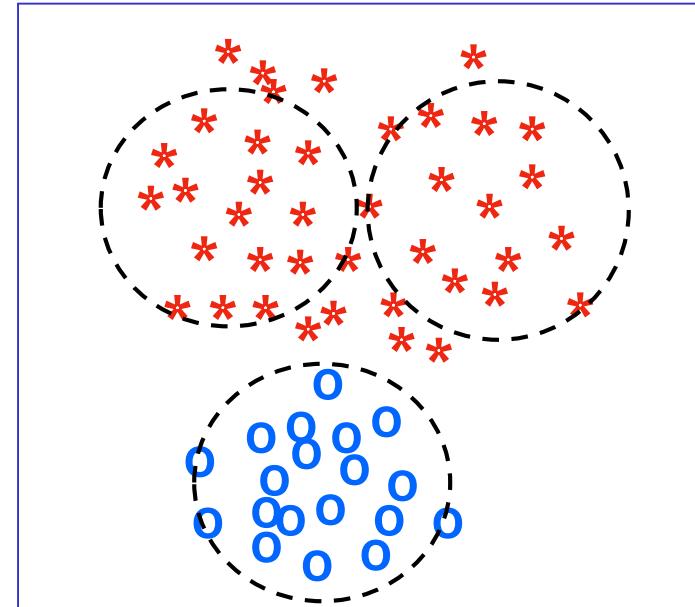
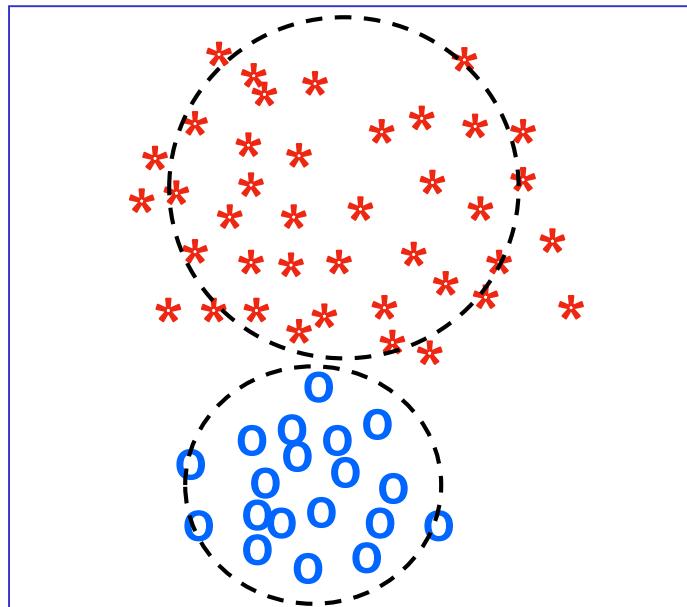


Gender Recognition Using a SVM With Equal Clustering

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

Equal Clustering

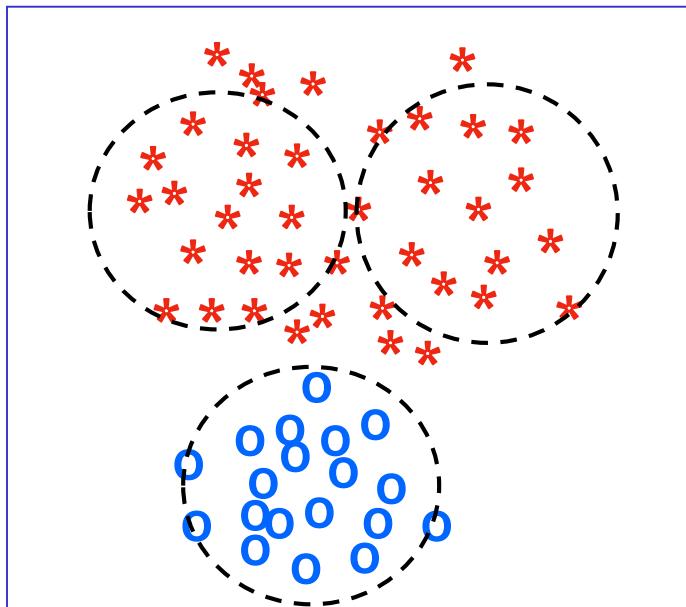
- Based on the algorithm “GeoClust”
(Choudhury, Nair and Keane, 2002)
- To generate spatially localized clusters that contain (nearly) equal number of samples to keep load balance.



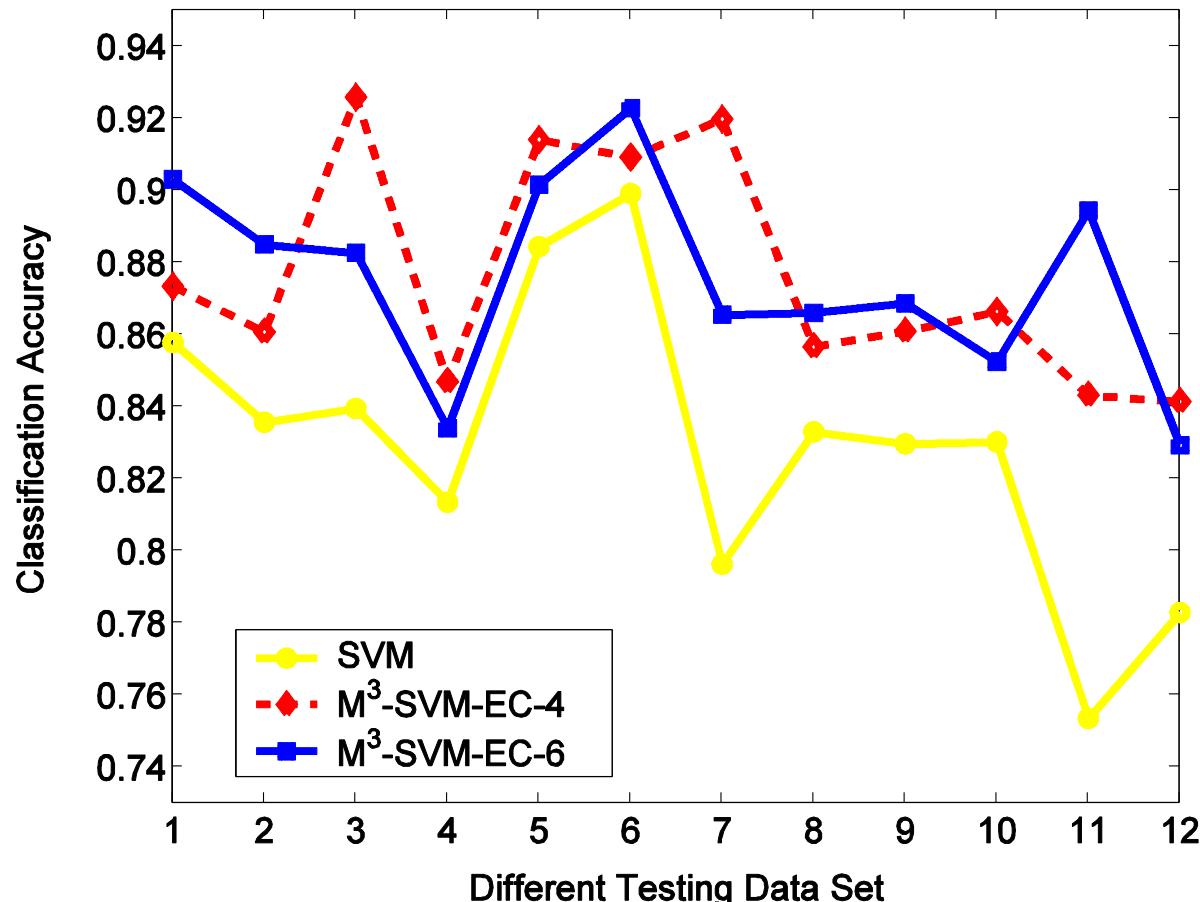
Basic Idea of Equal Clustering

- Solve an unconstrained nonlinear programming problem as follows:

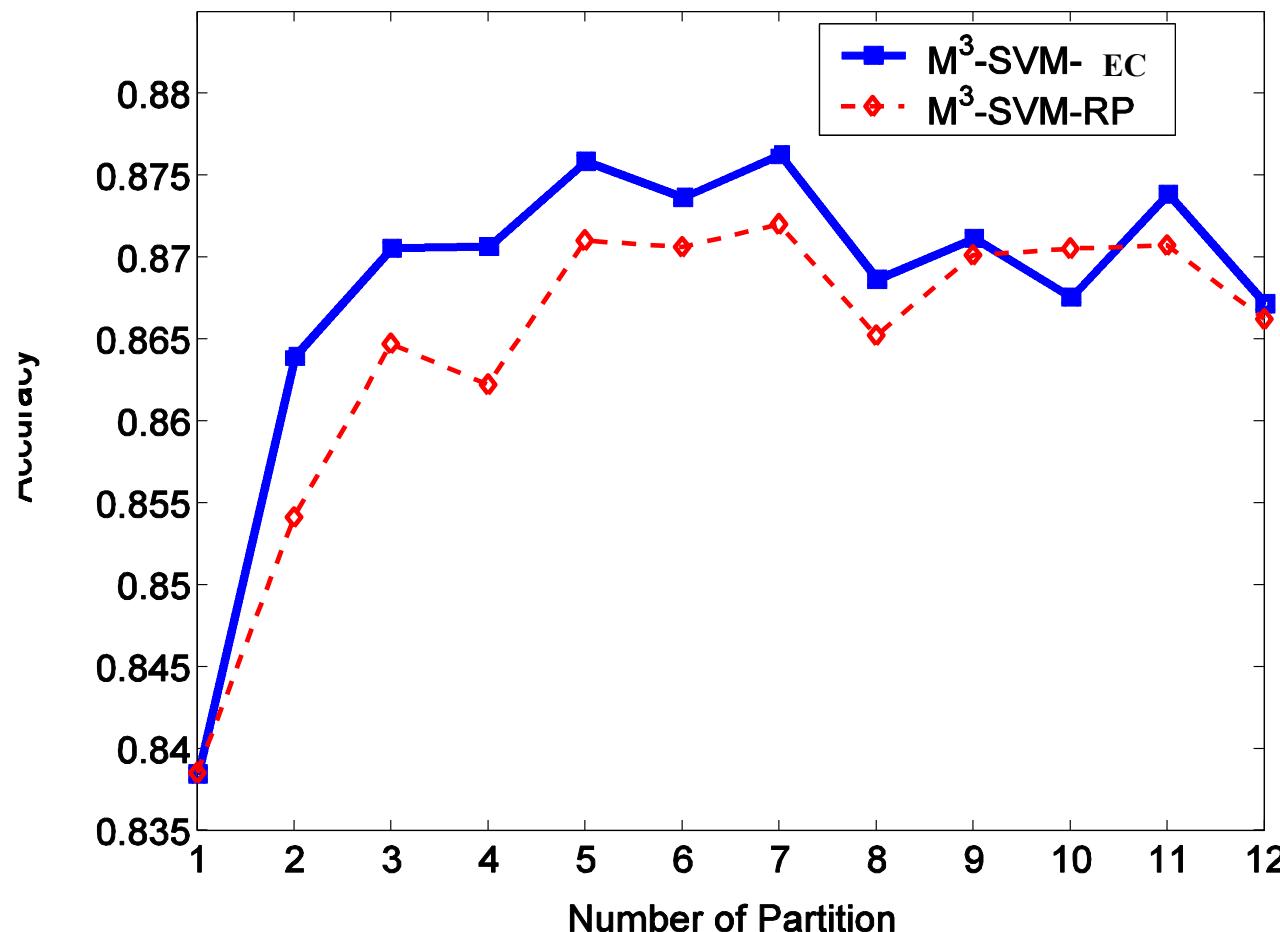
$$\text{Minimize}_{c_1, c_2, \dots, c_m} h = \max_{i=1}^m |W_i - \bar{W}|$$



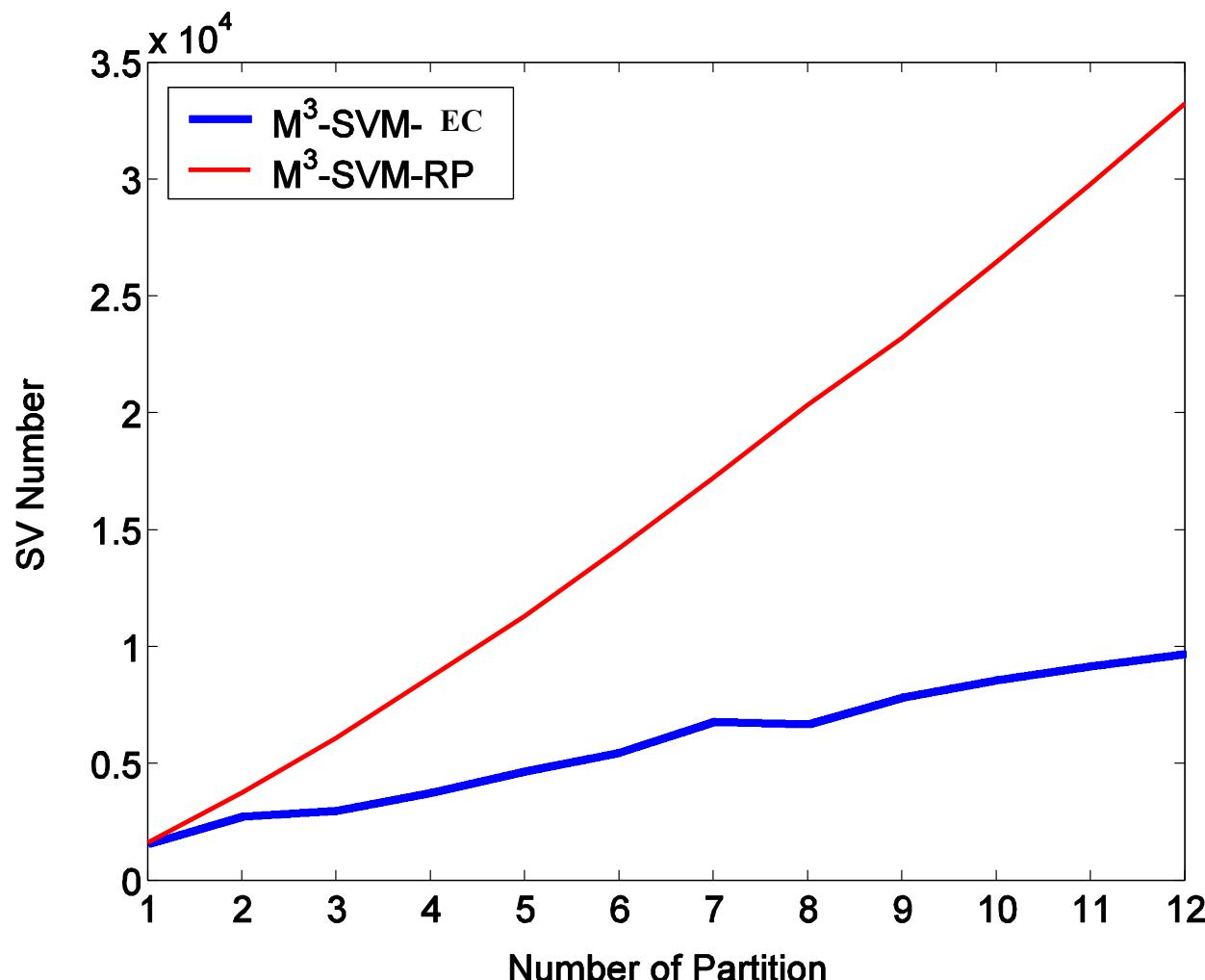
Gender Estimation on Peal dataset



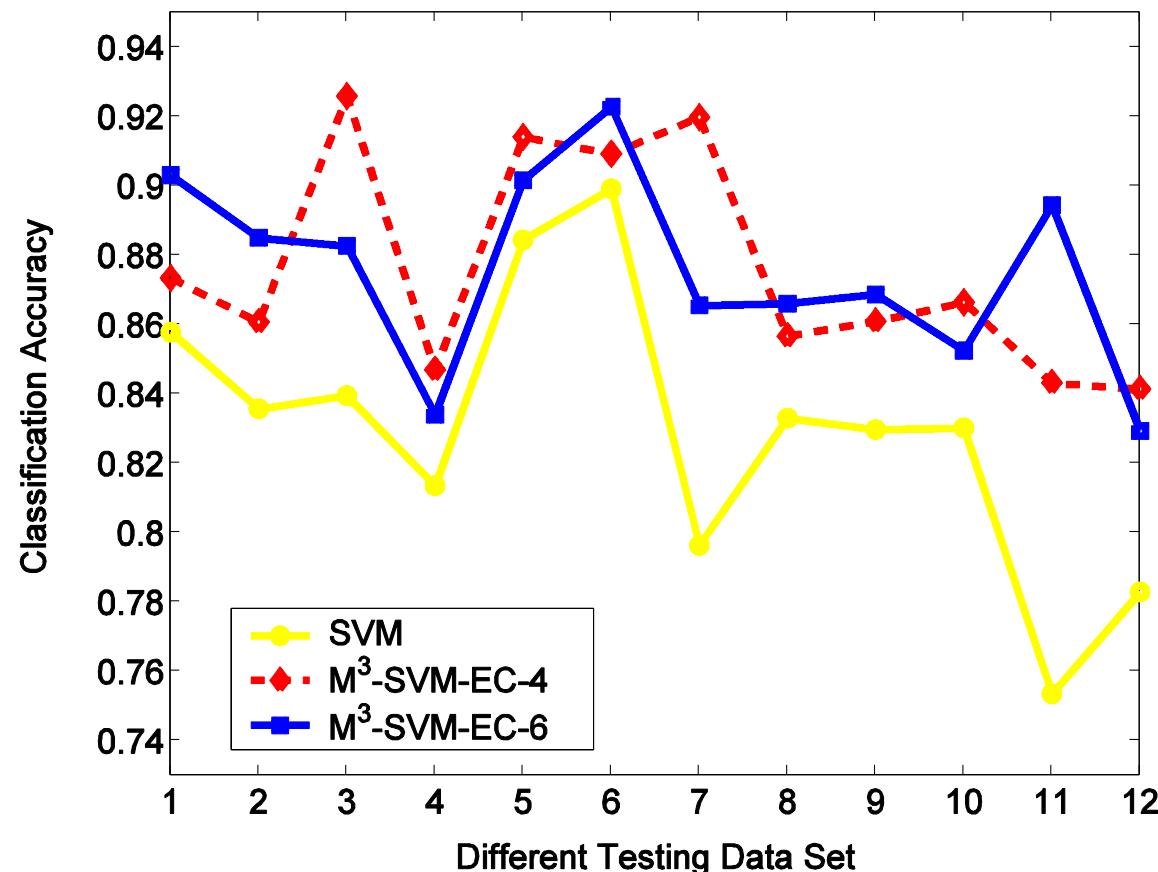
Comparison of Generalization Accuracy



Comparison of Number of SVs



Results of Gender Recognition



SVM software packages

□ LibSVM

- [Http://www.csie.ntu.edu.tw/~cjlin/libsvm/](http://www.csie.ntu.edu.tw/~cjlin/libsvm/)
- Chih-Chung Chang and Chin-Jen Lin

□ SVMlight

- <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

Outline of Transfer Learning

- Background
- Why Transfer Learning?
- Settings of Transfer Learning
- Approaches to Transfer Learning
- Negative Transfer
- Applications



Transfer of Learning

- A psychological point of view
- The study of dependency of human conduct, learning or performance on prior experience.
- [Thorndike and Woodworth, 1901] explored how individuals would transfer in one context to another context that share similar characteristics.

- C++ → Java
- Maths/Physics → Computer Science/Economics

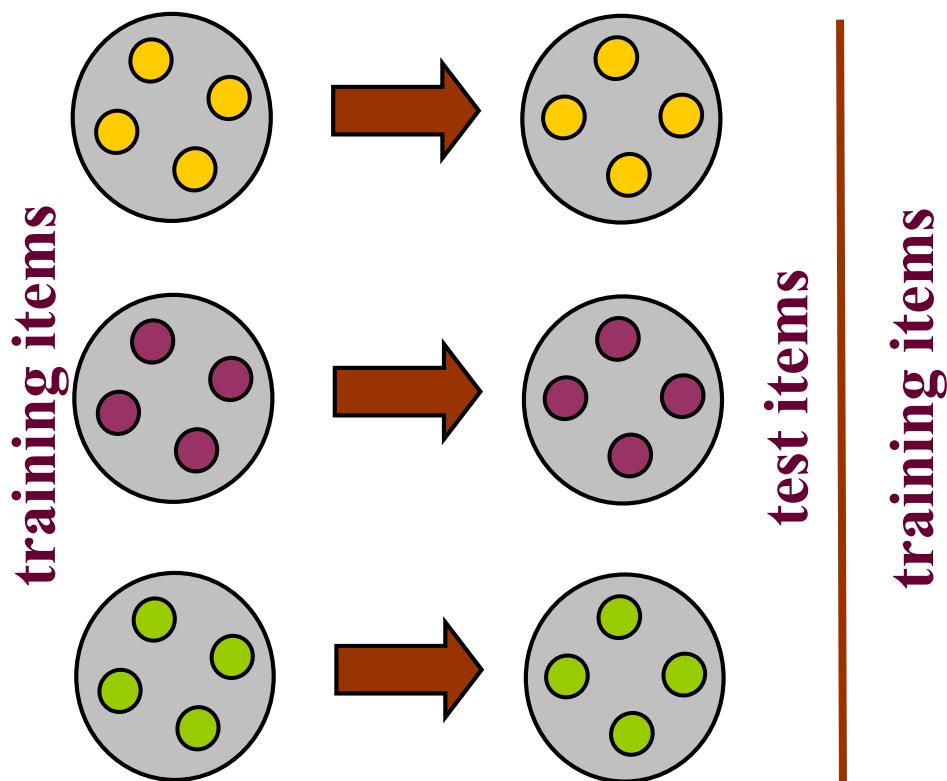
Thorndike and Woodworth, The Influence of Improvement in one mental function upon the efficiency of the other functions, 1901

Transfer Learning

- In the machine learning community
- The ability of a system to recognize and apply knowledge and skills learned in previous tasks to novel tasks or new domains, which share some commonality.
- Given a target task, how to identify the commonality between the task and previous (source) tasks, and transfer knowledge from the previous tasks to the target one?

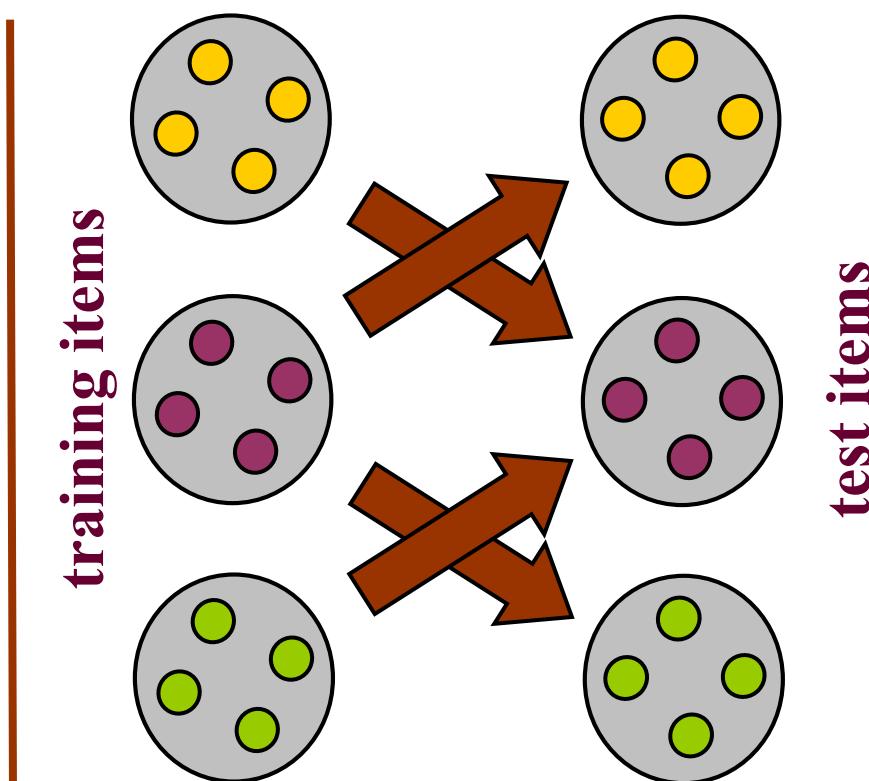
Traditional ML vs. TL

Traditional ML in
multiple domains



Humans can learn in many domains.

Transfer of learning
across domains



Humans can also transfer from one
domain to other domains.

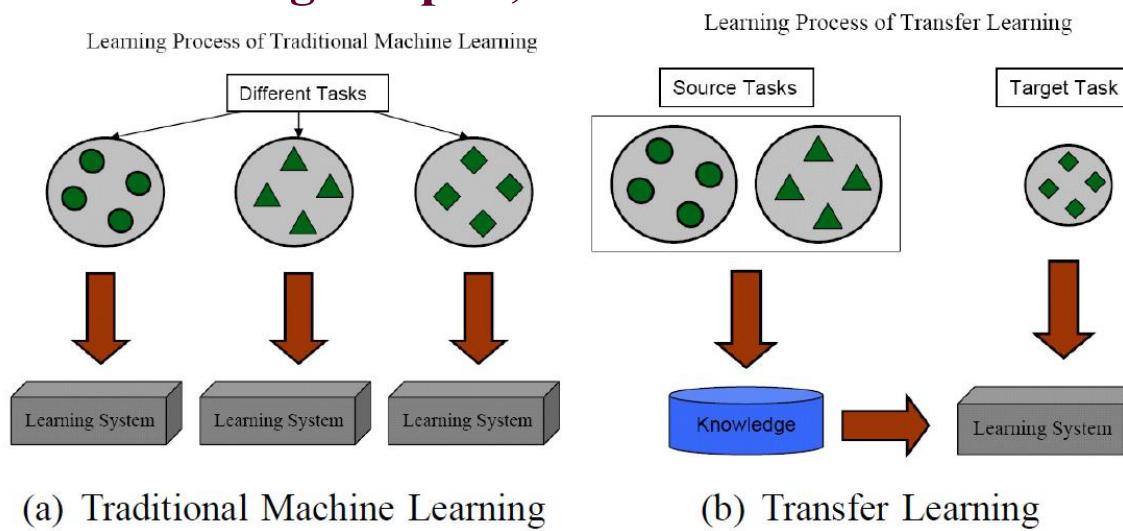
Traditional ML vs. TL

□ Traditional ML Assumptions

- Training and testing samples must be in the same feature distributions.
- Training samples must be enough.

□ Transfer Learning Conditions

- Source and target domains do not need to be in the same distributions.
- Less training samples, even none.



Notations

Domain:

It consists of two components: A feature space \mathcal{X} , a marginal distribution

$\mathcal{P}(X)$, where $X = \{x_1, x_2, \dots, x_n\} \in \mathcal{X}$

In general, if two domains are different, then they may have different feature spaces or different marginal distributions.

Task:

Given a specific domain and label space \mathcal{Y} . for each x_i in the domain, to predict its corresponding label y_i , where $y_i \in \mathcal{Y}$

In general, if two tasks are different, then they may have different label spaces or different conditional distributions

$\mathcal{P}(Y|X)$, where $Y = \{y_1, \dots, y_n\}$ and $y_i \in \mathcal{Y}$

Notations

For simplicity, we only consider at most two domains and two tasks.

Source domain:

$\mathcal{P}(X_S)$, where $X_S = \{x_{S_1}, x_{S_2}, \dots, x_{S_{n_S}}\} \in \mathcal{X}_S$

Task in the source domain:

$\mathcal{P}(Y_S|X_S)$, where $Y_S = \{y_{S_1}, y_{S_2}, \dots, y_{S_{n_S}}\}$ and $y_{S_i} \in \mathcal{Y}_S$

Target domain:

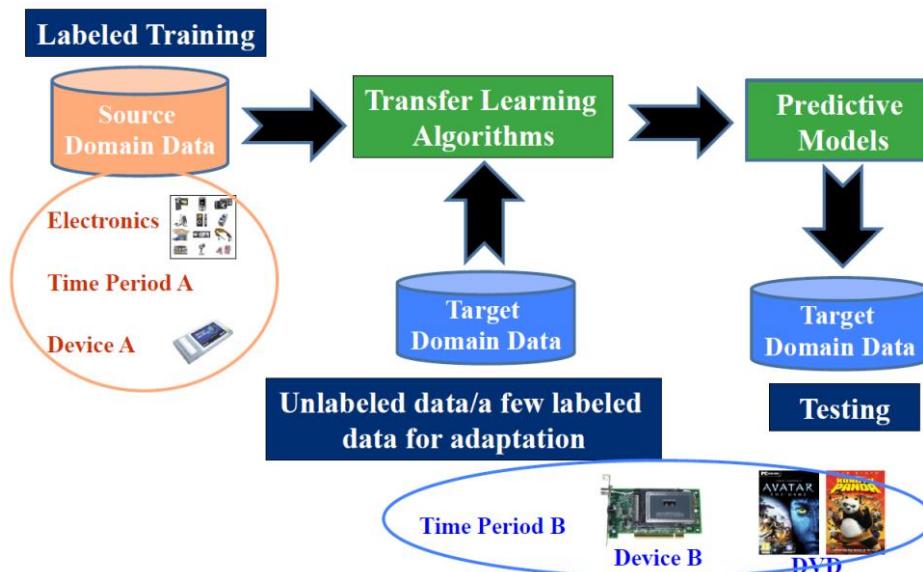
$\mathcal{P}(X_T)$, where $X_T = \{x_{T_1}, x_{T_2}, \dots, x_{T_{n_T}}\} \in \mathcal{X}_T$

Task in the target domain

$\mathcal{P}(Y_T|X_T)$, where $Y_T = \{y_{T_1}, y_{T_2}, \dots, y_{T_{n_T}}\}$ and $y_{T_i} \in \mathcal{Y}_T$

Why Transfer Learning?

- In some domains, labeled data are in short supply.
- In some domains, the calibration effort is very expensive.
- In some domains, the learning process is time consuming.
- How to extract knowledge learnt from related domains to help learning in a target domain with a few labeled data?
- How to extract knowledge learnt from related domains to speed up learning in a target domain?



ML: Yesterday, Today and Tomorrow



Qiang Yang. Transfer learning report.

An Example: Sentiment Classification

10 hours ago

Edward Priz replied:



You know, this isn't the first time that "States Rights" has been used as a cover for racist policies. In fact, the whole "States Rights" thing has become a sort of code for heavy-handed racist policies, hasn't it? And it does provide a sort of contextual link with those heroic days when evil was confronted in places like Selma and Little Rock, doesn't it? Thanks for making that link explicit.

10 hours ago

RICH HIRTH replied:



The issue here is probable cause. A police officer can question if he has probable cause, and he can document it. This law can be abused if being Latino is probable cause. That is license to harass for the police. As long as the law is applied fairly there should not be a problem. As far as documentation, Most states have laws that citizens must carry valid state ID, and no one cares. There is no reason the Executive branch needed to get involved in what the Court should decide.

2 hours ago

Julia Gomez replied:



The Arizona law is so clearly unconstitutional that I do not think it will ever reach the point of being enforced. The article did not say so, but the Republican governor is afraid of a GOP primary electorate that is even more reactionary than usual. That is why she signed the bill, not because she thinks it is legally defensible.



Sentiment Classification



Sentiment Classifier

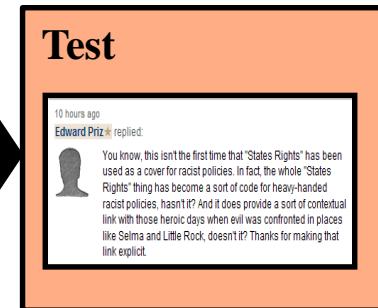


Classification Accuracy

~ 84.6%



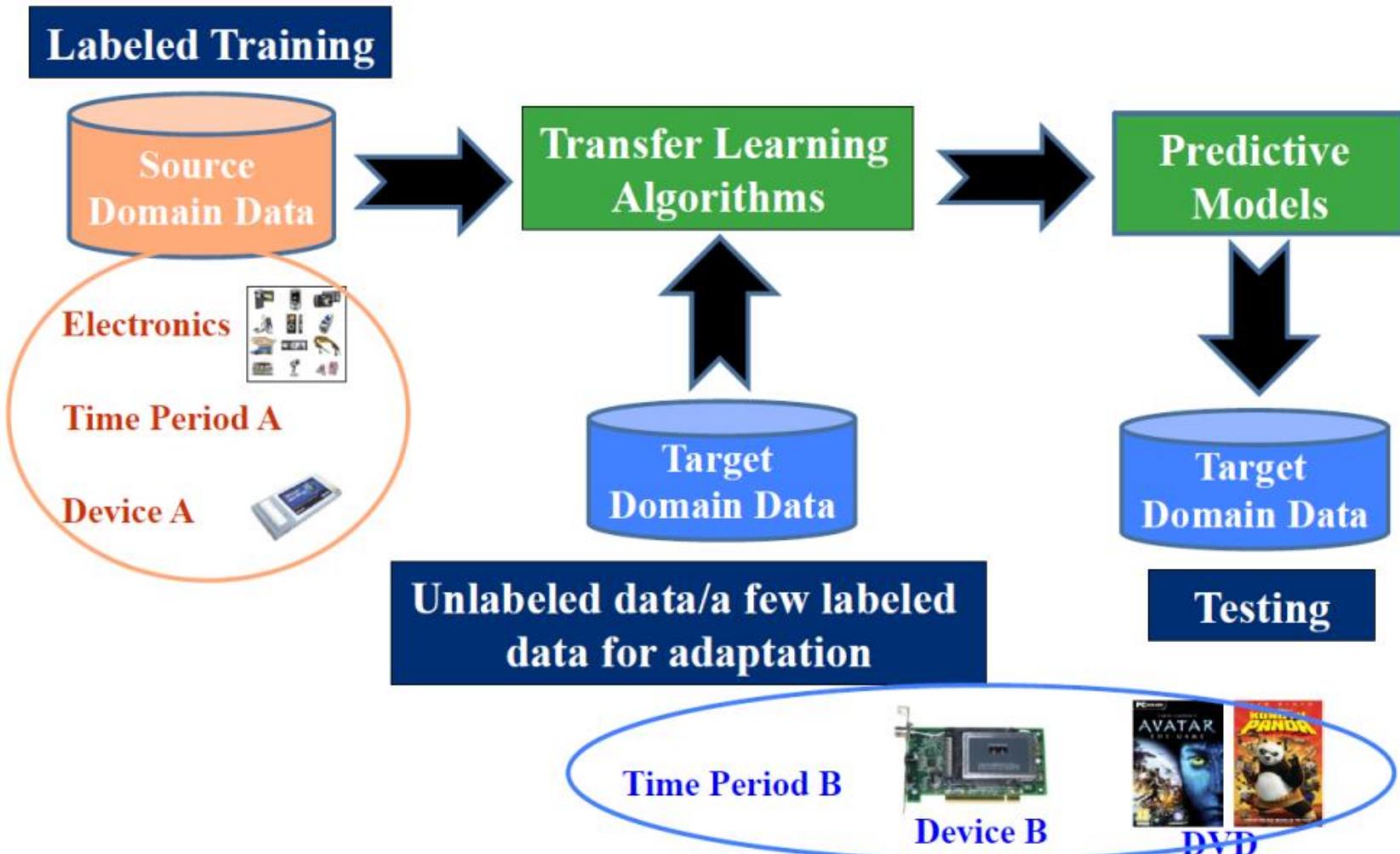
Sentiment Classifier



~72.65%



Framework of Transfer Learning?



Settings of Transfer Learning

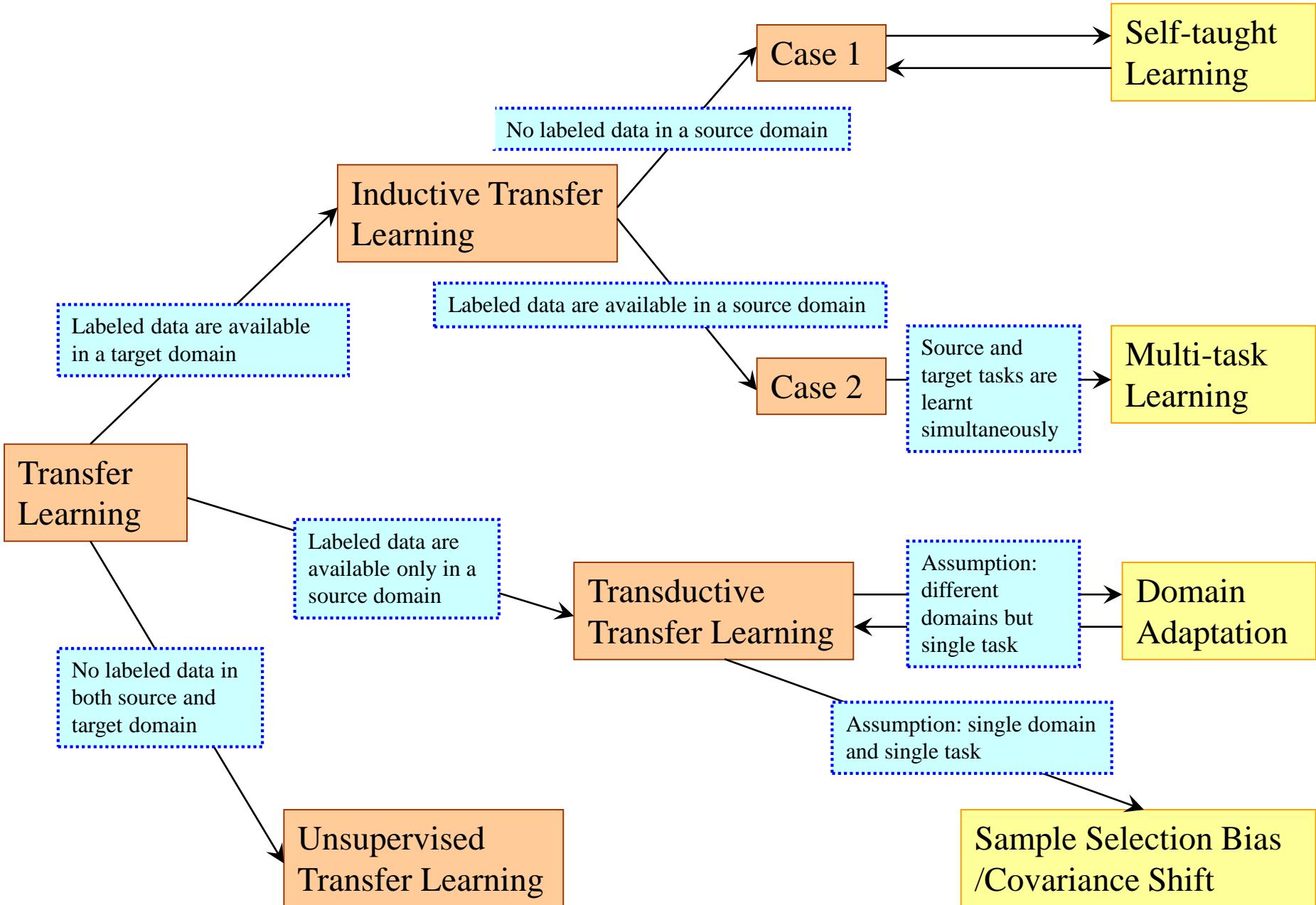
Relationship between Traditional Machine Learning and Various Transfer Learning Settings

Learning Settings		Source and Target Domains	Source and Target Tasks
Traditional Machine Learning		the same	the same
Transfer Learning	<i>Inductive Transfer Learning /</i>	the same	different but related
	<i>Unsupervised Transfer Learning</i>	different but related	different but related
	<i>Transductive Transfer Learning</i>	different but related	the same

Different Settings of Transfer Learning

Transfer Learning Settings	Related Areas	Source Domain Labels	Target Domain Labels	Tasks
<i>Inductive Transfer Learning</i>	Multi-task Learning	Available	Available	Regression, Classification
	Self-taught Learning	Unavailable	Available	Regression, Classification
<i>Transductive Transfer Learning</i>	Domain Adaptation, Sample Selection Bias, Co-variate Shift	Available	Unavailable	Regression, Classification
<i>Unsupervised Transfer Learning</i>		Unavailable	Unavailable	Clustering, Dimensionality Reduction

Pan S J, Yang Q. A survey on transfer learning , IEEE Transactions on knowledge and data engineering, 2010, 22(10): 1345-1359.



Transfer Component Analysis

Although the distributions of source domain and target domain in high dimensional space are different, we may find a low dimensional manifold space where the distributions of both domains are similar.

Transfer component analysis (TCA) tries to learn a set of common transfer components underlying both the source domain and the target domain. When projected to this subspace, the difference of feature distributions of both domains can be reduced.

$$\begin{aligned} P(\phi(X_S)) &\approx P(\phi(X_T)) \\ P(Y_S|\phi(X_S)) &\approx P(Y_T|\phi(X_T)) \end{aligned}$$

Negative Transfer

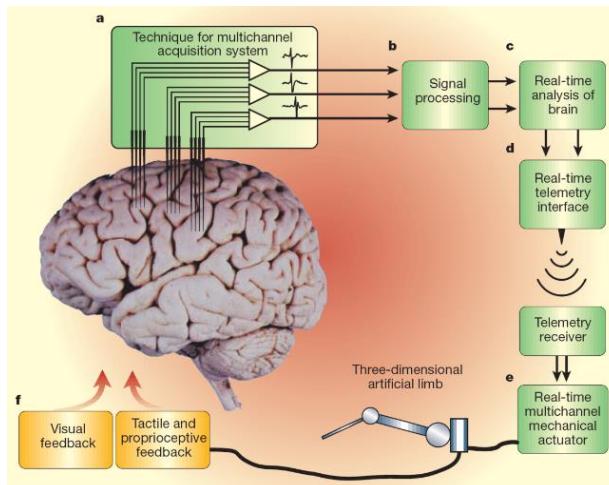
- Most approaches to transfer learning assume transferring knowledge across domains be always positive.
- However, in some cases, when two tasks are too dissimilar, brute-force transfer may even hurt the performance of the target task, which is called **negative transfer**.
- Some researchers have studied how to measure relatedness among tasks.
- How to design a mechanism to avoid negative transfer needs to be studied theoretically.

Personalizing EEG-based Affective Models with Transfer Learning

Wei-Long Zheng and Bao-Liang Lu, Personalizing EEG-based Affective Models with Transfer Learning,
Proc. of the 25th International Joint Conference on Artificial Intelligence (IJCAI-16).

Affective Brain-Computer Interface (aBCI)

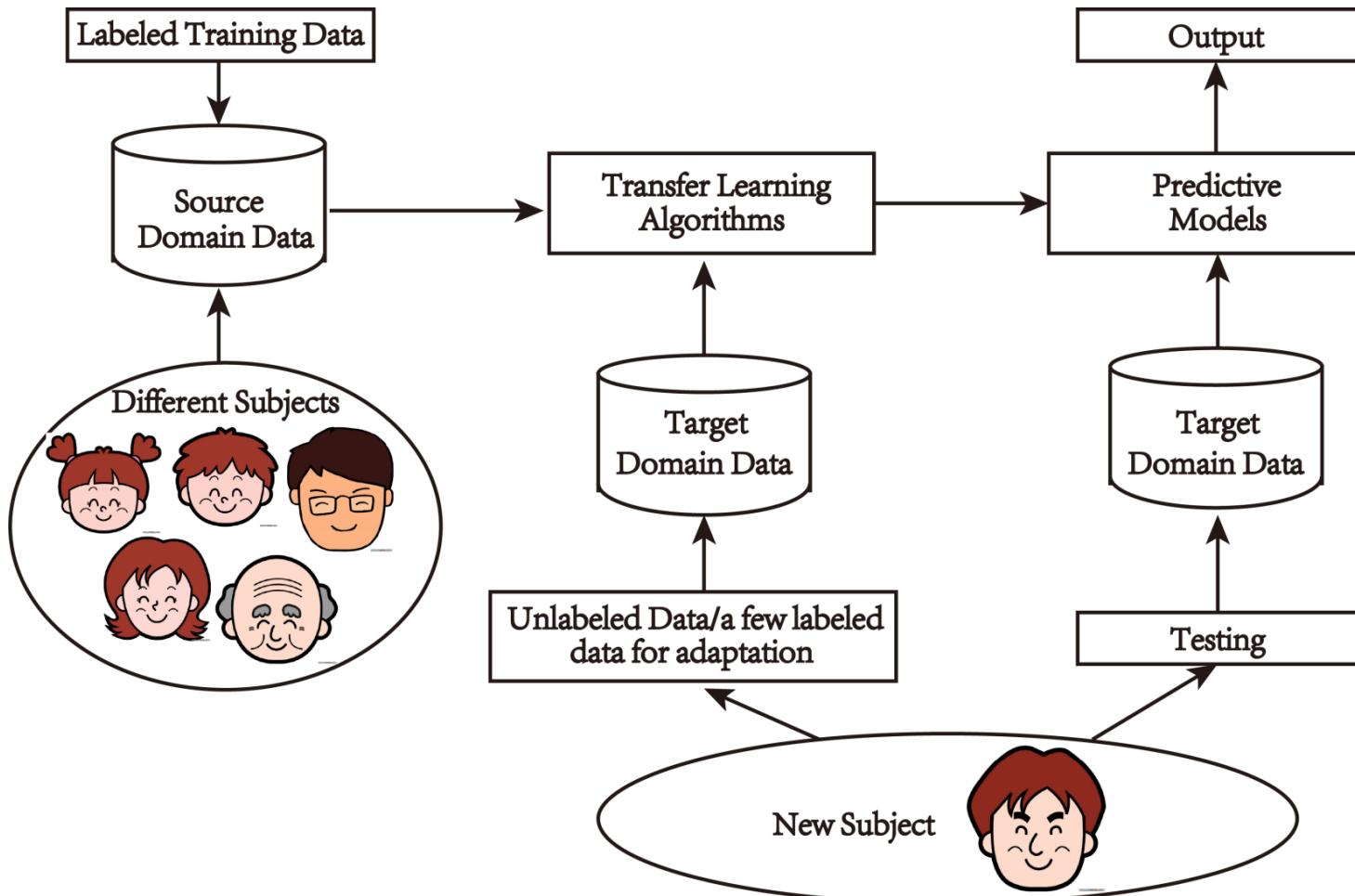
- A brain-computer interface (BCI) is a direct communication pathway between a brain and an external device.
- Affective Brain-Computer Interface (aBCI) aims to enhance brain-computer interface systems with the ability to detect, process, and respond to user's emotion.



Motivation of this work

- Most of the existing studies focus on training subject-specific affective models regarding individual differences.
 - In calibration phase, to acquire a large number of labeled data is time-consuming and annoying.
- Personalizing affective models for target subjects with knowledge transfer from labeled data of other subjects.

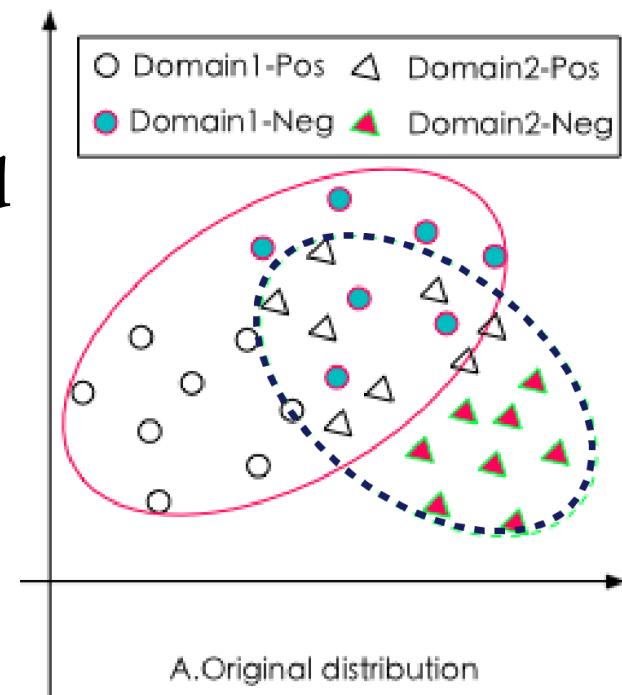
Transfer learning framework



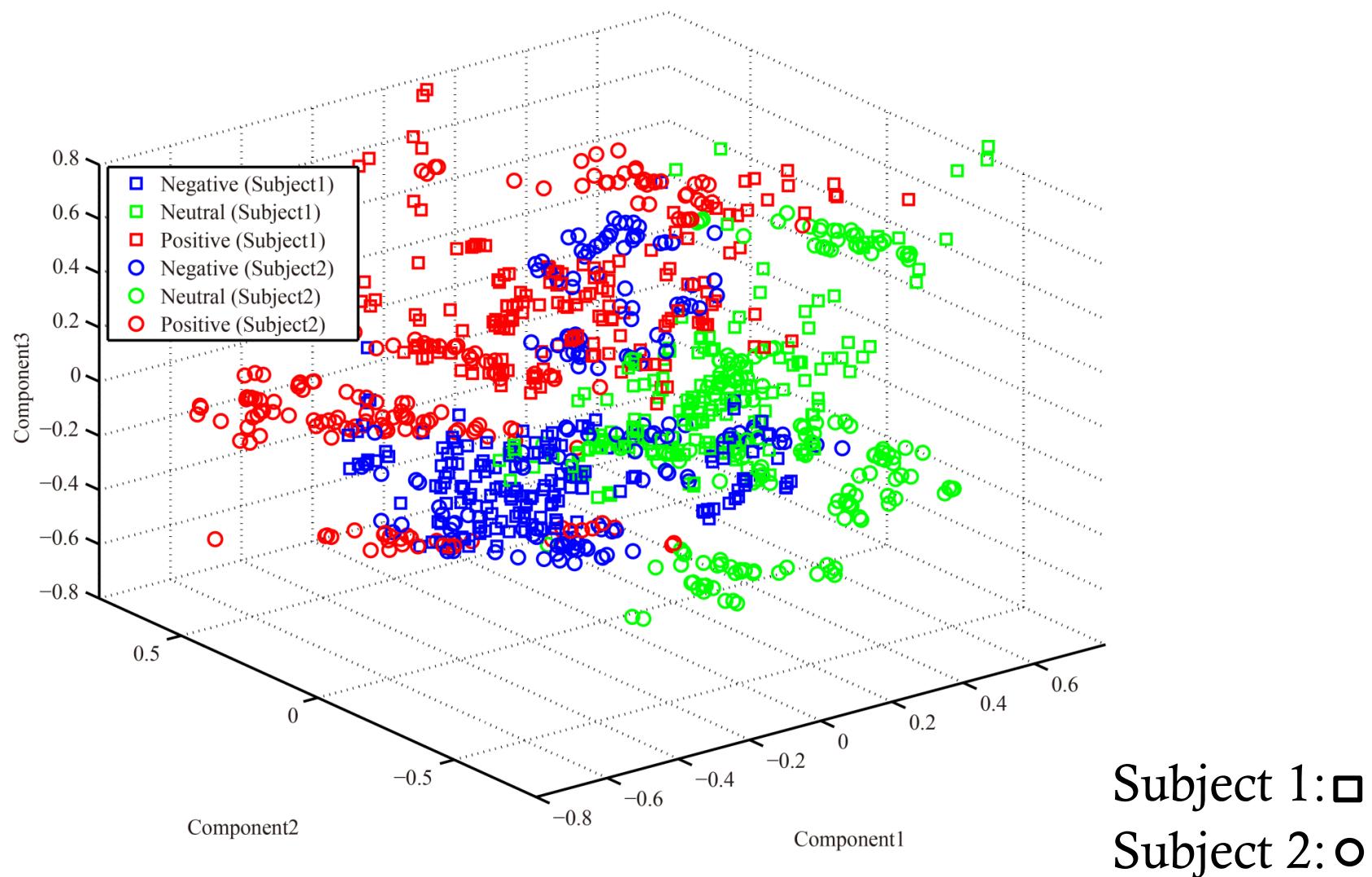
Challenges of Subject Transfer

The feature distributions of EEG data of source subjects and target subjects are not independently and identically distributed (i.i.d.) due to:

- The structural and functional variability between subjects
- The non-stationary nature of EEG signals
- The inherent changes of environmental variables



Challenges of Subject Transfer: An Example



Feature Reduction based Subject Transfer

Although the distributions of source domain and target domain in high dimensional space are different, we may find a low dimensional manifold space where the distributions of both domains are similar.

TCA and KPCA try to learn a set of common transfer components underlying both the source domain and the target domain. When projected to this subspace, the difference of feature distributions of both domains can be reduced.

$$\begin{aligned} P(\phi(X_S)) &\approx P(\phi(X_T)) \\ P(Y_S|\phi(X_S)) &\approx P(Y_T|\phi(X_T)) \end{aligned}$$

TCA-based Subject Transfer

TCA algorithm is firstly proposed by Pan et al. for cross-domain indoor WiFi localization and cross-domain text classification.

Algorithm 1 TCA-based Subject Transfer

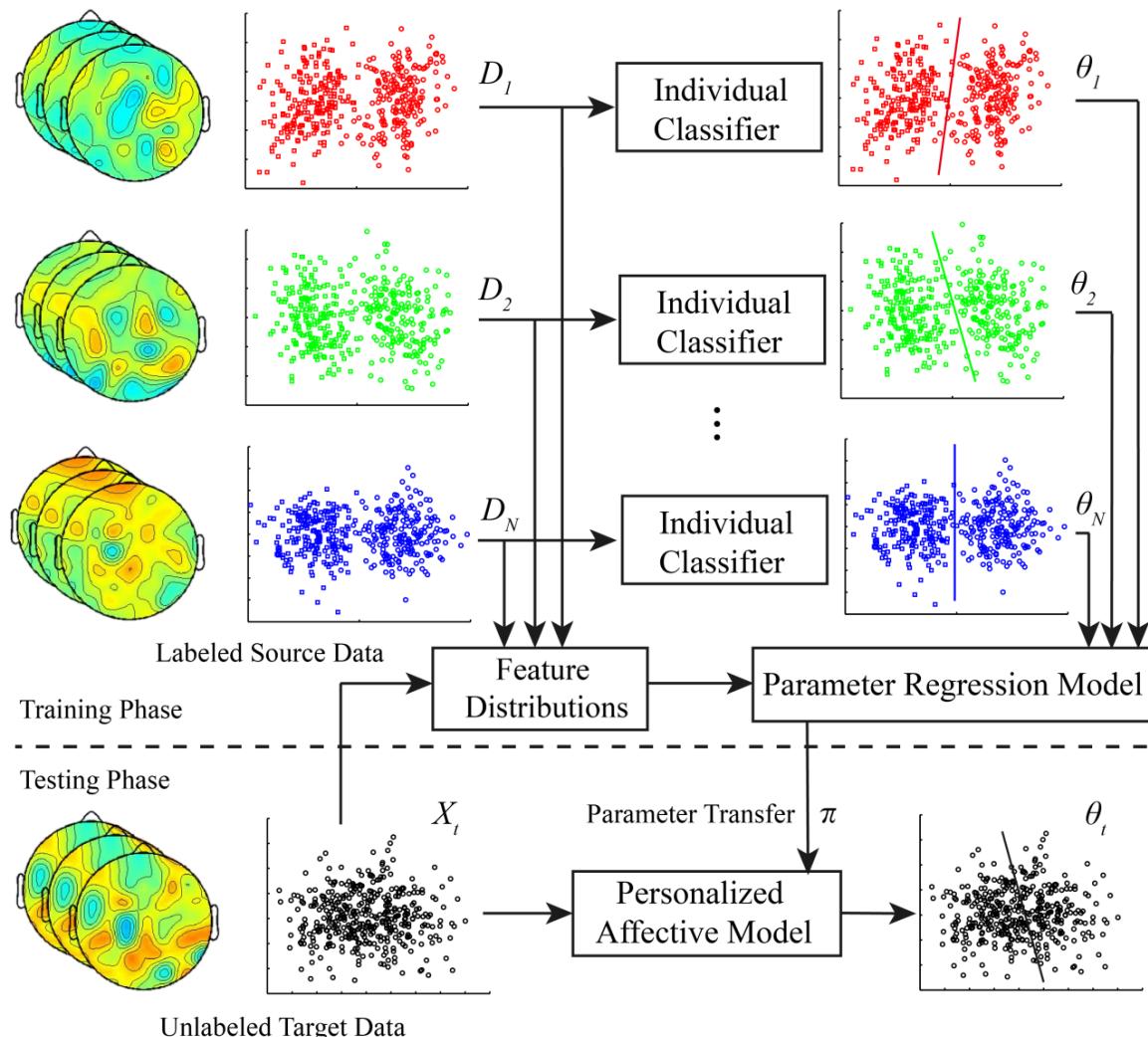
input : Source domain data set $\mathcal{D}_S = \{(x_{S_i}, y_{srci})\}_{i=1}^{n_1}$, and target domain data set $\mathcal{D}_T = \{x_{T_j}\}_{j=1}^{n_2}$.

output : Transformation matrix W .

- 1: Compute kernel matrix K from $\{x_{S_i}\}_{i=1}^{n_1}$ and $\{x_{T_j}\}_{j=1}^{n_2}$, matrix L , and the centering matrix H .
 - 2: Eigendecompose the matrix $(KLK + \mu I)^{-1}KLK$ and select the m leading eigenvectors to construct the transformation matrix W .
 - 3: **return** transformation matrix W .
-

Pan S J, Tsang I W, Kwok J T, et al. Domain adaptation via transfer component analysis [J]. IEEE Transactions on Neural Networks, 2011, 22(2): 199-210.

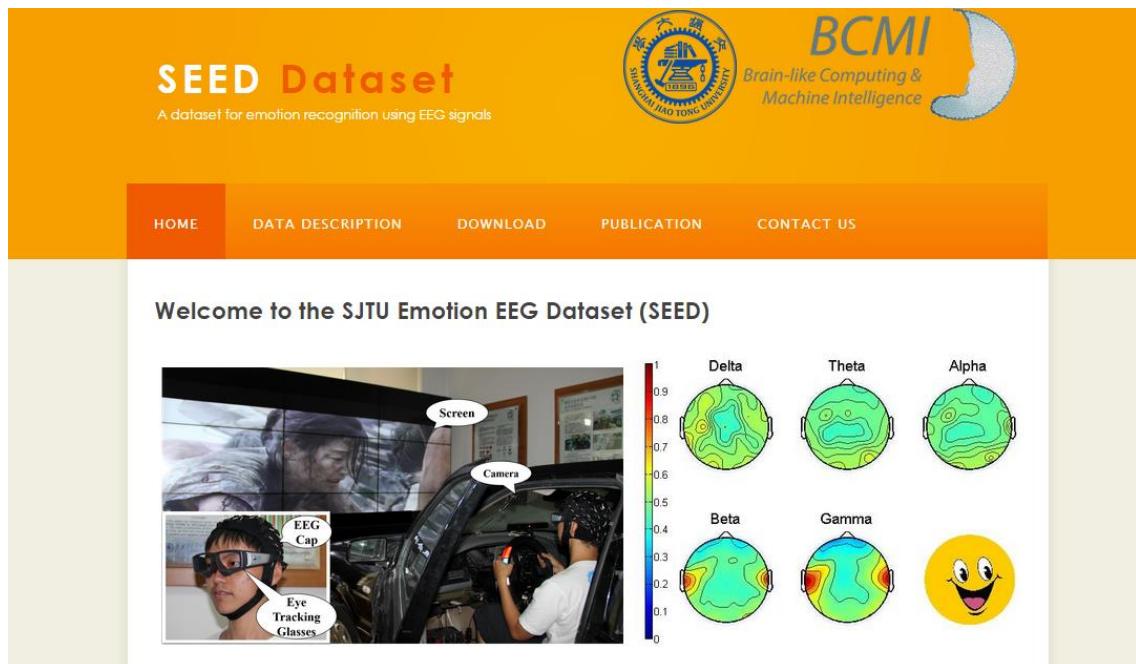
Transductive Parameter Transfer



Enver Sangineto, Gloria Zen, Elisa Ricci, and Nicu Sebe. We are not all equal: Personalizing models for facial expression analysis with transductive parameter transfer. In ACM International Conference on Multimedia, pages 357–366. ACM, 2014.

SJTU Emotion EEG Dataset (SEED)

- No. subjects: 15, No. EEG Channels: 62
- EEG Features: differential entropy in five frequency bands.
- Three emotions: positive, neutral and negative



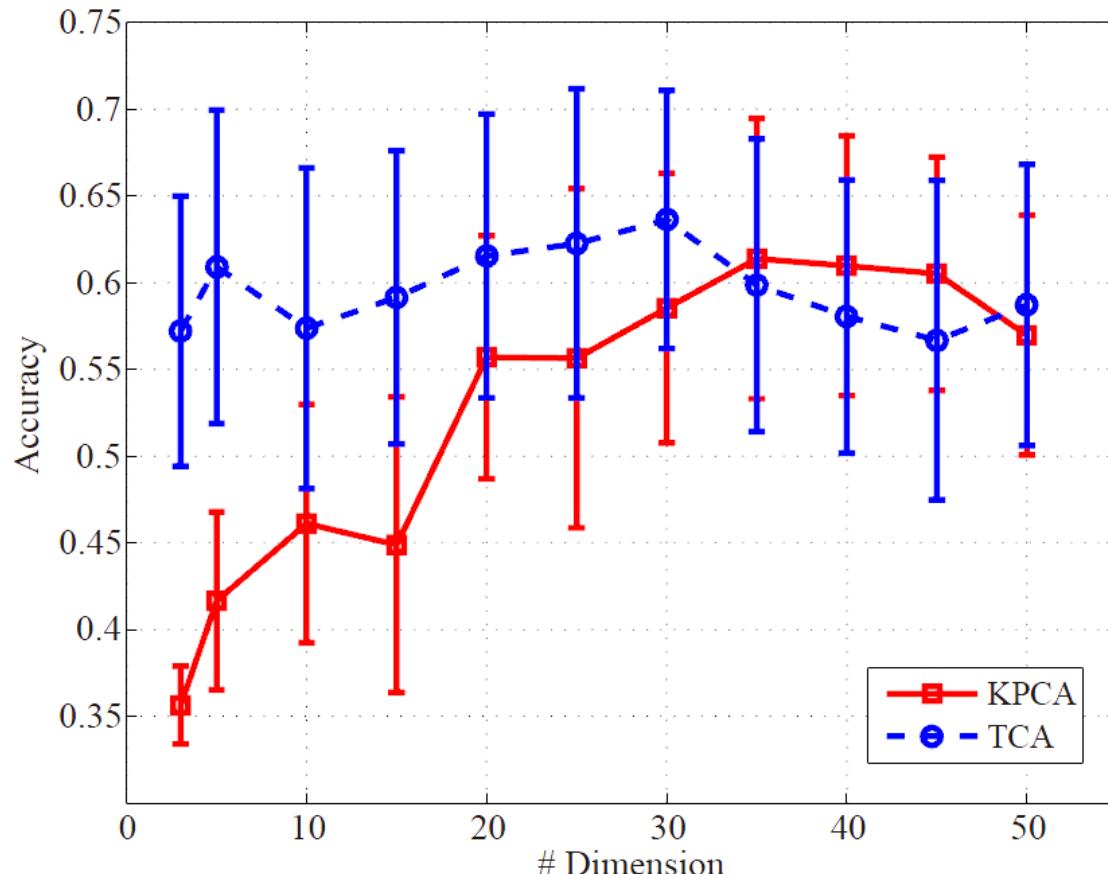
<http://bcmi.sjtu.edu.cn/~seed/>

"Investigating Critical Frequency Bands and Channels for EEG-based Emotion Recognition with Deep Neural Networks", Wei-Long Zheng, and Bao-Liang Lu, IEEE Transactions on Autonomous Mental Development (IEEE TAMD), 2015.

Evaluation Details

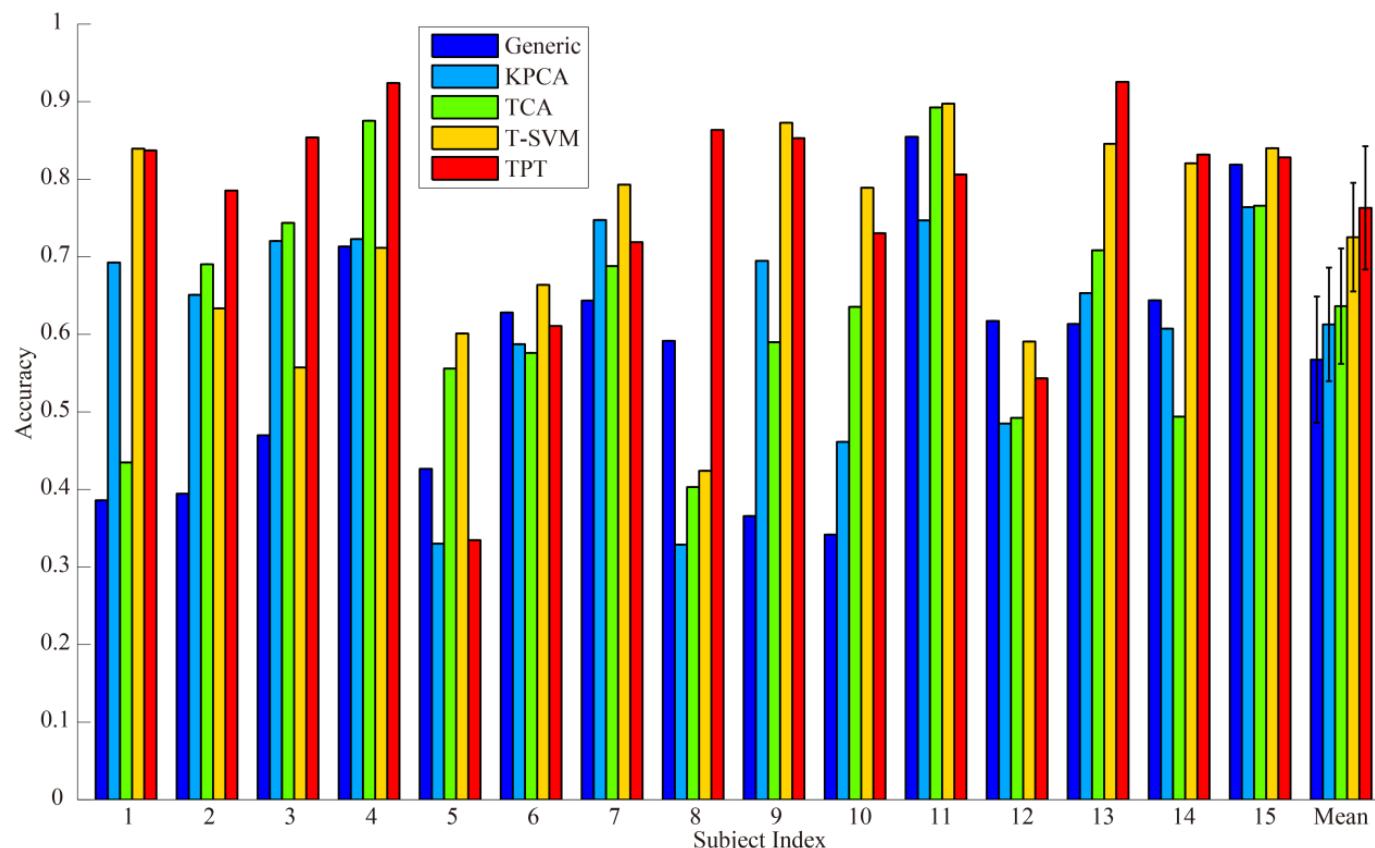
- **Leave-one-subject-out cross validation method**
- **Baseline method**
 - We concatenate data from all available subjects as training data and train a generic classifier with linear SVM.
- **Transductive SVM (T-SVM)**
 - It is developed to learn a decision boundary and maximize the margin with unlabeled data.
- **One vs. one strategy for multi-class classification**
- **All the algorithms are implemented in MATLAB.**

Experimental Results



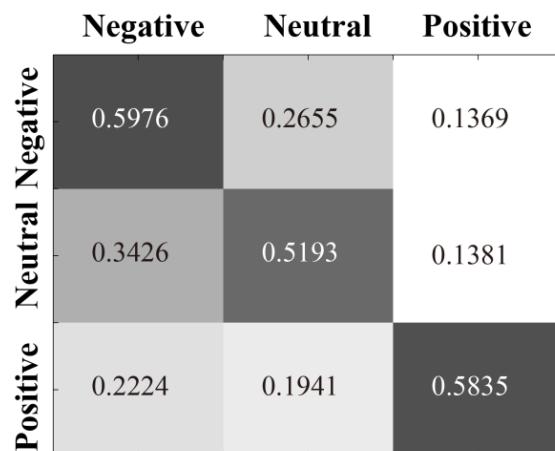
Comparison of KPCA and TCA approaches for different dimensionality of the subspace.

Experimental Results

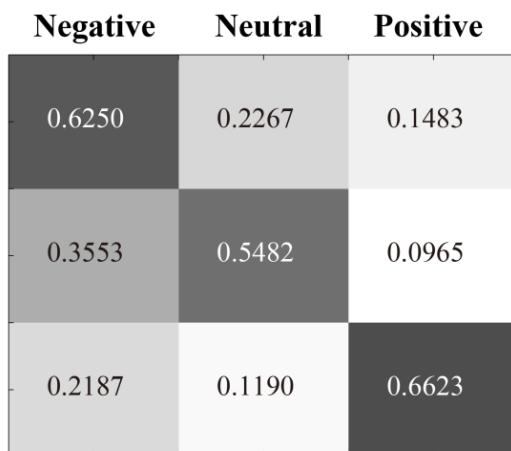


Stats.	Generic	KPCA	TCA	T-SVM	TPT
Mean	0.5673	0.6128	0.6364	0.7253	0.7631
Std.	0.1629	0.1462	0.1488	0.1400	0.1589

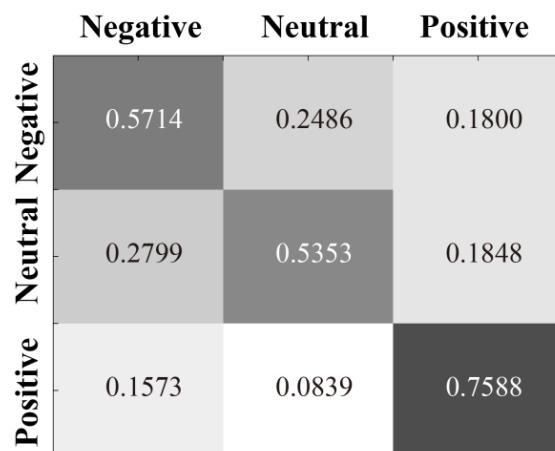
Generalization Performance



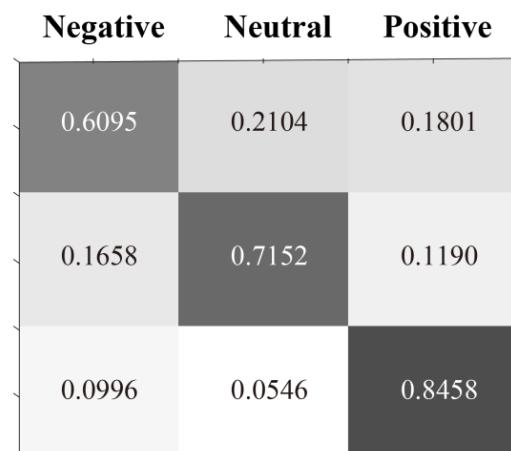
(a) Generic



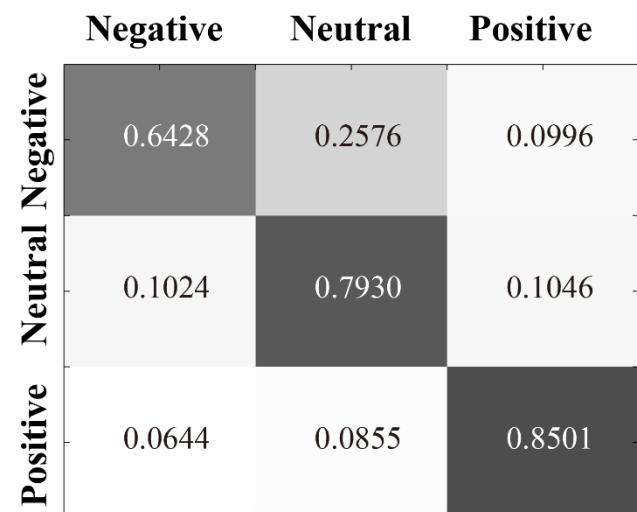
(b) KPCA



(c) TCA



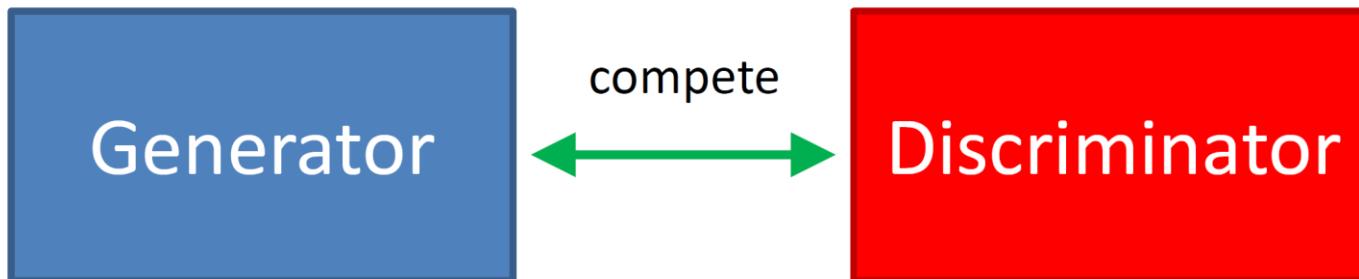
(d) T-SVM



(e) TPT

Generative Adversarial Network (GAN) Model

- Normal GANs consist of two components: generator and discriminator.
- Generator G: creates (fake) samples that the discriminator cannot distinguish.
- Discriminator D: determine whether samples are fake or real.



Goodfellow, Ian; Pouget-Abadie, Jean; Mirza, Mehdi; Xu, Bing; Warde-Farley, David; Ozair, Sherjil; Courville, Aaron; Bengio, Joshua (2014). "Generative Adversarial Networks". arXiv:1406.2661

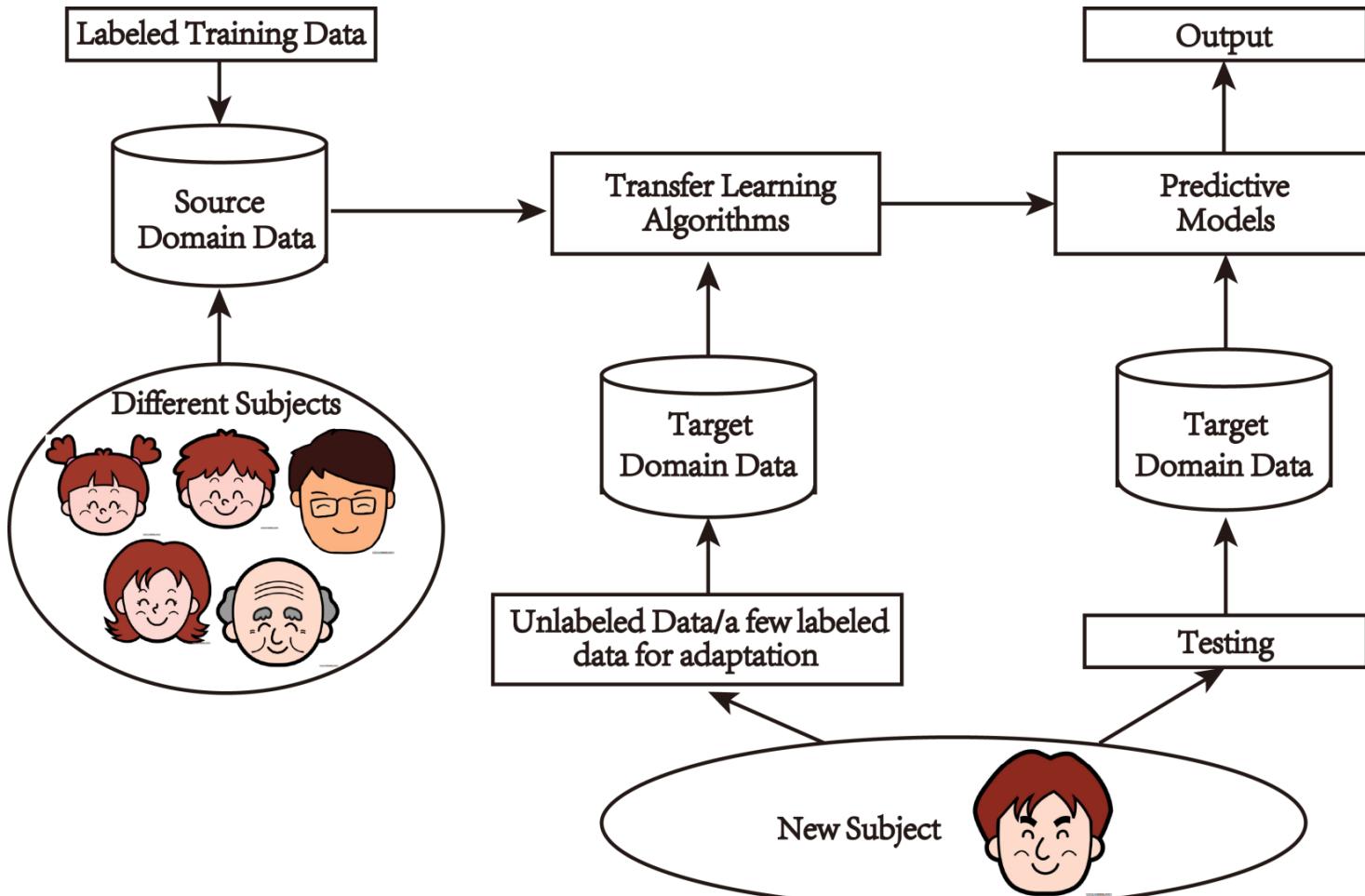
GANs Formulation

- The adversarial training is formulated as a minimax game, where:
 - The discriminator is trying to maximize its reward $V(D, G)$
 - The generator discriminator is trying to maximize its reward $V(D, G)$
- $$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))]$$
- The Nash equilibrium (纳什均衡, global optimality) of this particular game is achieved at:

$$P_{\text{data}}(x) = P_{\text{gen}}(x) \quad \forall x$$

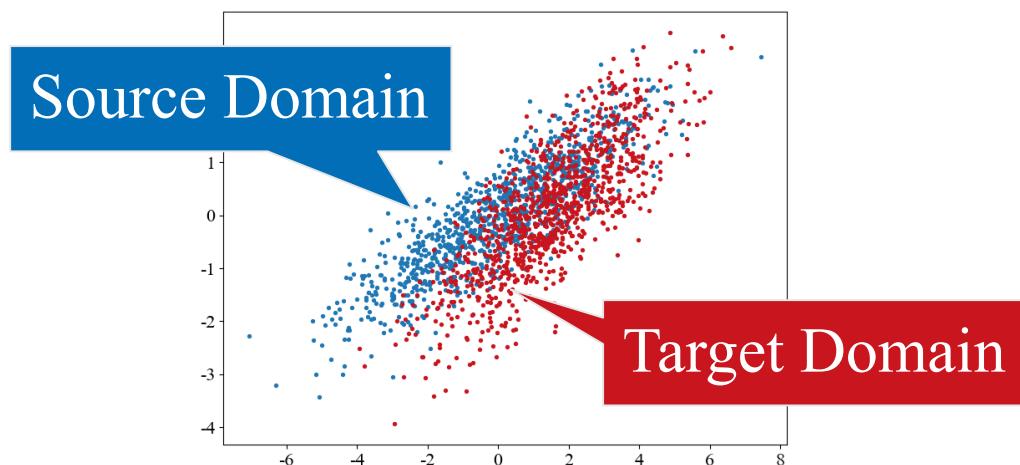
$$D(x) = \frac{1}{2} \quad \forall x$$

Transfer learning framework



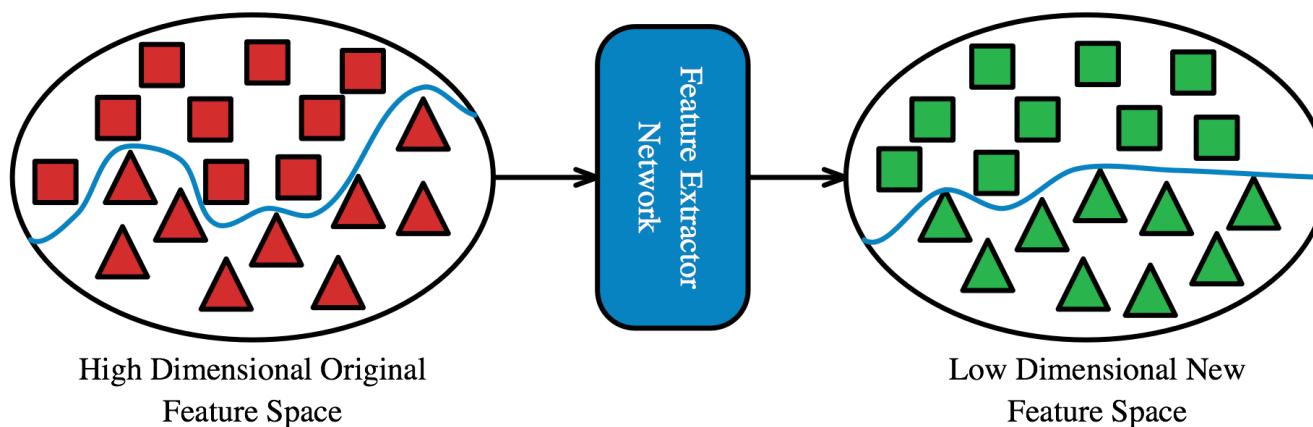
Domain Shift and Domain Adaptation

- Training data are drawn from source domain $\{X_s, \varphi_s\}$, and test data are drawn from target domain $\{X_t, \varphi_t\}$.
- Both domain have the same set of features ($X_s, X_t \in R^m$).
- Domain shift ($P(X_s) \neq P(X_t)$) makes ordinary learning methods degenerate.
- We can use domain adaptation methods to eliminate or reduce the domain shift.



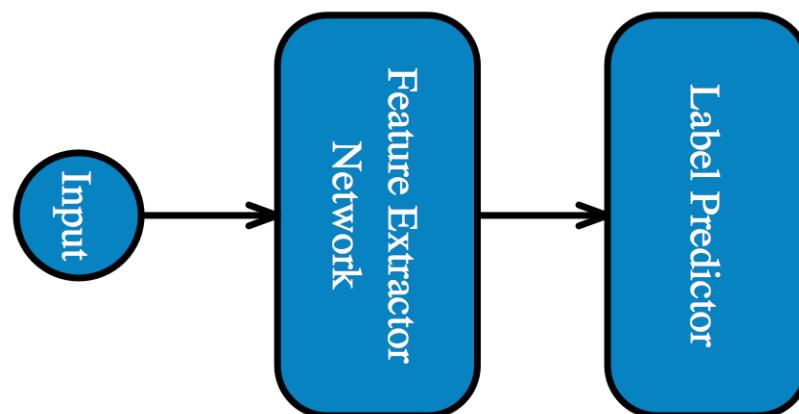
Domain Adaptation with Adversarial Networks

- To reduce the domain shift, neural networks can be used to transform the original feature into a new feature space where two conditions must be satisfied:
 - The new features keep essential information for label prediction
 - Domain shift is reduced (features from both domains share similar distributions in the new feature space).



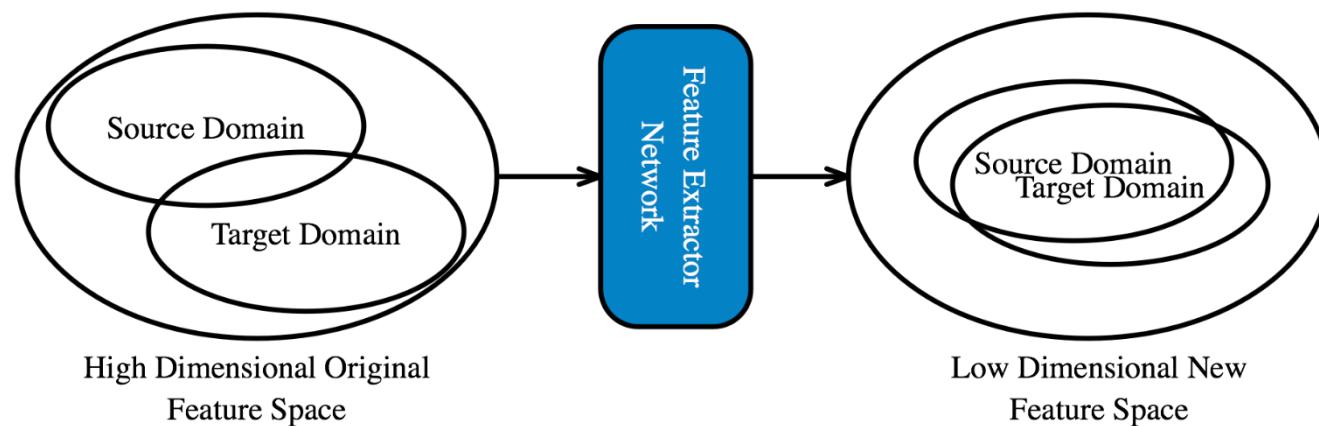
Domain Adaptation with Adversarial Networks

- To reduce the domain shift, neural networks can be used to transform the original feature into a new feature space where two conditions must be satisfied:
 - The new features keep essential information for label prediction
 - Domain shift is reduced (features from both domains share similar distributions in the new feature space).



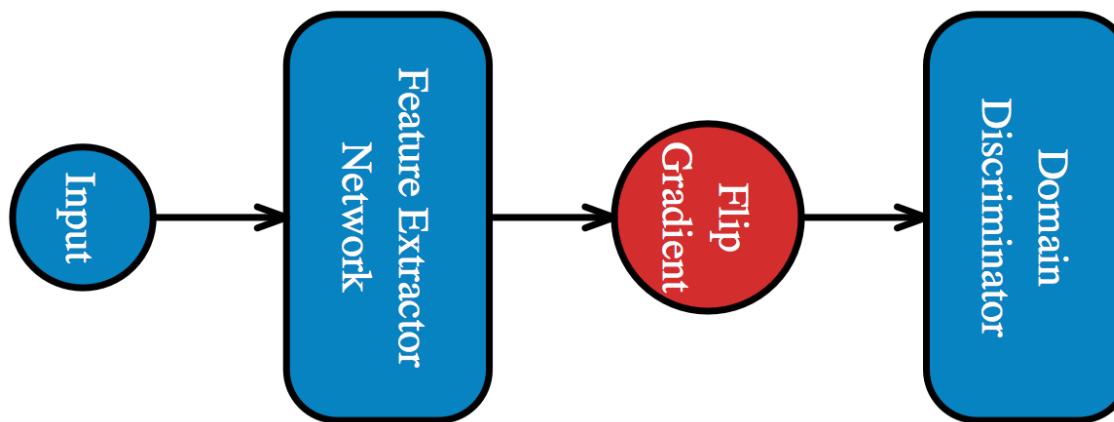
Domain Adaptation with Adversarial Networks

- To reduce the domain shift, neural networks can be used to transform the original feature into a new feature space where two conditions must be satisfied:
 - The new features keep essential information for label prediction
 - Domain shift is reduced (features from both domains share similar distributions in the new feature space).



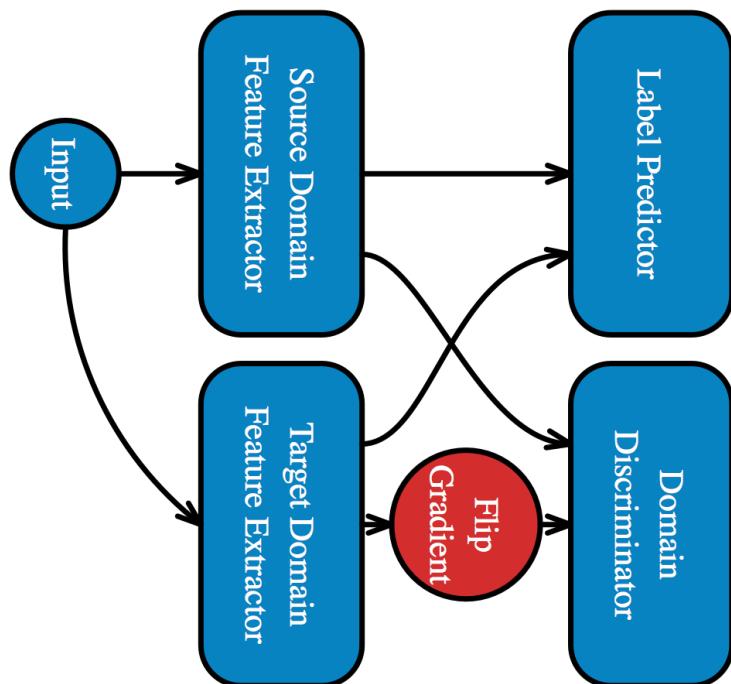
Domain Adaptation with Adversarial Networks

- To reduce the domain shift, neural networks can be used to transform the original feature into a new feature space where two conditions must be satisfied:
 - The new features keep essential information for label prediction
 - Domain shift is reduced (features from both domains share similar distributions in the new feature space).



Adversarial Discriminative Domain Adaptation

- Adversarial discriminative domain adaptation (ADDA) differs from the DANN in that there are different feature extractors for source and target domains and their parameters are trained in different stages.



E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, “Adversarial discriminative domain adaptation,” in IEEE Conference on Computer Vision and Pattern Recognition, July 2017, pp. 2962–2971.

Sleep Quality Estimation with Adversarial Domain Adaptation: From Laboratory to Real Scenario

Jia-Jun Tong, Yun Luo, Bo-Qun Ma, Wei-Long Zheng, Bao-Liang Lu, Xiao-Qi Song, Shi-Wei Ma, Sleep Quality Estimation with Adversarial Domain Adaptation: From Laboratory to Real Scenario, **Proc. IEEE IJCNN 2018**, Rio de Janeiro, 2018

Sleep Quality Estimation for CRH

- China's high-speed railway is currently the world's largest high-speed rail network
- A large number of lines and departures have resulted in a considerable amount of sleep loss for the high-speed train drivers



Li-Li Wang, Wei-Long Zheng, Hai-Wei Ma and Bao-Liang Lu, Measuring Sleep Quality from EEG with Machine Learning Approaches, **Proc. of IEEE IJCNN**, 2016

In the Laboratory

- Before we apply sleep quality estimation to the real scenario of CRH, we have done plenty of experiments in the lab with
 - Controlled last night sleep time of subjects: 4, 6, 8 h
 - High quality EEG acquisition: 62-channel wet electrode cap
 - Different state for same subjects: 3 times, 30 min each experiments



4h
6h
8h

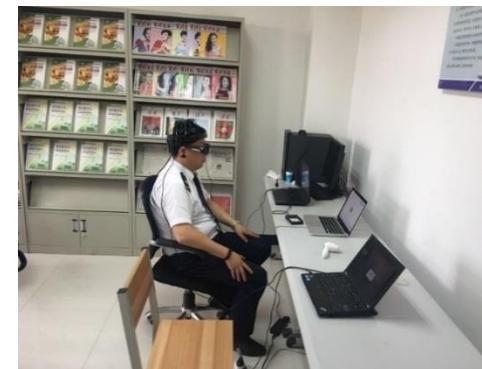


In the Real Scenario (CRH)

- To keep the drivers' daily routine, the experiment settings have to be adjusted to meet the need of real-scenario application.
 - No control on sleep time: range of 2~10 h
 - Easy to set up devices: 18-channel dry electrode cap
 - One experiment for each subjects: 1 time, 6 min each



Sleep:
2~10h



Domain difference

Categories	Specifications	Laboratory	Real-scenario
Subjects	Age	21-26	25-49
	Sex-distribution	4 males, 6females	70 males
	Occupation	Students	High-speed train drivers
	Experiments per subject	3	1
Controlled conditions	Sleep time	4,6,8 hours	Not controlled
	Monitoring sleep	Yes	No
	Head Cleaning	Yes	No
	Experiment Starts	≤1 hour after wake up	1-10 hour after wake up
Experiment settings	Experiment duration	30 min	6 min
	Task	Eyes open	Eyes open and close
	Environment	Quite and isolated	Noisy
Devices	Electrods type	Wet electrodes	Dry electrodes
	Resolution	62 channels	18 channels
	Reference	REF(Between CPZ and PZ)	A1 and A2 (On the ears)
	Common Mode Follower	0	1 (CMF)
	Sample rate	1000 Hz	300 Hz

Domain Adaptation

- After the preprocessing that unifies the format of data, the two domains still possess a lot of discrepancies
- View the Laboratory data as source domain, and the real scenario we can then use domain adaptation methods to transfer knowledges
- Baseline methods
 - TCA, ITL, GFK, JDA, SA, TJM, MIDA
- Adversarial Networks
 - Domain Adversarial Neural Network

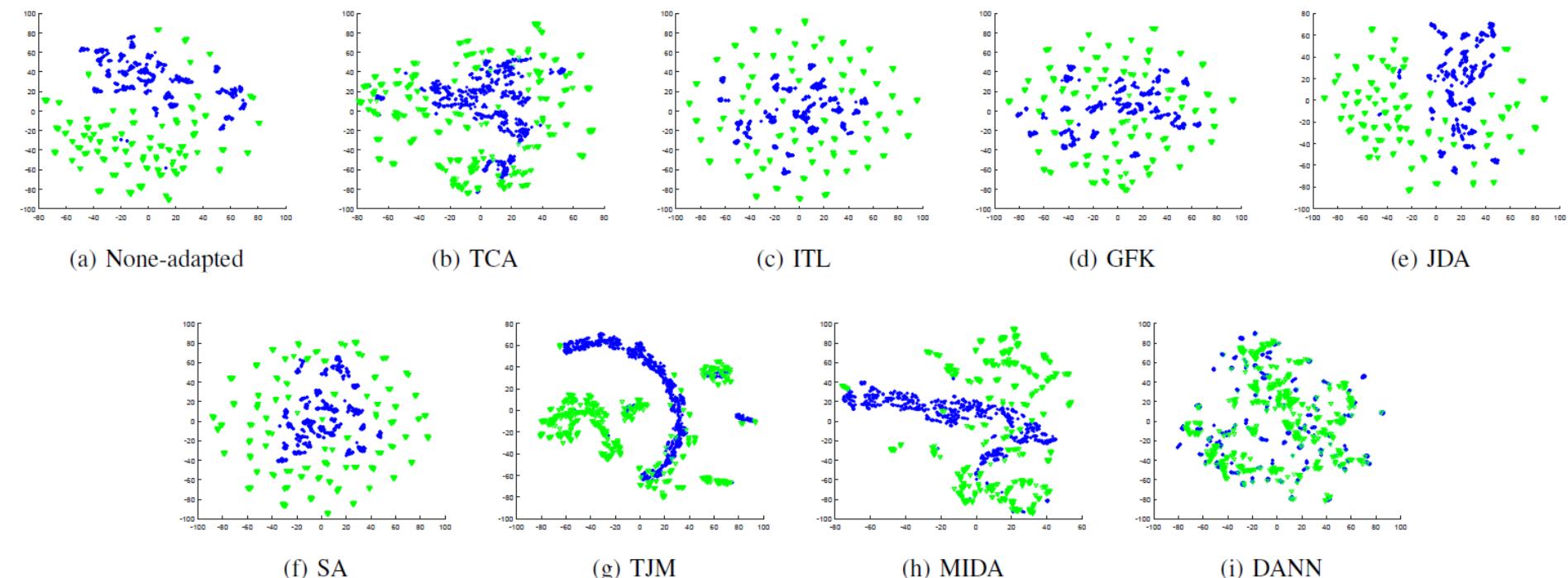
Results

ACCURACY (%) OF CROSS-SCENARIO TASK

Method	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean±Std
SVM	43.86	40.35	42.18	38.98	43.59	41.79±2.10
TCA	54.60	61.19	38.89	67.34	56.87	55.60±10.6
ITL	53.69	63.10	24.68	60.71	49.41	50.32±13.7
GFK	49.64	62.46	40.99	70.95	49.25	54.66±11.9
JDA	47.70	61.43	35.71	66.11	66.94	55.58±13.5
SA	42.74	62.54	37.50	61.23	54.17	51.64±11.1
TJM	41.67	60.28	38.97	59.88	60.60	52.28±10.9
MIDA	56.47	66.35	38.69	68.93	56.87	57.46±11.9
DANN	69.10	64.68	60.79	62.6296	69.25	65.29±3.80

Results (cont.)

The effects of adaptation on cross-scenario task

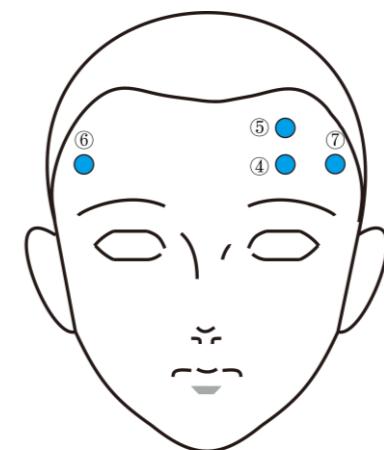


Multimodal Vigilance Estimation with Adversarial Domain Adaptation Networks

He Li, Wei-Long Zheng, and Bao-Liang Lu, Multimodal Vigilance Estimation with Adversarial Domain Adaptation Networks, **Proc. IEEE IJCNN 2018**, Rio de Janeiro, 2018.

The SEED-VIG Dataset

- The SEED-VIG dataset is a publicly available dataset. It contains 23-subject EEG and EOG signals recorded during driving procedures. The vigilance levels are also labeled in terms of PERCLOS indices.



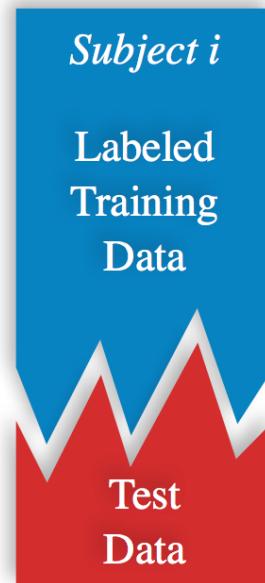
Wei-Long Zheng and Bao-Liang Lu, A multimodal approach to estimating vigilance using EEG and forehead EOG. *Journal of Neural Engineering*, 14(2): 026017, 2017.

Difficulty in Data Acquisition

- Conventionally, to train a vigilance estimation system for subject A, we have to obtain his EEG&EOG signals during the driving procedure, as well as the corresponding labels (here, the vigilance level indices).
- However, it is usually expensive or effort demanding to obtain such labeled data set:
 - Some kinds of vigilance level indices require the subject perform a parallel task while driving, which is disturbing.
 - Some kinds of vigilance level indices are obtained by watching the video of the subject and labeling the vigilance level by hand.
 - Some kinds of vigilance level indices demand for extra device, e.g., eye tracking glasses, which is both expensive for the facilitators and inconvenient for the subjects.

Cross-Subject Vigilance Estimation

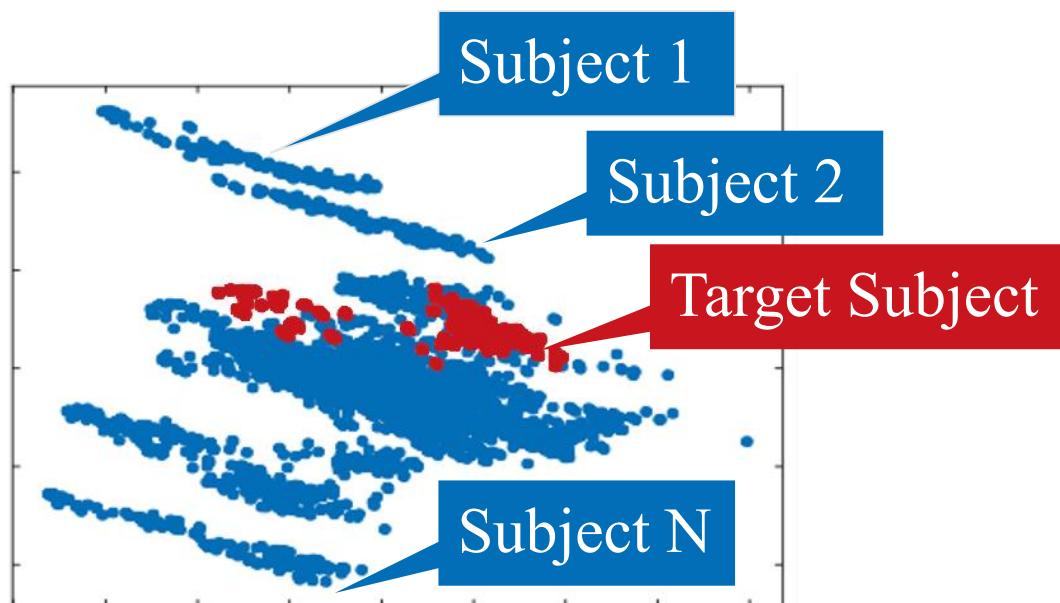
- To solve the problem, in this study, we utilize the existing labeled data from other subjects (source domain) for training of the models for the target subject (target domain). In this way, label information is not necessary for the target subject and the existing labeled data can be reused.



Subject Oriented Model

Inter-Subject Discrepancy (Domain Shift)

- However, the data distributions vary a lot for different subjects due to the inter-subject discrepancy. As the result, the model trained on the blue source domain data can perform badly on the red target domain data.



Eliminating the Discrepancy

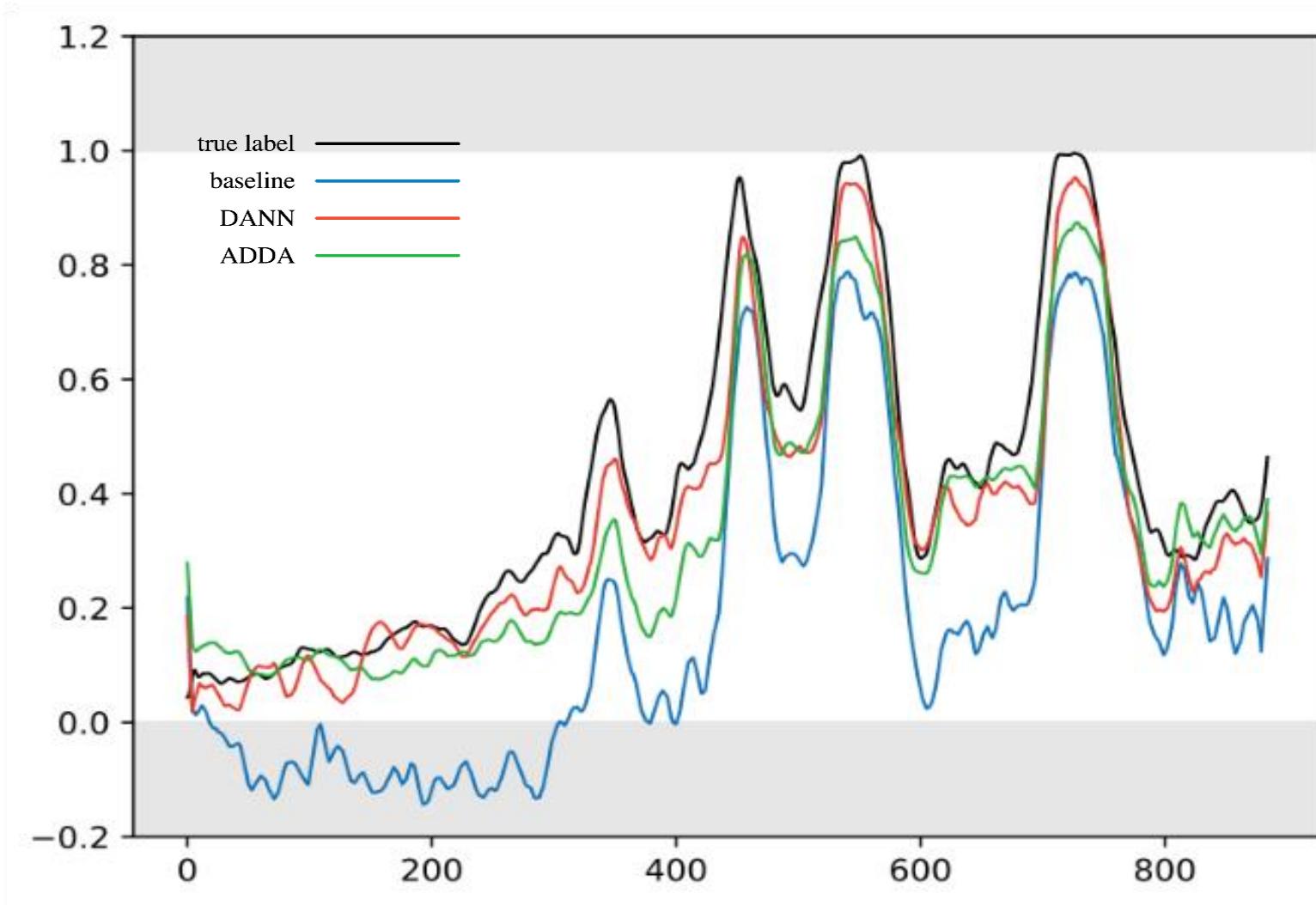
- In order to eliminate the inter-subject discrepancy, we adopt domain adaptation methods: DANN and ADDA.
- In our study, the source domain is the existing labeled data, and the target domain is the unlabeled target subject data.
- Because our task is a regression problem, we use single neuron with identity activation function as the output layer for the neural networks.
- ADDA does not support transfer learning with multiple source domains, so we combine all the source subject data to form a single source domain.

Results

- Because our problem is a regression problem, we adopt Pearson correlation coefficients (PCCs) and root-mean-square errors (RMSEs) for the evaluation of the results. The “Baseline” indicates no domain adaptation method is applied.

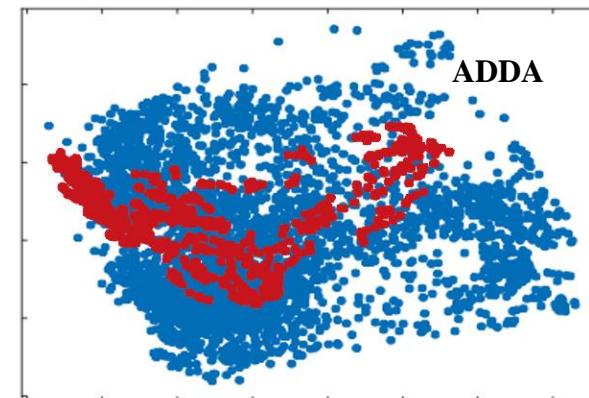
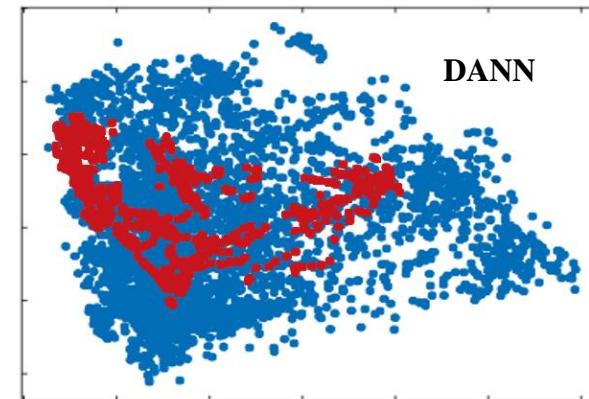
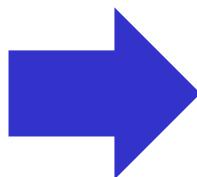
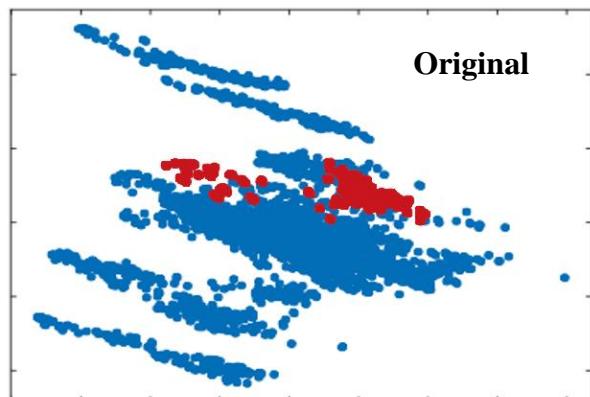
		Baseline	TCA	MIDA	SA	DANN	ADDA
PCC	AVG	0.7606	0.7722	0.8024	0.7707	0.8402	0.8442
	STD	0.2314	0.2061	0.1629	0.0745	0.1535	0.1336
RMSE	AVG	0.1689	0.1607	0.1701	0.1667	0.1427	0.1405
	STD	0.0673	0.0513	0.0686	0.0746	0.0588	0.0514

Results (cont.)



Data Distributions

- To show how the methods influence the data distribution, we reduce the dimension of the data with PCA and plot the distributions as follows.



Useful Sites on Transfer Learning

- Resources:
 - Open source program: <http://www.cse.ust.hk/TL/>
 - Qiang Yang: <http://www.cs.ust.hk/~qyang/>
 - Sinno Jialin Pan: <http://www.ntu.edu.sg/home/sinnopan/>
 - Wenyuan Dai: <http://www.4paradigm.com/homepage.html>

- Survey:
 - A survey on Transfer Learning.
 - Transfer learning for activity recognition: A survey.
 - Transitive Transfer Learning.
 - Fuzzy Transfer Learning: Methodology and application.

Performance evaluation Index

混淆矩阵 (Confusion Matrix)

	预测正类	预测负类
正类	TP	FN
负类	FP	TN

TP (True Positive)：样本属于正类，预测结果为正类

FP (False Positive)：样本属于负类，预测结果为正类

FN (False Negative)：样本属于正类，预测结果为负类

TN (True Negative)：样本属于负类，预测结果为负类

常用的评价指标：精度

$$\text{Accuracy} = \frac{a + d}{a + b + c + d} = \frac{TP + TN}{TP + TN + FP + FN}$$

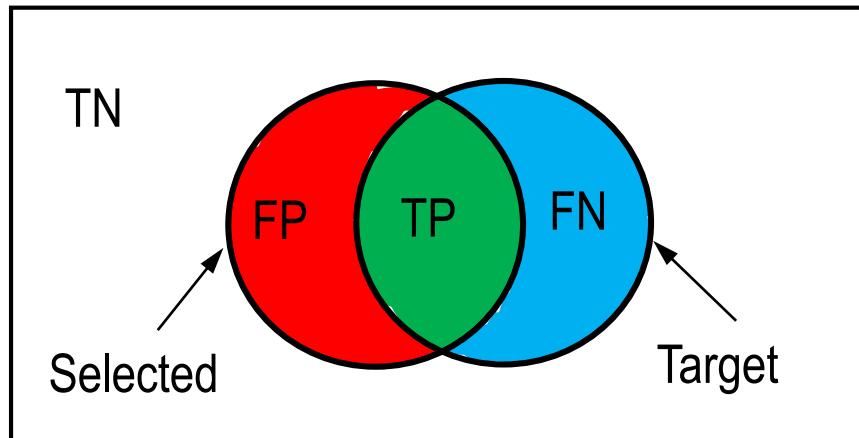
使用精度度量分类器的局限性

- 考虑一个2类问题：
 - 第一类有9990个测试样本
 - 第二类只有10个测试样本
- 如果分类器把全部测试样本都分为第一类， 其精度为 $9990/10000=99\%$ ！
- 显然， 这里凸显出精度的局限性。因为， 它未能全面地衡量分类器对第二类的分类性能。

$$\text{Accuracy} = \frac{a + d}{a + b + c + d} = \frac{TP + TN}{TP + TN + FP + FN}$$

精确率、召回率、F1度量

- 真正 (True positive, TP); 假负 (False negative, FN)
- 假正 (False positive, FP) ;真负 (True negative, TN)
- 真正率 (True positive rate, TPR): $TPR = TP / (TP + FN)$
- 假正率 (False positive rate, FPR): $FPR = FP / (FP + TN)$
- 精确率 (Precision): $p = TP / (TP + FP)$
- 召回率 (Recall): $r = TP / (TP + FN)$
- F1度量: $F1 = 2r * p / (r + p)$



p: 查准率
r: 查全率

Macro-p

Macro-r

Macro-F1

先计算p、r再求平均

Micro-p

Micro-r

Micro-F1

先求TP、FP、TN、FN
平均, 再求p、r、F1

Macro & Micro-p、r、F1

p: 查准率

r: 查全率

Macro-p

Macro-r

Macro-F1

先计算p、r，再求平均

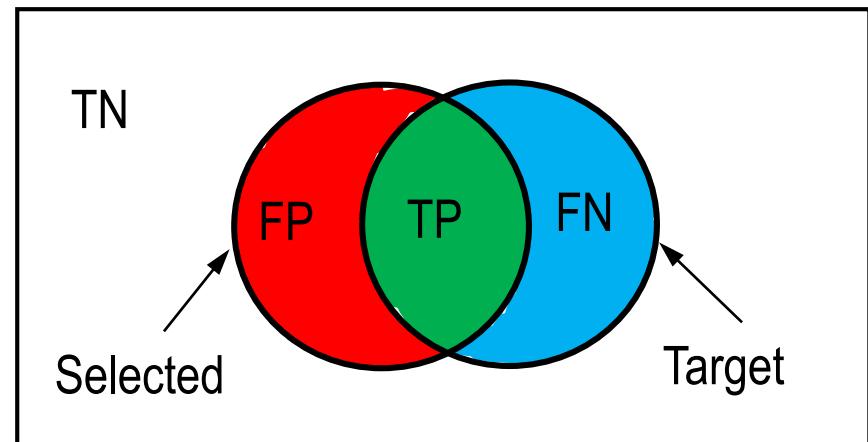
Micro-p

Micro-r

Micro-F1

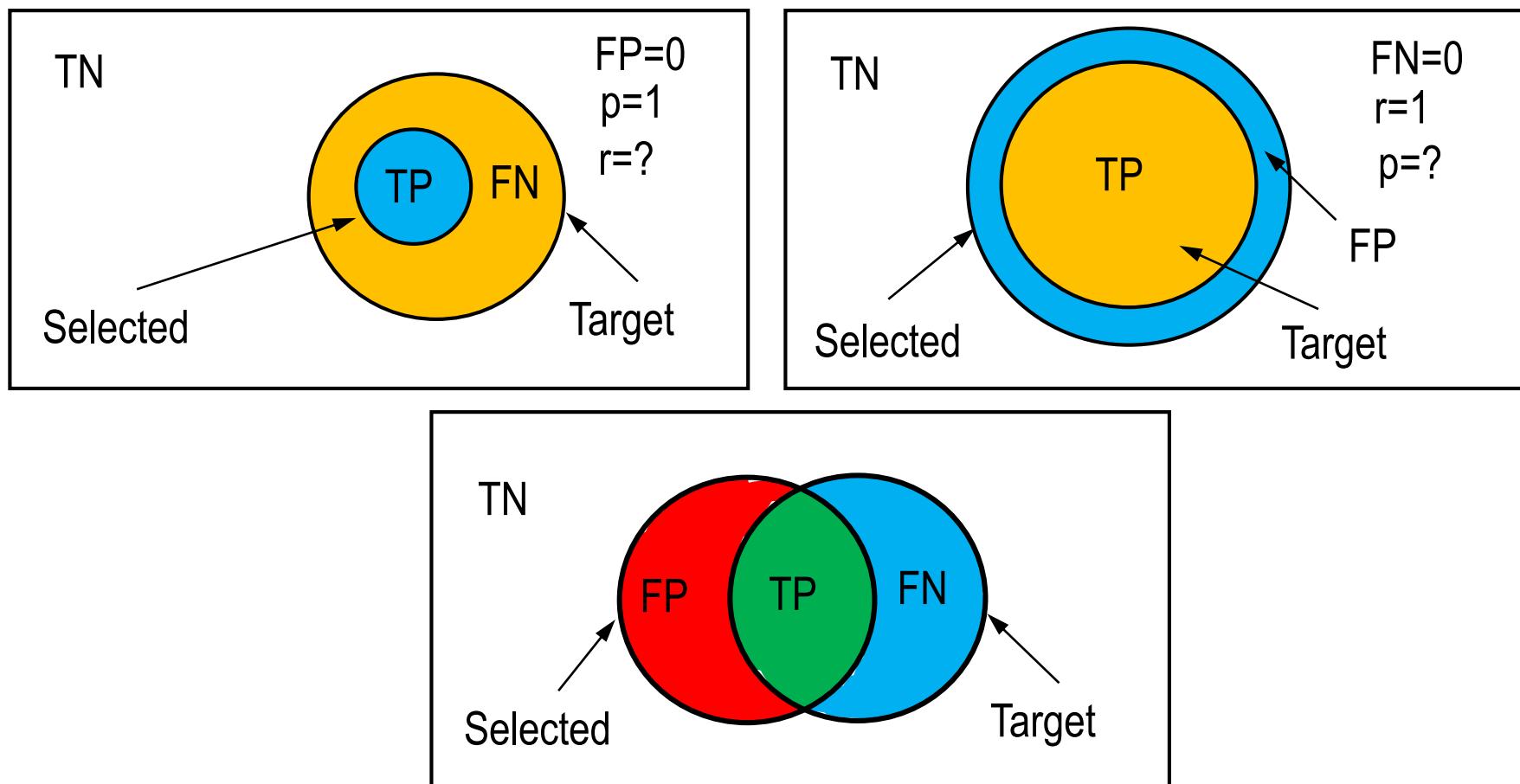
先求TP、FP、TN、FN

平均，再求p、r、F1



精确率、召回率、F1度量

- 精确率 (Precision): $p=TP/(TP+FP)$
- 召回率 (Recall): $r=TP/(TP+FN)$
- F1度量: $F1=2*r*p/(r+p)$

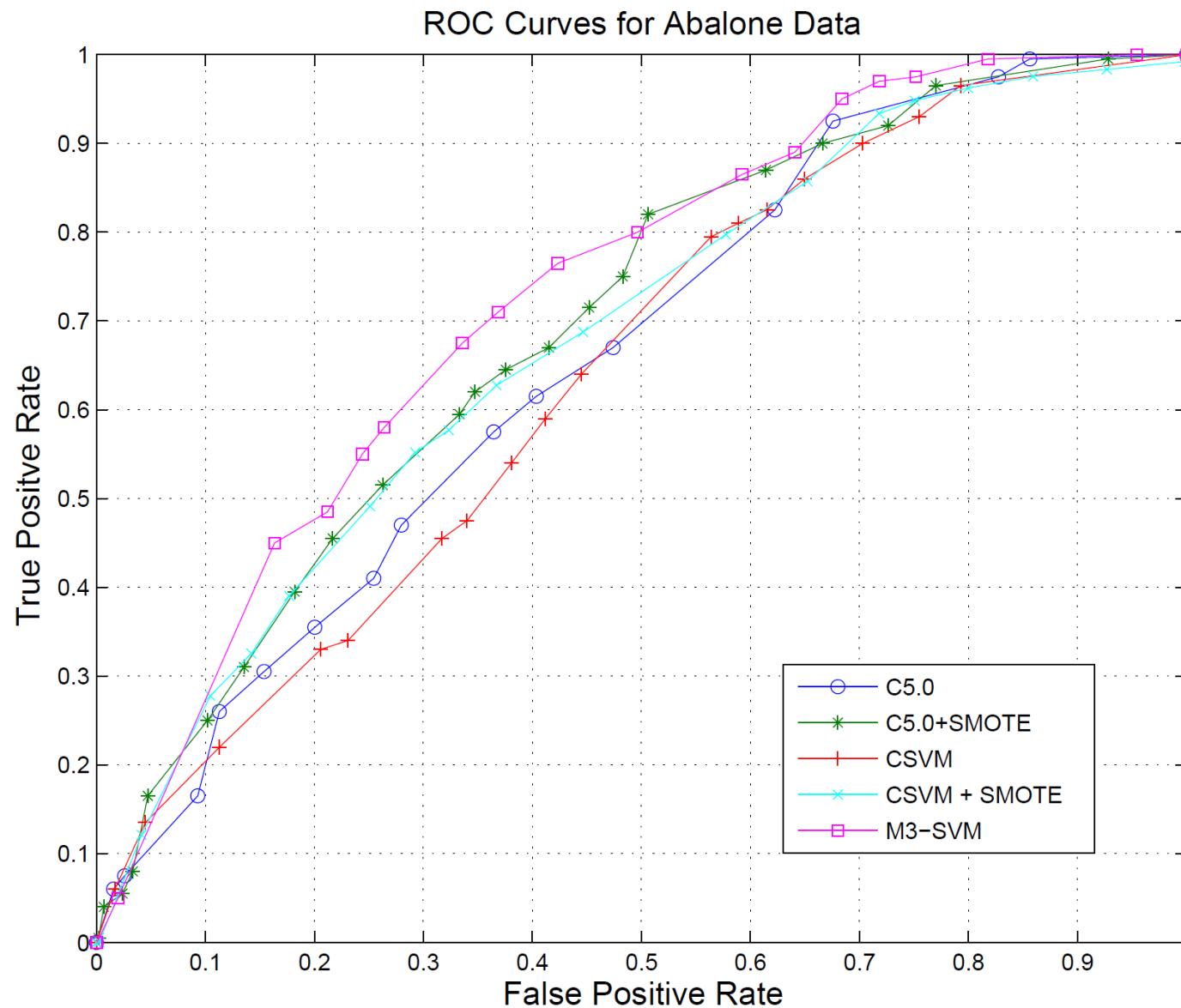


ROC (Receiver Operating Characteristic) 曲线

ROC (Receiver Operating Characteristic) 曲线

- Developed in 1950s for signal detection theory to analyze noisy signals
 - Characterize the trade-off between positive hits and false alarms
- ROC curve plots TPR (on the y-axis) against FPR (on the x-axis)
- Performance of each classifier represented as a point on the ROC curve
 - Changing the threshold of algorithm, sample distribution or cost matrix changes the location of the point
- AUC: Area under ROC Curve

使用ROC曲线比较 分类算法

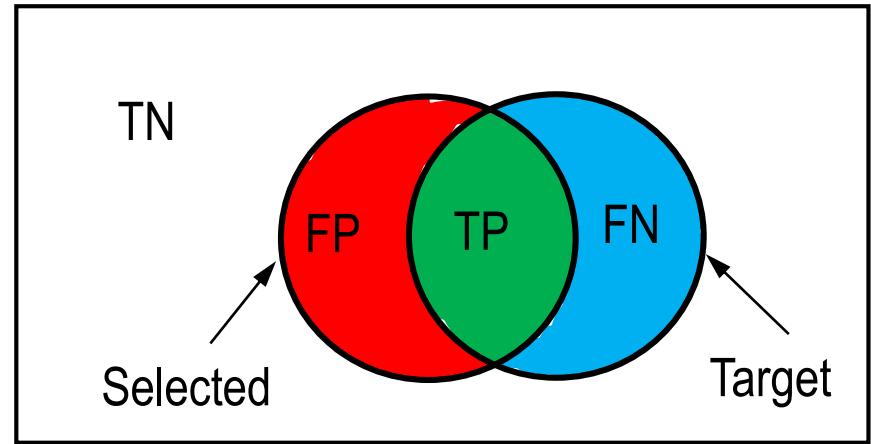


评估指标

□ 真正类率（召回率）

TPR (True Positive Rate):

$$TPR = \frac{TP}{TP + FN}$$



□ 假正类率

FPR (False Positive Rate):

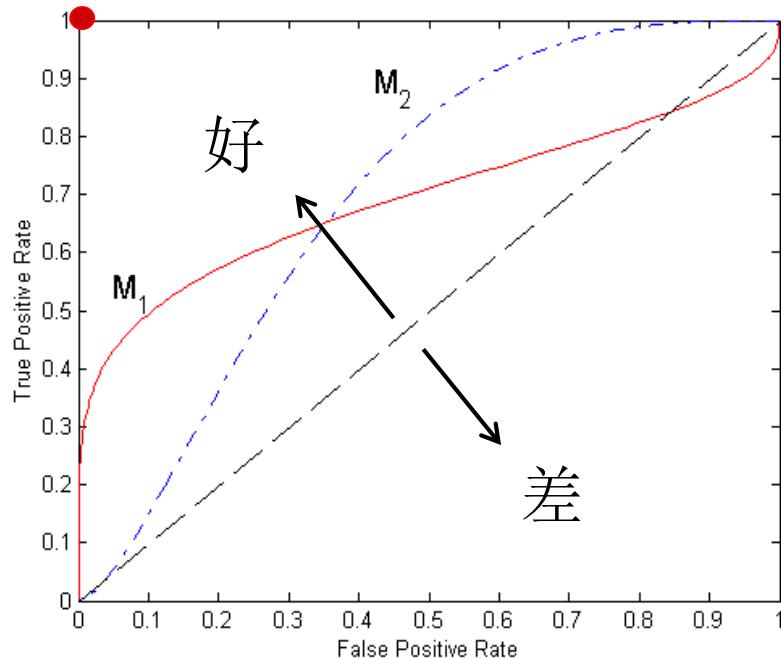
$$FPR = \frac{FP}{FP + TN}$$

□ TPR越大，分类效果越好。而FPR越大，分类效果越差。

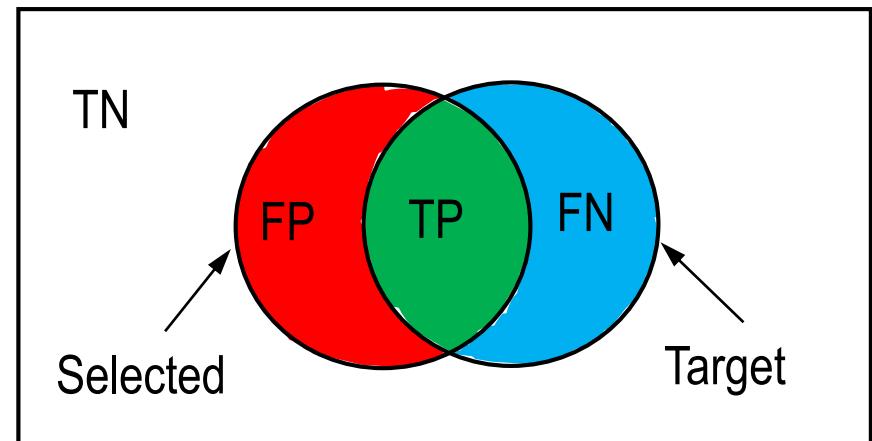
ROC 曲线的几个关键点

- $(TPR=0, FPR=0)$: 把每个输入都预测为负类;
- $(TPR=1, FPR=1)$: 把每个输入都预测为正类
- $(TPR=1, FPR=0)$: 理想模型
- 主对角线: 随机猜测

(随机猜测是指以固定概率 p 把输入分为正类)



$$TPR = TP / (TP + FN)$$
$$FPR = FP / (FP + TN)$$



使用AUC指标比较分类算法

表 6.2 5种方法在三个数据上的结果

数据	方法	TP%	TN%	B-ACC%	AUC
Rooftop	C5.0	78.5	80.2	79.9	87.43
	CSVM	80.3	81.8	81.1	87.98
	C5.0 + SMOTE	79.9	80.1	80.0	88.22
	CSVM + SMOTE	81.3	80.4	80.9	87.87
	M3-SVM	81.6	81.4	81.5	89.28
Park	C5.0	82.6	85.8	84.2	90.39
	CSVM	84.9	85.5	85.2	93.93
	C5.0 + SMOTE	84.3	83.8	84.2	90.96
	CSVM + SMOTE	85.4	85.1	85.3	94.10
	M3-SVM	87.2	87.7	87.5	94.54
Abalone	C5.0	61.5	59.6	60.6	66.84
	CSVM	59.0	58.8	58.9	64.25
	C5.0 + SMOTE	64.5	62.4	63.5	69.53
	CSVM + SMOTE	62.7	63.3	63.0	68.00
	M3-SVM	67.5	66.4	67.0	72.67

不平衡分类问题研究综述

叶志飞¹, 文益民², 吕宝粮^{1,3}

(1. 上海交通大学 计算机科学与工程系, 上海 200240; 2. 湖南工业职业技术学院 信息工程系, 湖南 长沙 410208;
3. 上海交通大学 智能计算与智能系统教育部微软重点实验室, 上海 200240)

摘要: 实际的分类问题往往都是不平衡分类问题, 采用传统的分类方法, 难以得到满意的分类效果。为此, 十多年来, 人们相继提出了各种解决方案。对国内外不平衡分类问题的研究做了比较详细地综述, 讨论了数据不平衡性引发的问题, 介绍了目前几种主要的解决方案。通过仿真实验, 比较了具有代表性的重采样法、代价敏感学习、训练集划分以及分类器集成在 3 个实际的不平衡数据集上的分类性能, 发现训练集划分和分类器集成方法能较好地处理不平衡数据集, 给出了针对不平衡分类问题的分类器评测指标和将来的工作。

关键词: 机器学习; 不平衡模式分类; 重采样; 代价敏感学习; 训练集划分; 分类器集成; 分类器性能评测

二类分类器预测过程

- 一般二类分类器在预测时会计算一个评估函数

$$f(x; w)$$

其中， x 为待预测样本的特征向量， w 为已训练的参数。对于线性分类器有 $f(x; w) = x \cdot w_0$ 。

- 预测结果如下输出：

- 若 $f(x; w) > 0$ ，分类器输出正类
- 若 $f(x; w) < 0$ ，分类器输出负类

二类分类器预测过程

- 通过引入阈值 t ，改变分类器预测时对正负类的倾向：
 - 若 $f(x; w) > t$ ，分类器输出正类。
 - 若 $f(x; w) < t$ ，分类器输出负类。
- 阈值 t 增大，分类器预测结果偏向负类。
- 阈值 t 减小，分类器预测结果偏向正类。

ROC曲线的绘制

- ROC曲线以FPR为横轴， TPR为纵轴。
- 设置不同的阈值 t ，分类器预测结果有不同的 FPR值和TPR值。

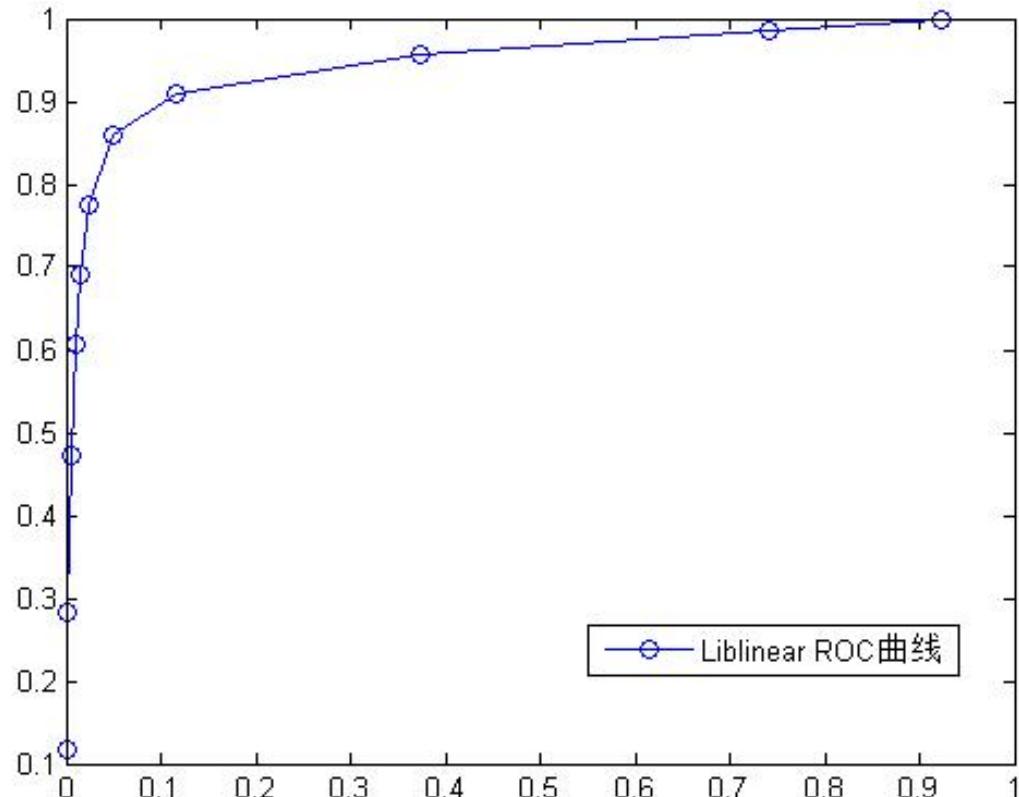
例： Liblinear的ROC曲线

- 分别以-8,-4,-2,-1,-0.5,0,0.5,1,2,4,8为阈值， 使用 Liblinear进行预测，结果如下表：

t	-8	-4	-2	-1	-0.5	0	0.5	1	2	4	8
FPR	0.001	0.002	0.005	0.010	0.014	0.025	0.049	0.115	0.373	0.741	0.924
TPR	0.117	0.282	0.472	0.606	0.691	0.774	0.859	0.910	0.956	0.986	0.998

例：Liblinear的ROC曲线

- 使用画图工具将表格中的数据绘成图片。如下图：



ROC曲线与分类效果

- ROC曲线下方面积越大，分类效果越好。下图是Liblinear的ROC曲线与M3-Liblinear的ROC曲线。

