On Decomposition of Two-Class Pattern Classification Problems

Bao-Liang Lu (吕宝粮)

Center for Brain-Like Computing and Machine Intelligence
Department of Computer Science and Engineering
MOE-Microsoft Key Lab for Intelligent Computing and Intelligent Systems
Shanghai Jiao Tong University, China

bllu@sjtu.edu.cn
http://bcmi.sjtu.edu.cn/~blu

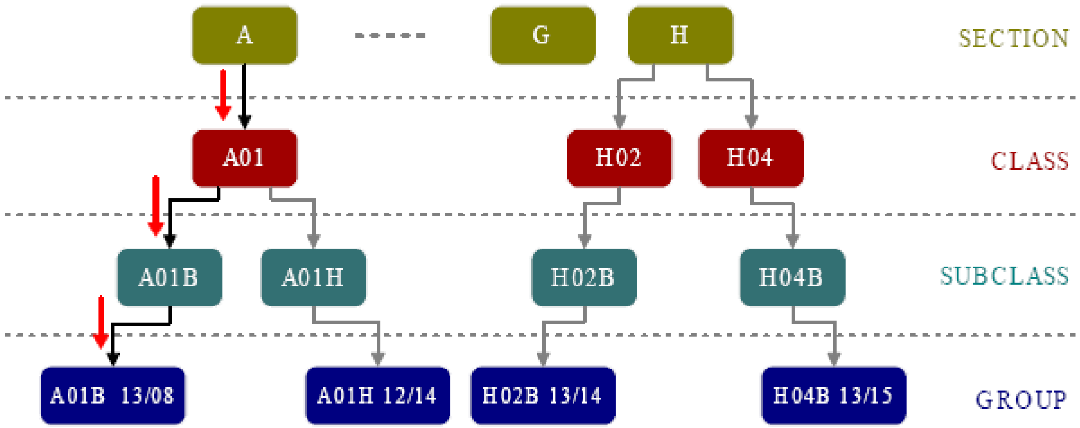
Motivation

- Training data are increasing rapidly
- Cluster systems are available for machine learning researchers
- Real-world applications need parallel machine learning
- Existing approaches:
  - Implement traditional learning algorithms in parallel
  - Parallel and distributed learning model

Problem

- Supervised learning
- Characteristics of classification problems
  - Large-scale data set
  - Imbalance
  - Multi-label
  - Hierarchical
  - Time-varying feature
- An example
  - Japanese patent classification
  - Patent applications from 1993 to 2002
  - Total number of patent data is 3496137

Japanese patent classification



		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

Existing two task decomposition strategies

- One-versus-all (OVA)
  - K-class problem → K two-class sub-problems

$$X_i = \left\{ X_l^{(i)} \right\}_{l=1}^{L_i} \quad \text{for } i = 1, 2, \ldots, K$$

$$T_i = \underline{\left\{ (X_l^{(i)}, 1-\varepsilon) \right\}_{l=1}^{L_i}} \cup \underline{\left(\bigcup_{j=1, j\neq i}^{K} \left\{ (X_l^{(j)}, \varepsilon) \right\}_{l=1}^{L_j} \right)} \quad \text{for } i = 1, \ldots K$$

- One-versus-one (OVO)
  - K-class problem → K(K-1)/2 two-class sub-problems

$$X_i = \left\{ X_l^{(i)} \right\}_{l=1}^{L_i} \quad \text{for } i = 1, 2, \ldots, K$$

$$T_{ij} = \underline{\left\{ (X_l^{(i)}, 1-\varepsilon) \right\}_{l=1}^{L_i}} \cup \underline{\left\{ (X_l^{(j)}, \varepsilon) \right\}_{l=1}^{L_j}} \quad \text{for } i = 1, \ldots K \text{ and } \; j = i+1$$

Virtues and limitations

- Decomposition is unique
- OVA:
  - The sizes of all of the two-class problems are the same as the original one
  - Some of the two-class problems become imbalanced problem, e.g. 50540|3445608;1:68
- OVO:
  - Some of the two-class problems may still be too large to learn , e.g. 857587+786473=1644060
- Rifkin & Klautau, “In defense of one-vs-all classification”, JMLR, 2004

Reasons to decompose two-class problems

- Incorporate prior knowledge into learning
- Balance training data
- Deal with multi-label task
- Implement parallel and distributed learning

Speed-up training and improve generalization performance!

Gender classification problem

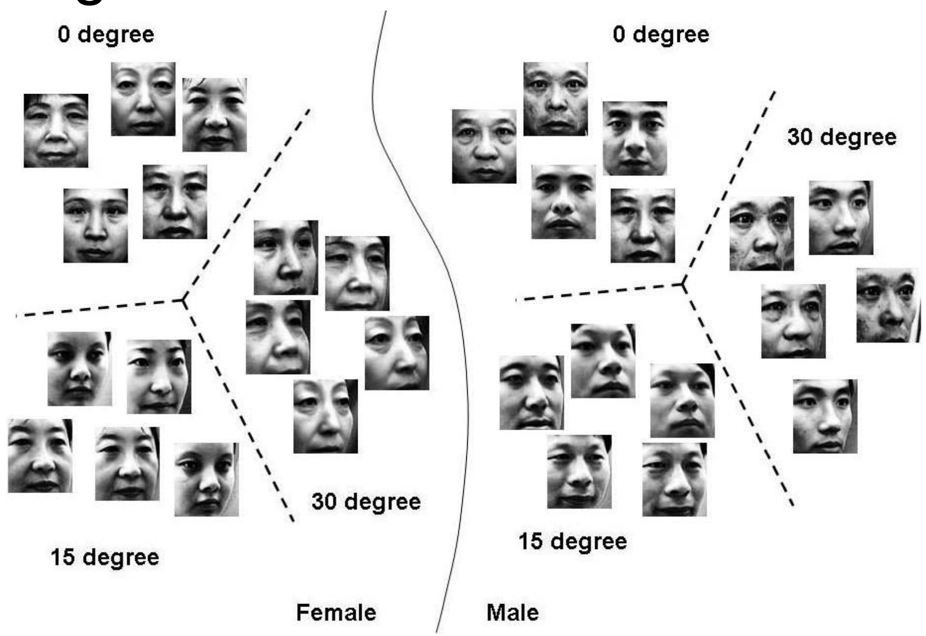
- It is two-class problem!

$$X_i = \left\{ X_l^{(i)} \right\}_{l=1}^{L_i} \quad \text{for } i = 1, 2$$

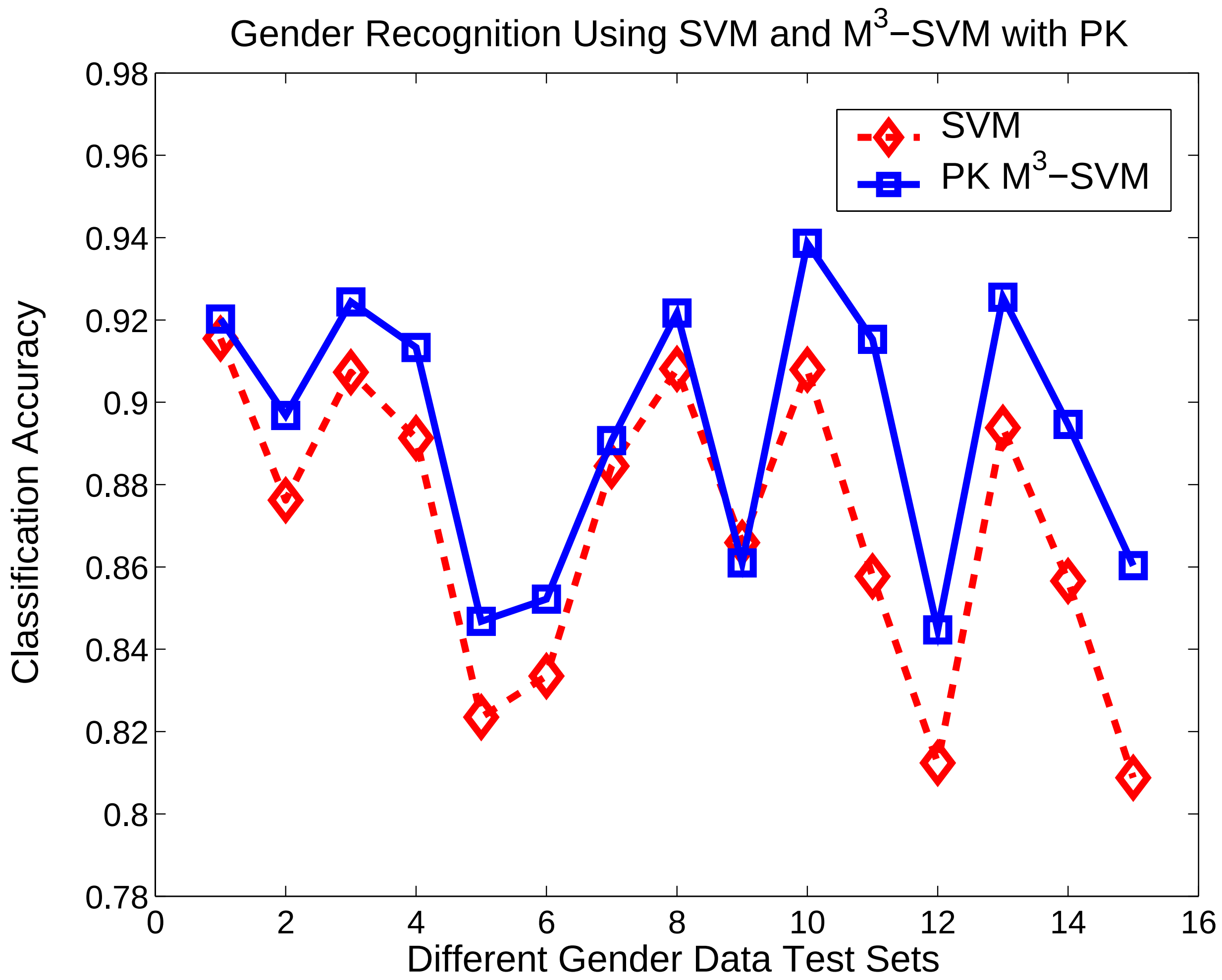
$$T = \left\{ (X_l^{(1)}, 1-\varepsilon) \right\}_{l=1}^{L_1} \cup \left\{ (X_l^{(2)}, \varepsilon) \right\}_{l=1}^{L_2}$$

- Some explicit Prior knowledge
  - Different View
  - Different Ages
  - Different races

  ⋮



Effect of incorporating prior knowledge



An example of Japanese patent

	PATENT-JA-UPA-1998-000001
<Bibliography>	
[publication date]	【公開日】平成１０年（１９９８）１月６日
[title of invention]	【発明の名称】土壌改良方法とその作業機
---	---
<Abstract>	
[purpose]	【課題】 心土破砕、特に雪上心土破砕作業の際に積雪
[solution]	【解決手段】心土破砕を行うために用いるサブソイラの
---	---
<Claims>	
[claim1]	【請求項１】 サブソイラ作業機を用いて心土破砕作業
[claim2]	【請求項２】 サブソイラ作業機において、そのナイフ
---	---
<Description>	
[technique field]	【発明の属する技術分野】本発明は、土壌改良方法とそ
[prior art]	【従来の技術】圃場の表面がまだ積雪に覆われている状
[problem to be solved]	【発明が解決しようとする課題】心土破砕は通常春先に
---	---
<Explanation of Drawing>	
[figure1]	【図１】 本発明を施す圃場断面図である。
---	---

International patent classification (IPC)

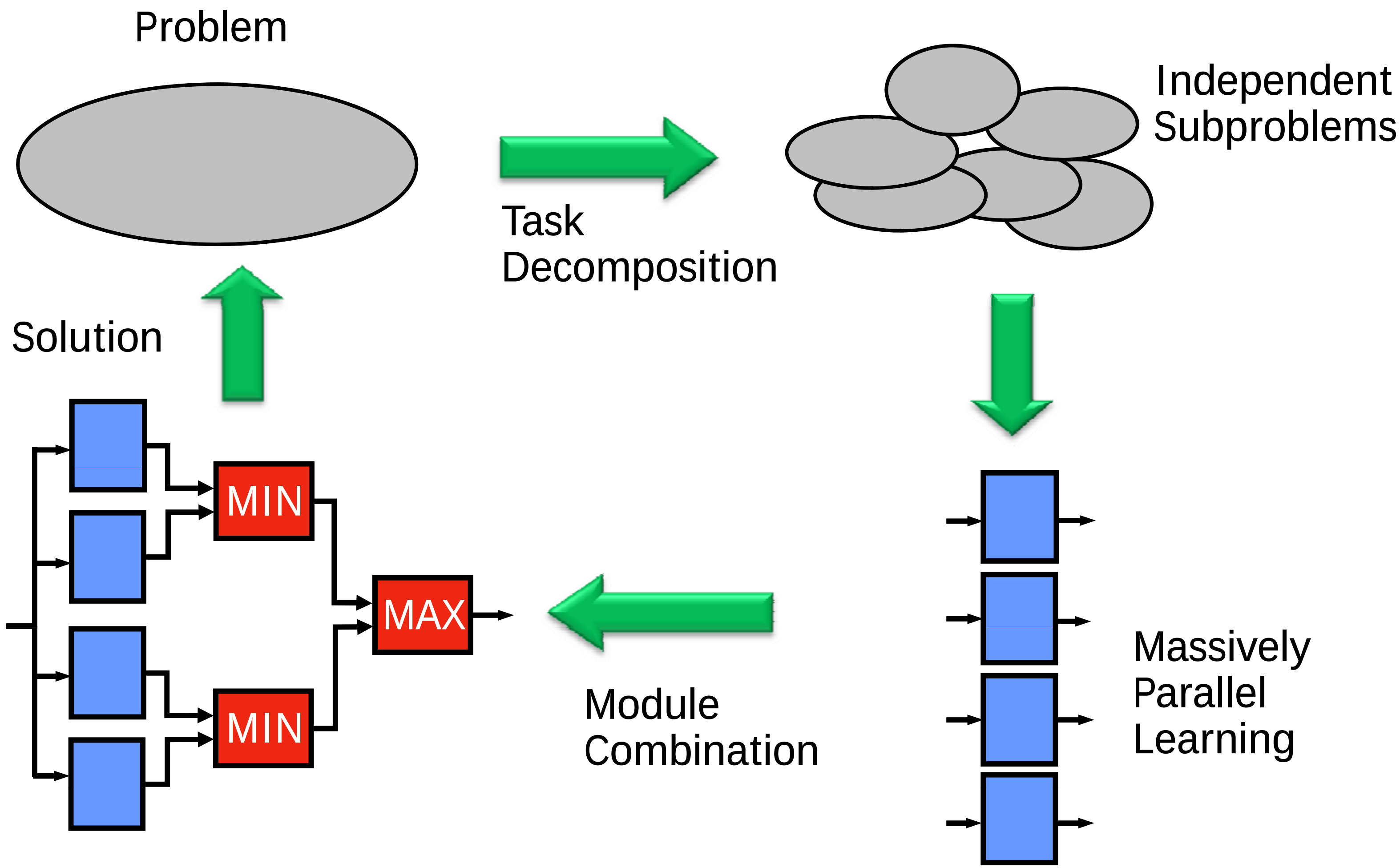
A	01	B	1	/02
セクション				
クラス				
サブクラス				
メイングループ				
サブグループ				

A	セクション	生活必需品
A01	クラス	農業、林業、畜産、狩猟、捕獲、漁業
A01B	サブクラス	農業または林業における土作業、農業機械または器具の部品、細部または附属具一般
A01B 1/00	メイングループ	手作業具
A01B 1/02	サブグループ	鋤、ショベル

Min-Max Modular (M3) Neural Network Model

(Lu & Ito, 1997, 1999)

Min-Max Modular Network Model



Min-Max Modular Support Vector Machine

(Lu *et al.*, 2004)

Min-Max Modular Support Vector Machine

- Part-vs-part: Any two-class problem can be further decomposed into a number of two-class sub-problems as small as needed.
- Two module combination rules.
- It is independent of learning tasks

Part-versus-part task decomposition

- Training data for a K-class problem

$$T = \{ (X_l, Y_l) \}_{l=1}^{L}$$

- Decompose a K-class problem into K(K-1)/2 two-class problems

$$X_i = \left\{ X_l^{(i)} \right\}_{l=1}^{L_i} \quad \text{for } i = 1, 2, \ldots, \mathrm{K}$$

$$T_{ij} = \left\{ (X_l^{(i)}, 1-\varepsilon) \right\}_{l=1}^{L_i} \bigcup \left\{ (X_l^{(j)}, \varepsilon) \right\}_{l=1}^{L_j} \text{ for } i = 1, \ldots K \text{ and } \; j = i+1$$

- Decompose a two-class problem into a number of relatively balanced two-class problems as smaller as needed

$$\text{Partition of } X_i \text{ into } N_i \text{ subsets } \; X_{ij} = \left\{ X_l^{(ij)} \right\}_{l=1}^{L_i^{(j)}} \quad \text{for } \; j = 1, \ldots, N_i$$

$$T_{ij}^{(u,v)} = \left\{ (X_l^{(iu)}, 1-\varepsilon) \right\}_{l=1}^{L_i^{(u)}} \bigcup \left\{ (X_l^{(jv)}, \varepsilon) \right\}_{l=1}^{L_j^{(v)}}$$

$$\text{for } u = 1, \ldots, N_i, v = 1, \ldots, N_j \text{, and } \; j \neq i$$

Number of Two-class Problems

- Number of smaller two-class problems

$$\sum_{i=1}^{K-1} \sum_{j=i+1}^{K} N_i \times N_j$$

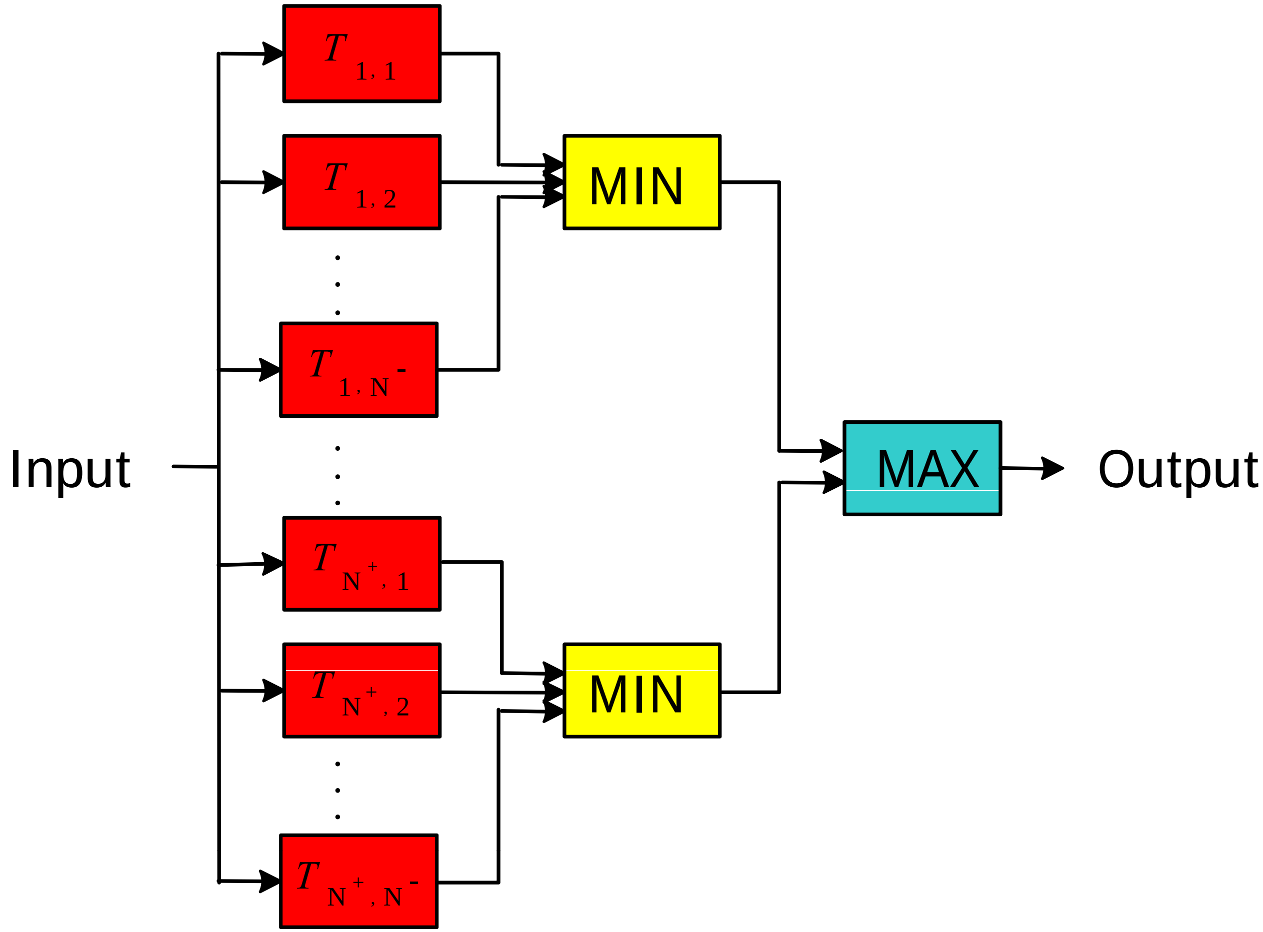
N_i is the number of subsets for class C_i

- Number of training data for each of the two-class sub-problems is about

$$\lceil L_i / N_i \rceil + \lceil L_j / N_j \rceil$$

L_i is the number of training data for class C_i

Two-spirals problem



Time Complexity Analysis

- Empirical observation (Joachims, 2002):

$O((l^{+} + l^{-})^{c})$ c is domain-specific (1.2~1.7)

- Time complexity of M3-SVM in a parallel way

$$O\left(\left(\left\lfloor \frac{l^{+}}{N^{+}} \right\rfloor + \left\lfloor \frac{l^{-}}{N^{-}} \right\rfloor\right)^{c}\right)$$

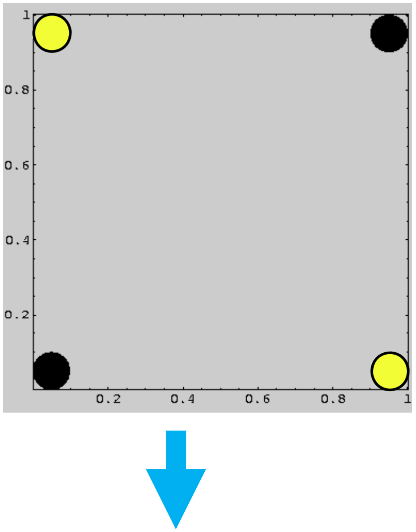
- Time complexity of M3-SVM in a serial way:

$$O\left(\frac{N^{2}}{N^{c}}\left(l^{+} + l^{-}\right)^{c}\right) \quad \text{suppose } N^{+} = N^{-} = N$$

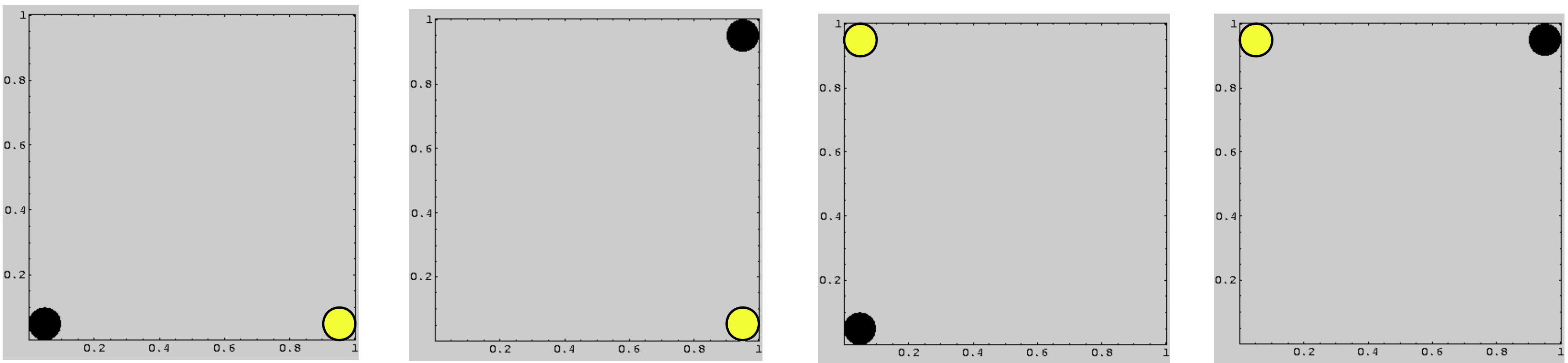
Advantages of part-versus-part method

- A large-scale two-class problem can be divided into a number of relatively smaller two-class problems
- A serious imbalanced two-class problem can be divided into a number of balance two-class problems
- Massively parallel learning can be easily implemented
- Domain/prior knowledge of training data can be incorporated into learning by dividing training data

Decomposition of XOR Problem



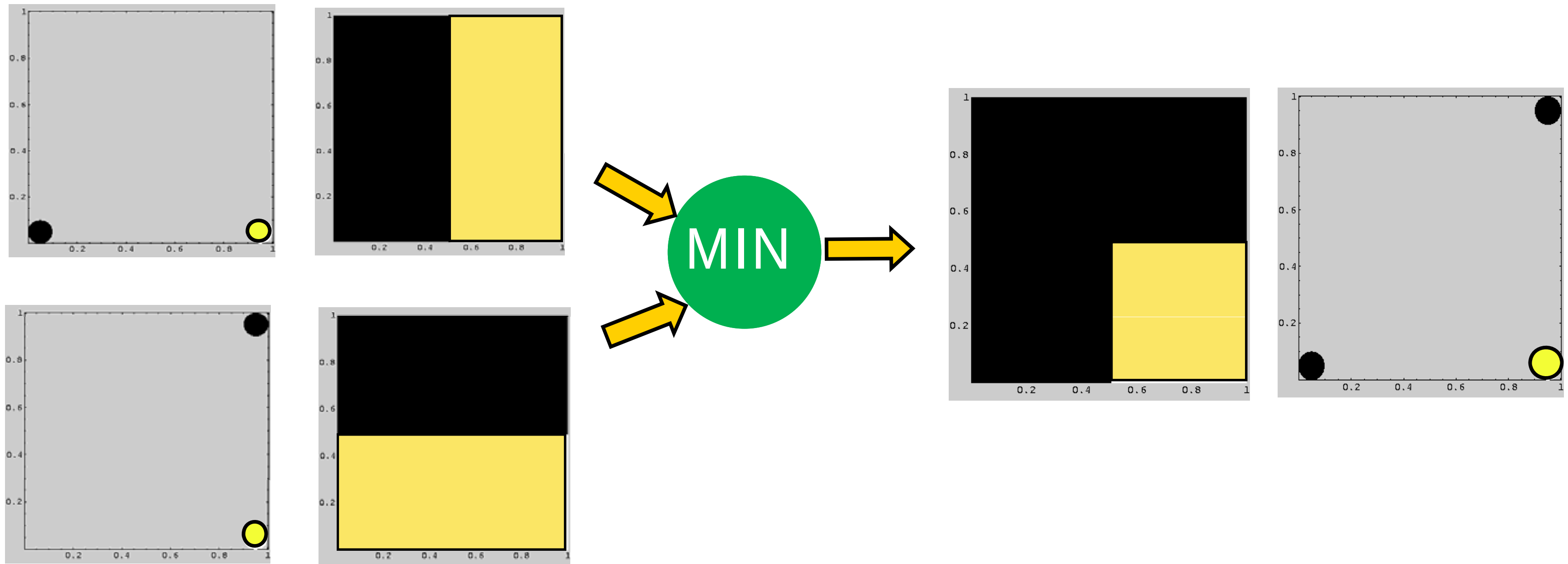
Four linearly separable problems



Two Module Combination Rules

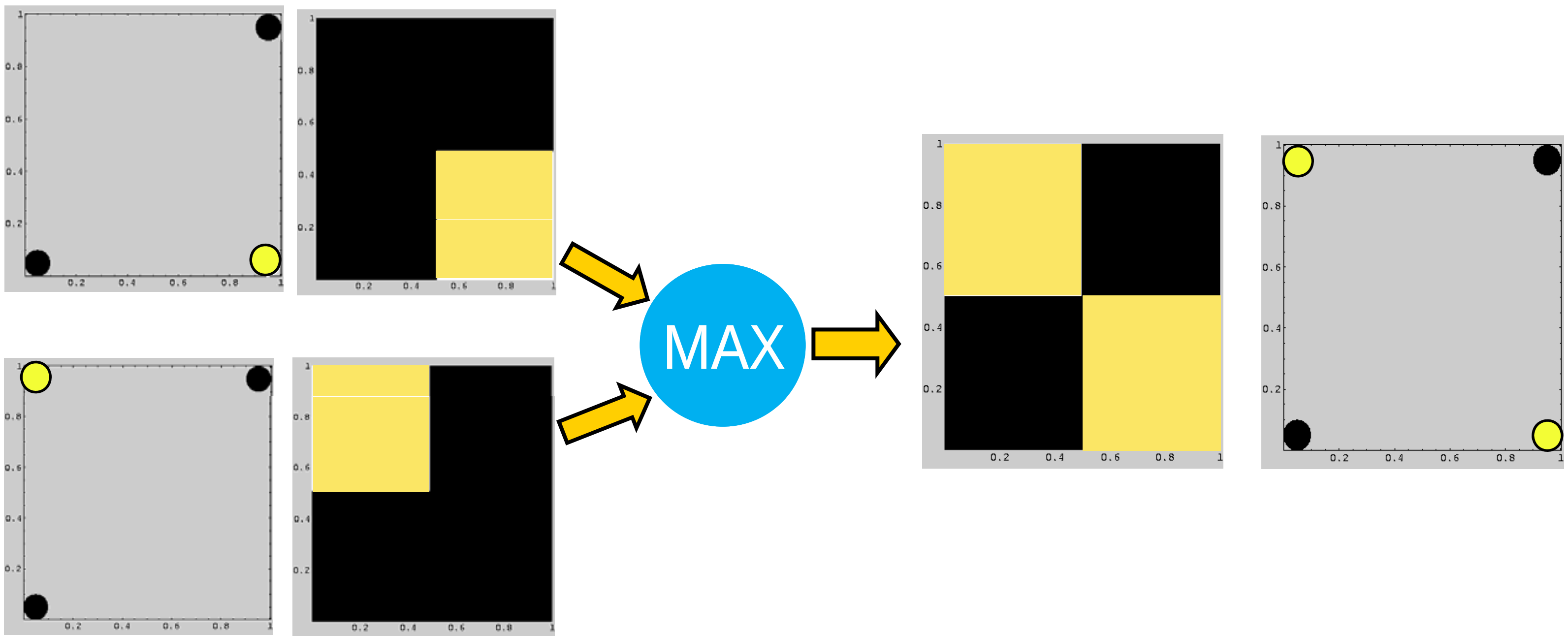
Combination rule : Minimization (AND gate)

The modules, which were trained on the data sets which have the same training inputs corresponding to desired output “1” (●), should be integrated by the MIN unit.

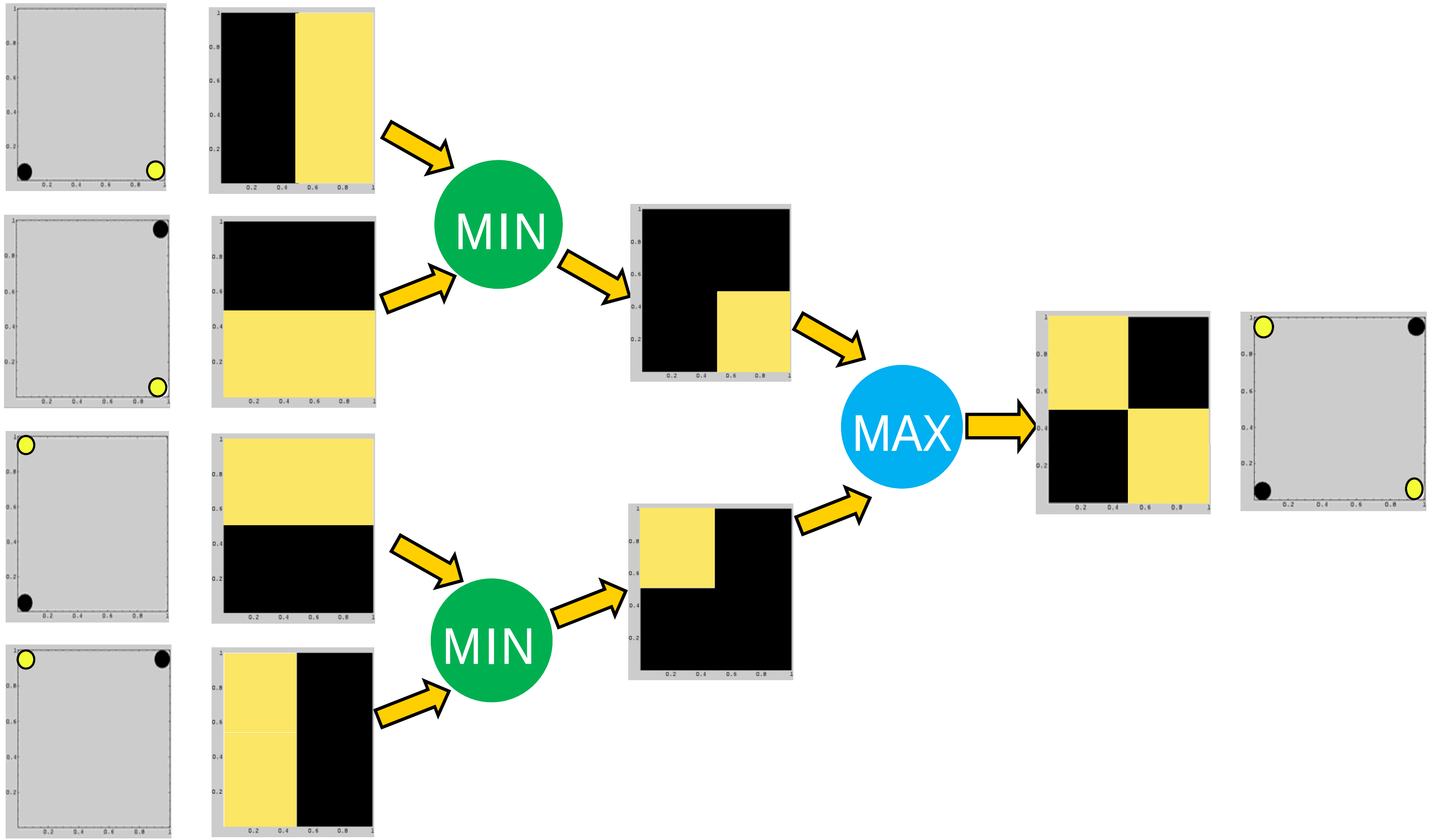


Combination rule: Maximization (OR gate)

The modules, which were trained on the data sets which have the same training inputs corresponding to desired output “0” (●), should be integrated by the MAX unit.



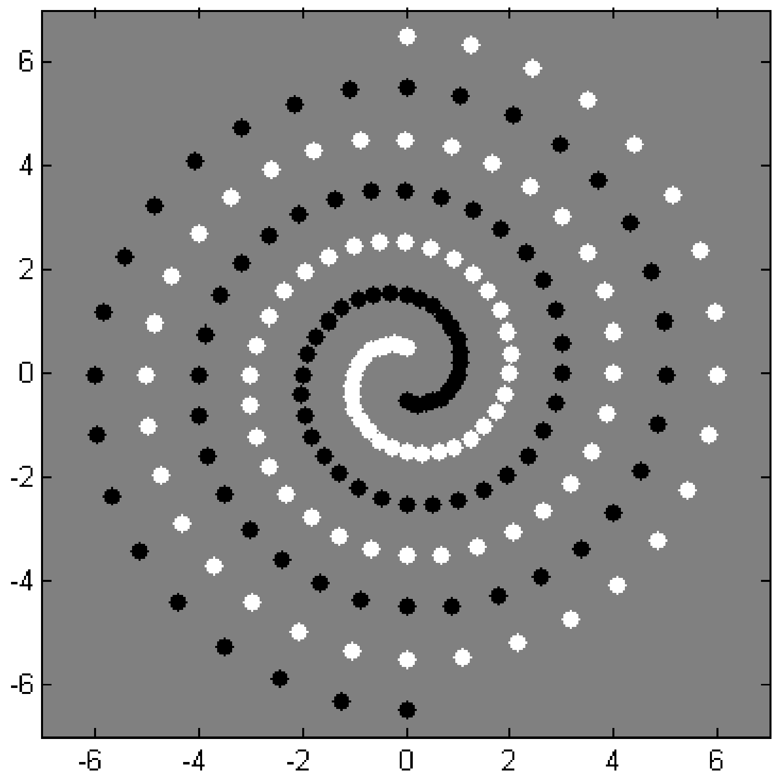
Combination of four modules for XOR problem



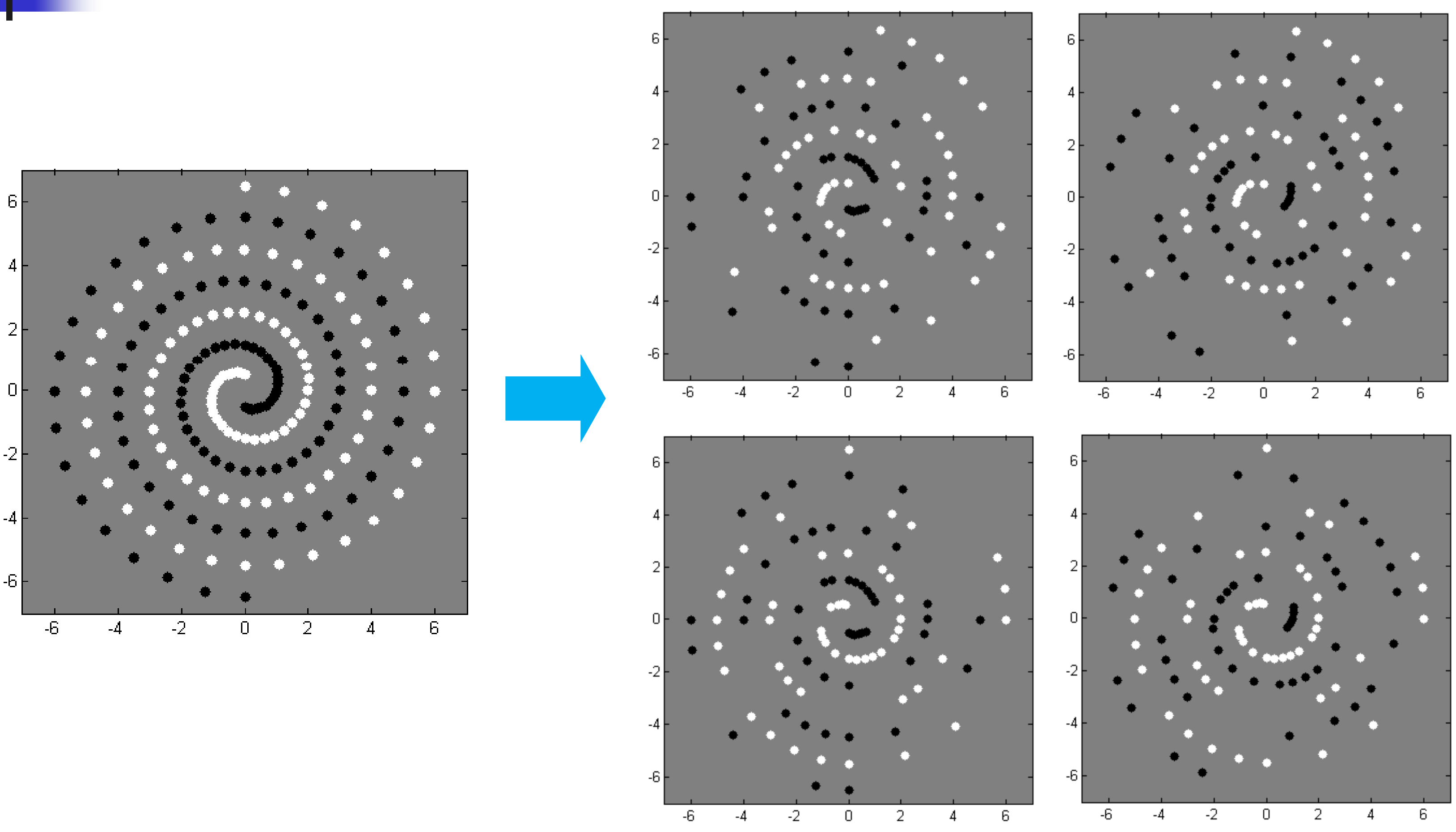
Task Decomposition Strategies

- Random (Lu & Ito, 1987)
- Hyperplane (Lu & Ito, 1987; Zhao & Lu, 2004)
- Equal-Clustering (Wen *et al*, 2005)
- Prior knowledge
  - Gender classification (Lian and Lu, 2006)
  - Age estimation (Lian and Lu, 2007)
  - Patent classification (Lu and Wang, 2008)
  - Protein subcellular localization (Yang and Lu, 2009)

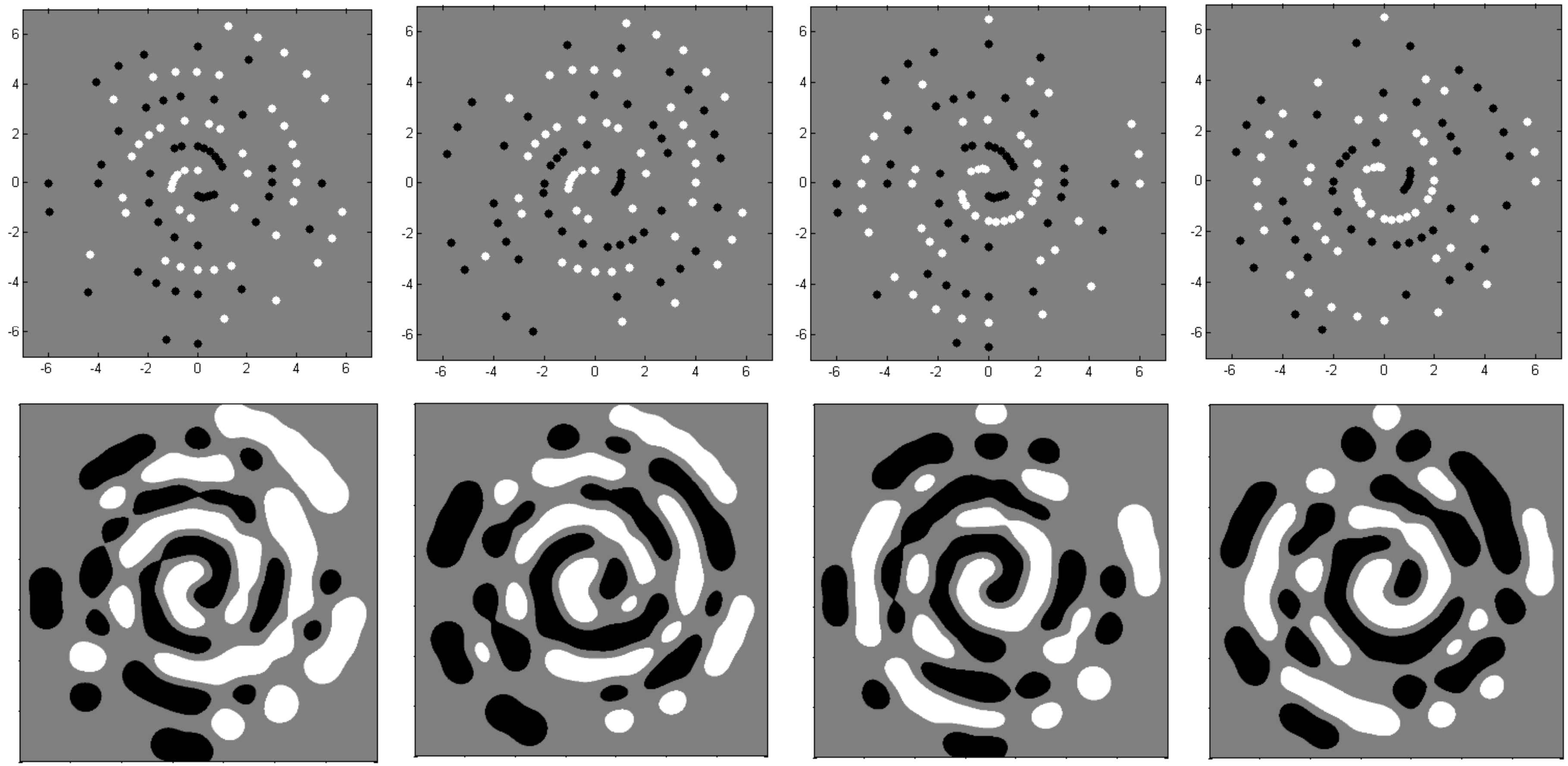
Two-spirals problem



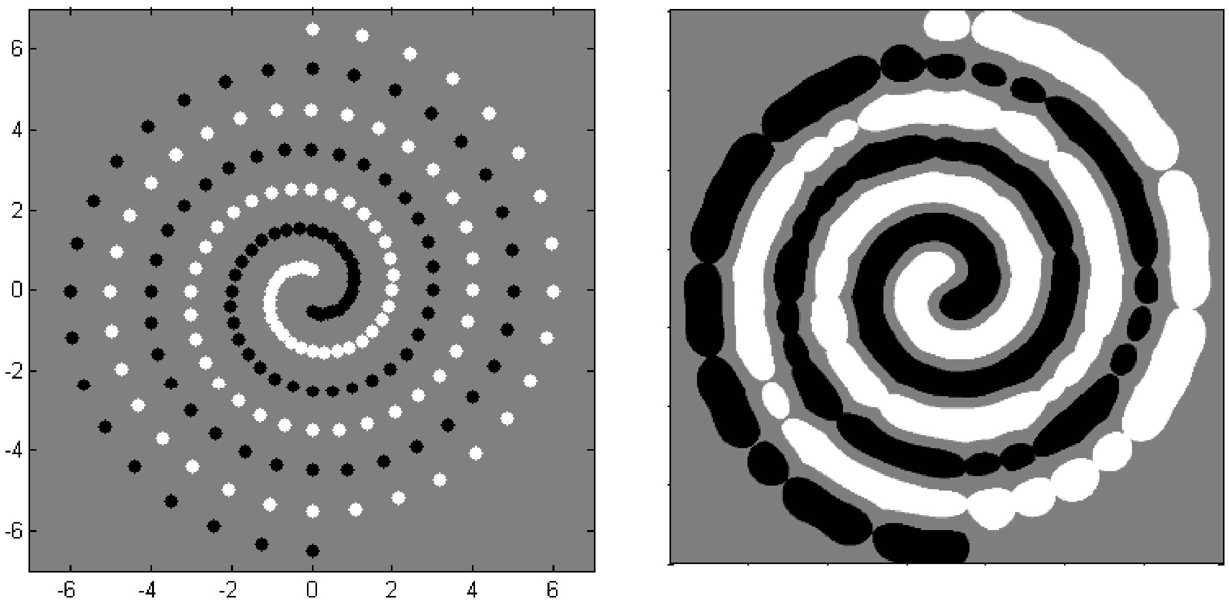
Random Partition



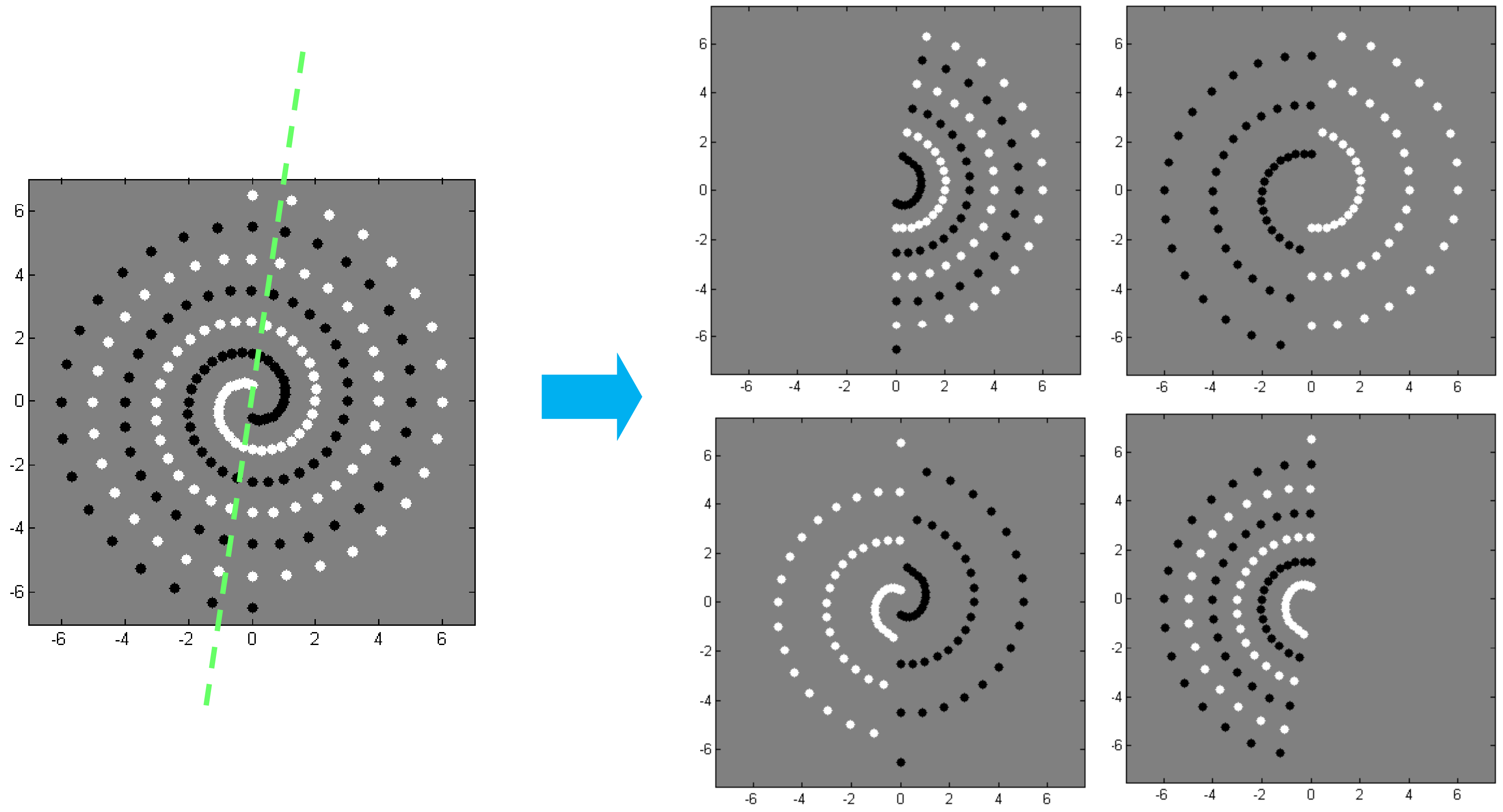
Subproblems and trained MLP module



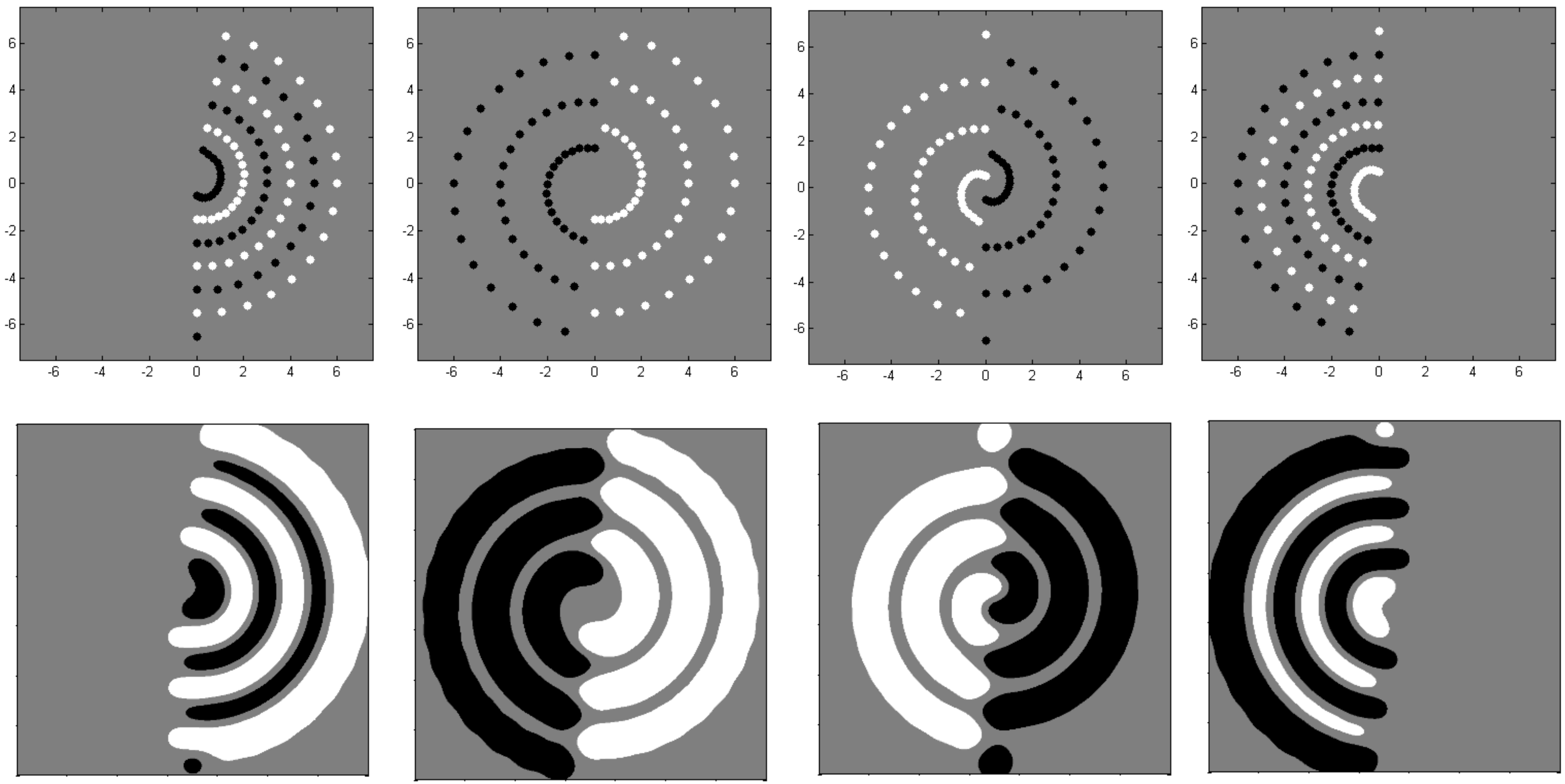
Learning result with random decomposition



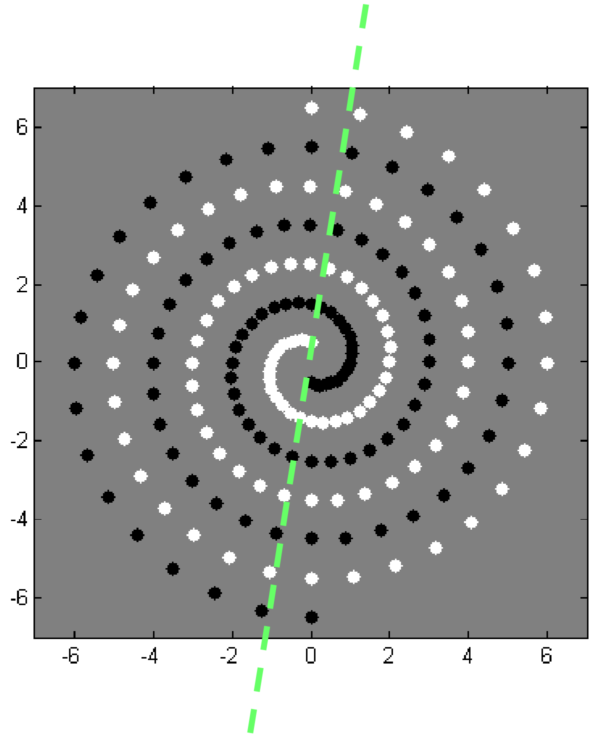
Hyper-plane task partition (Lu & Ito, 1997)

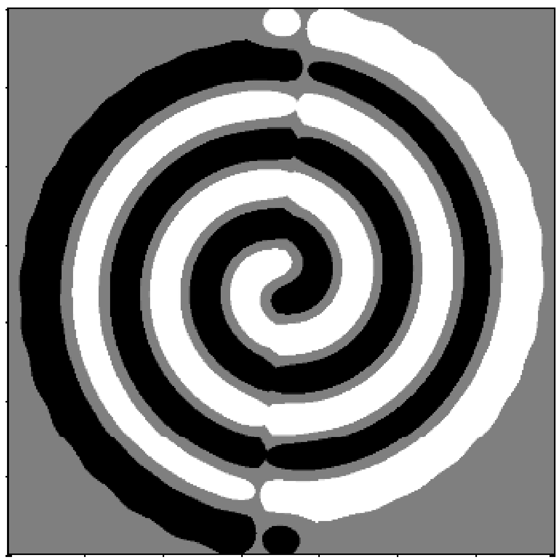


Subproblems and trained MLP modules

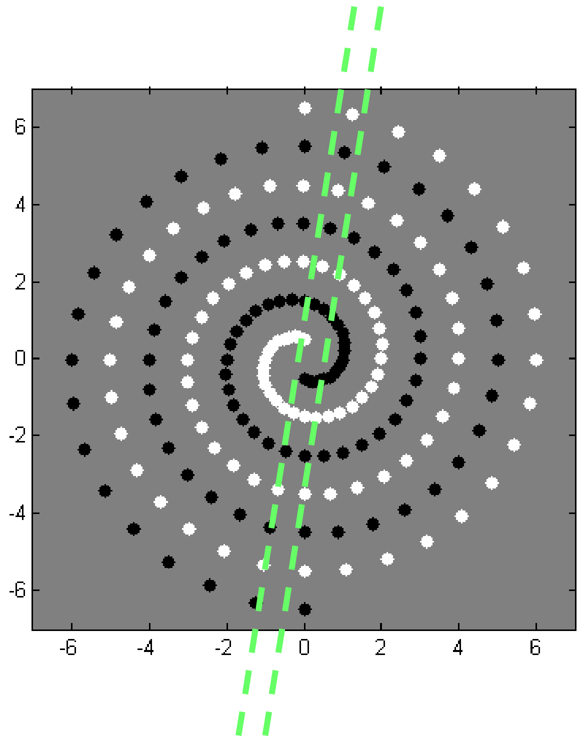


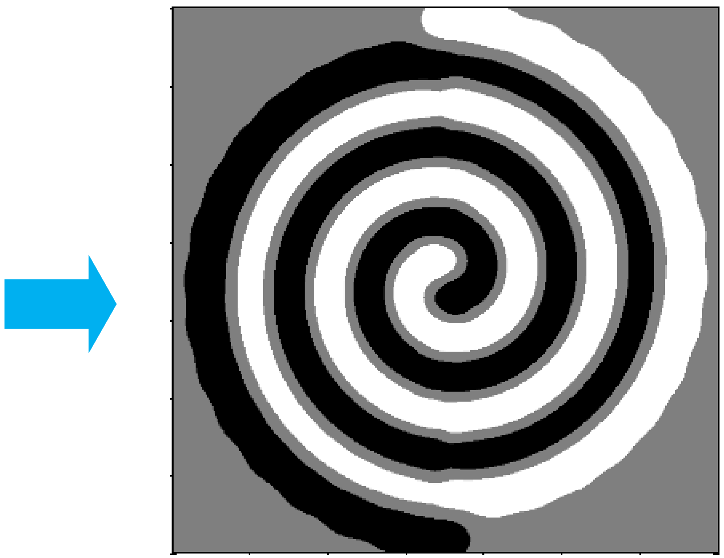
M^3-MLP with hyper-plane decomposition





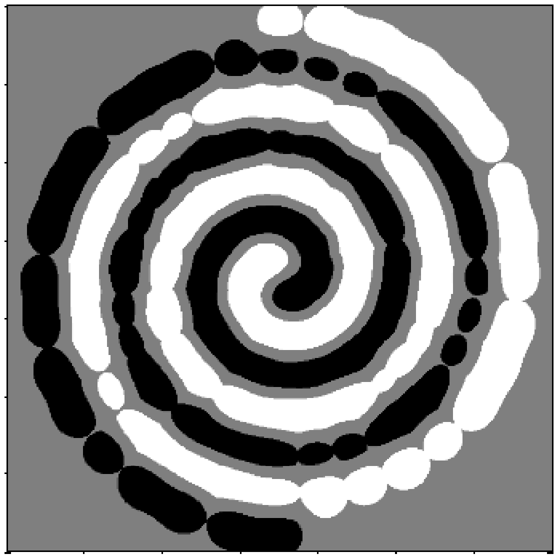
Hyper-plane partition with overlapping



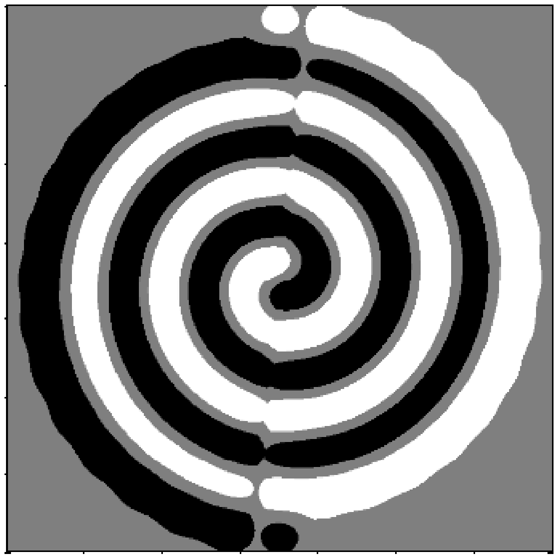


Overlapping means two subsets share the training data around the hyperplance

Three different partition strategies



Random

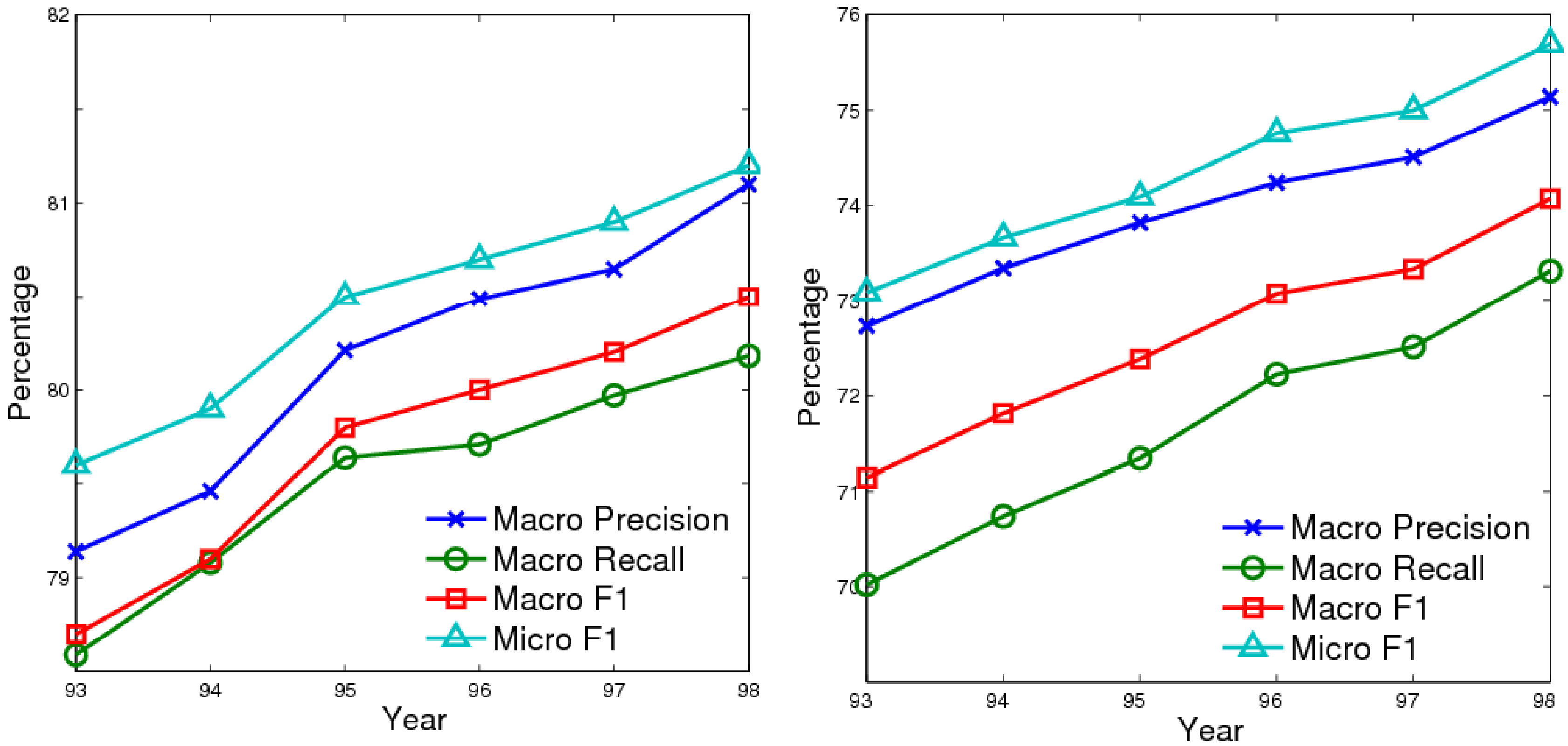


Hyper-plane

Hyper-plane with overlapping

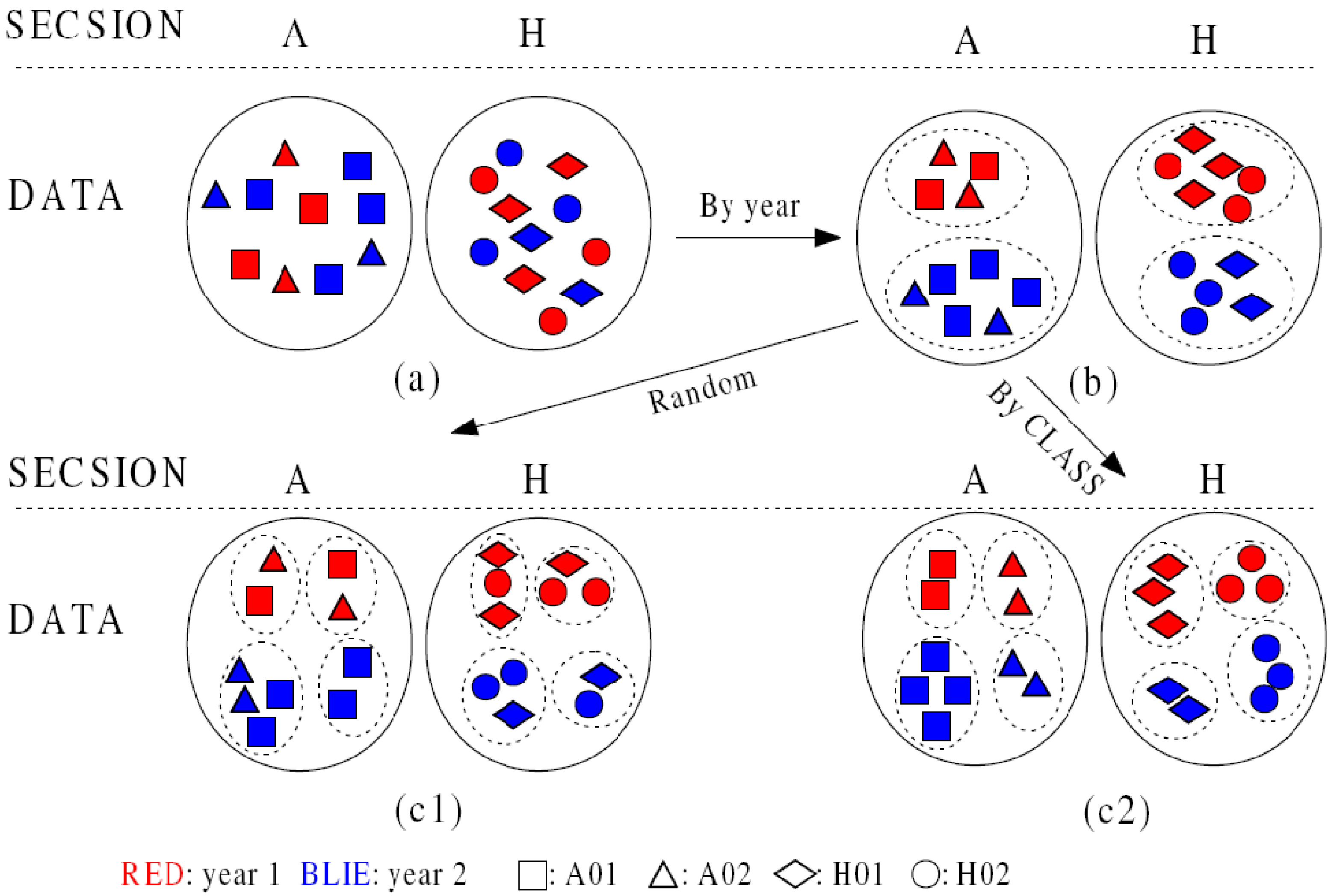
Incorporating prior knowledge into classifying Japanese Patents

Time-Varying Features of Patents



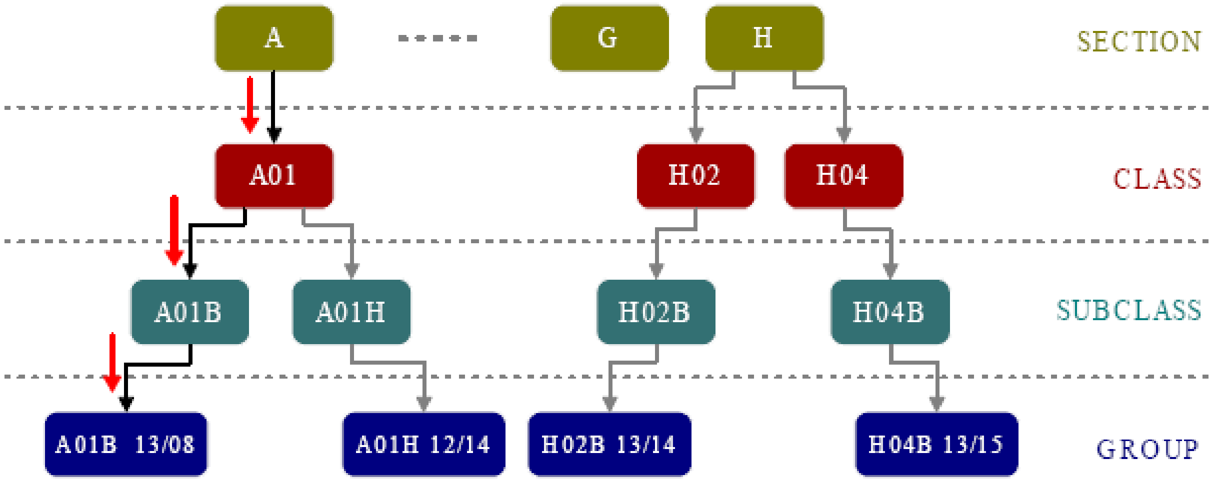
The data of 1999 as training data other years as test data

Year-class decomposition strategy

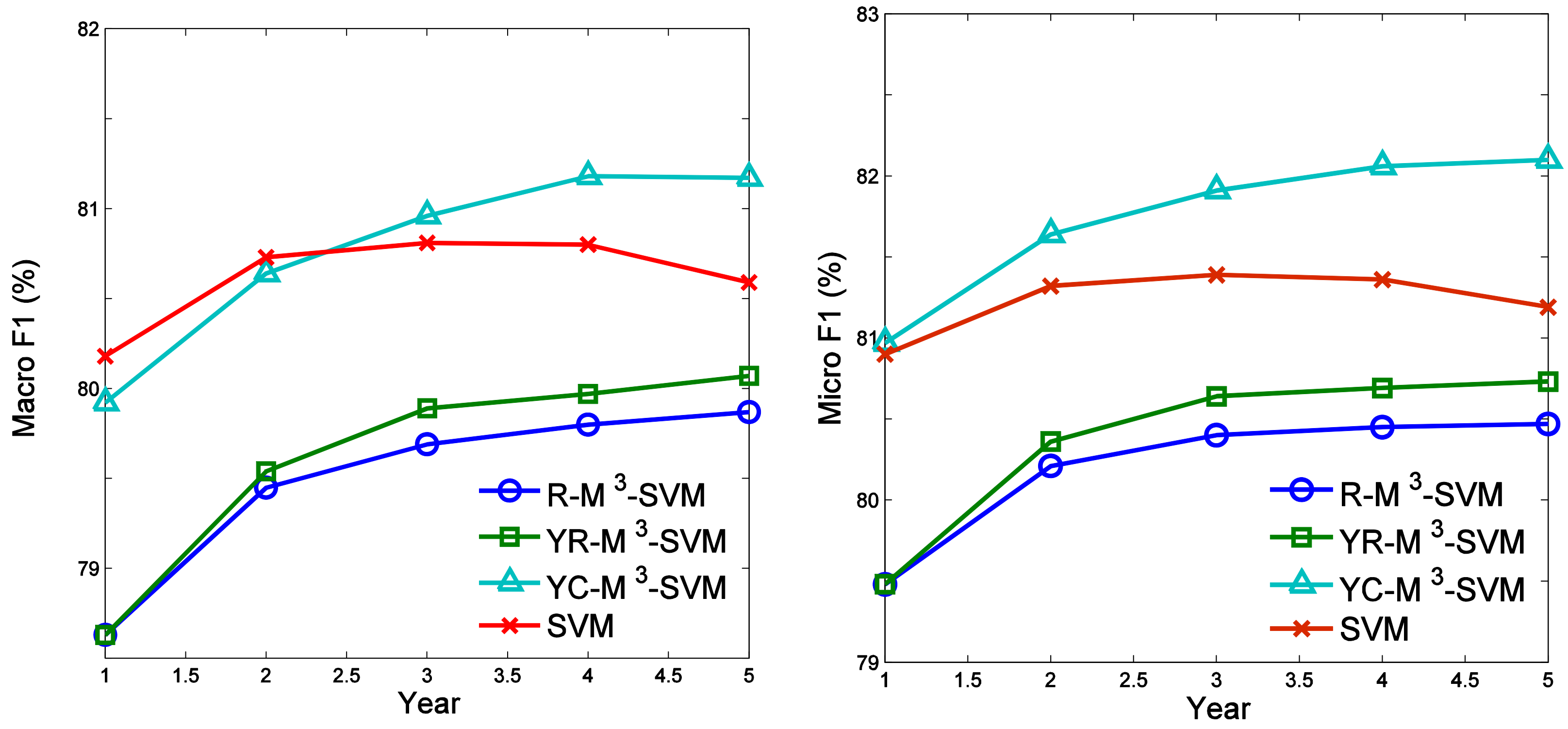


Performance comparison

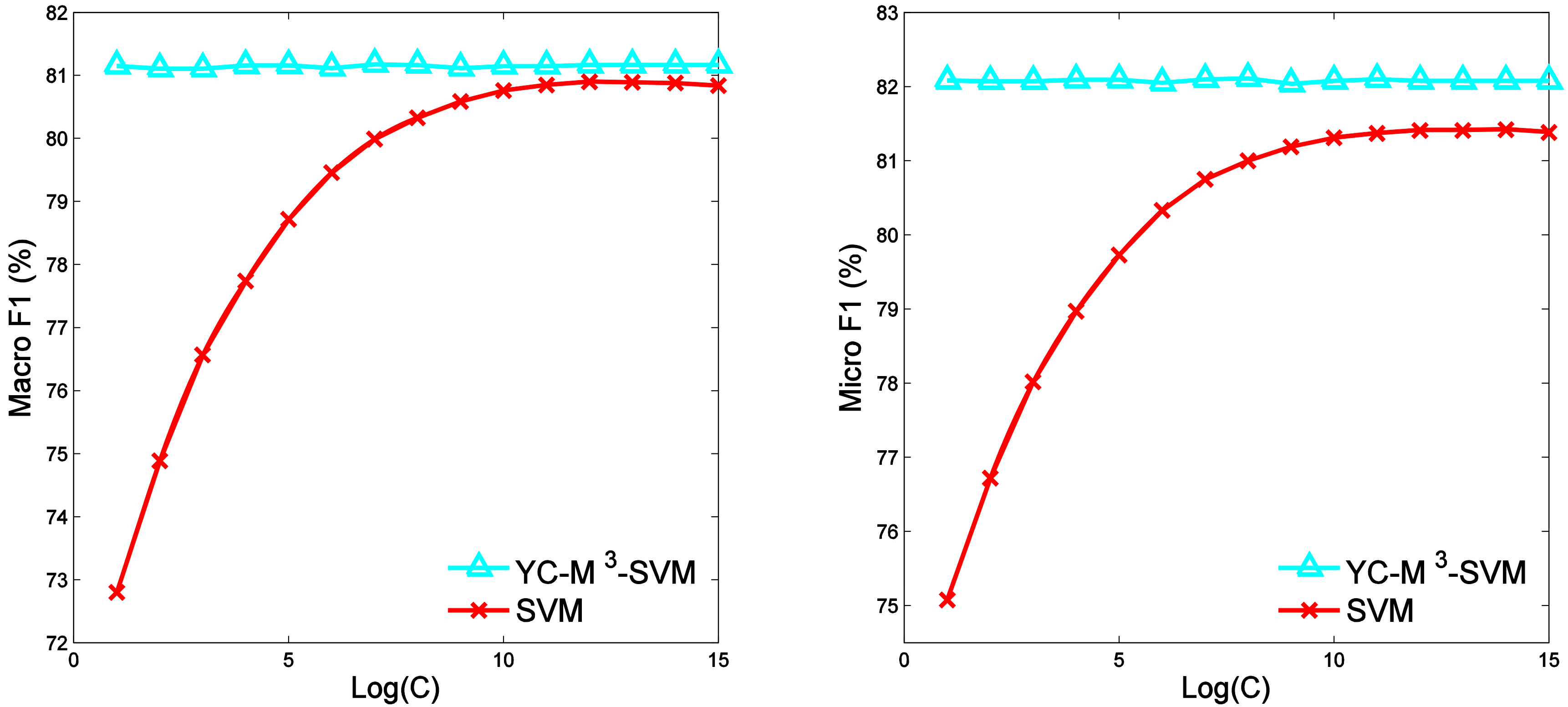
- R-M3-SVM, decompose task randomly
- YR-M3-SVM, decompose task only by year
- YC-M3-SVM, decompose task by year and class
- Conventional SVMs are selected as a baseline



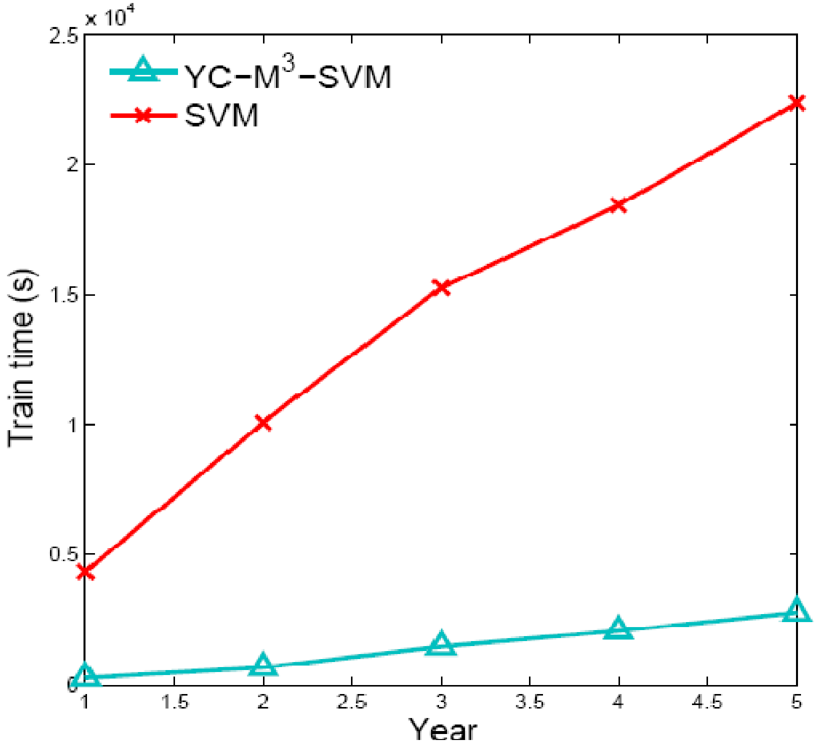
Performance comparison

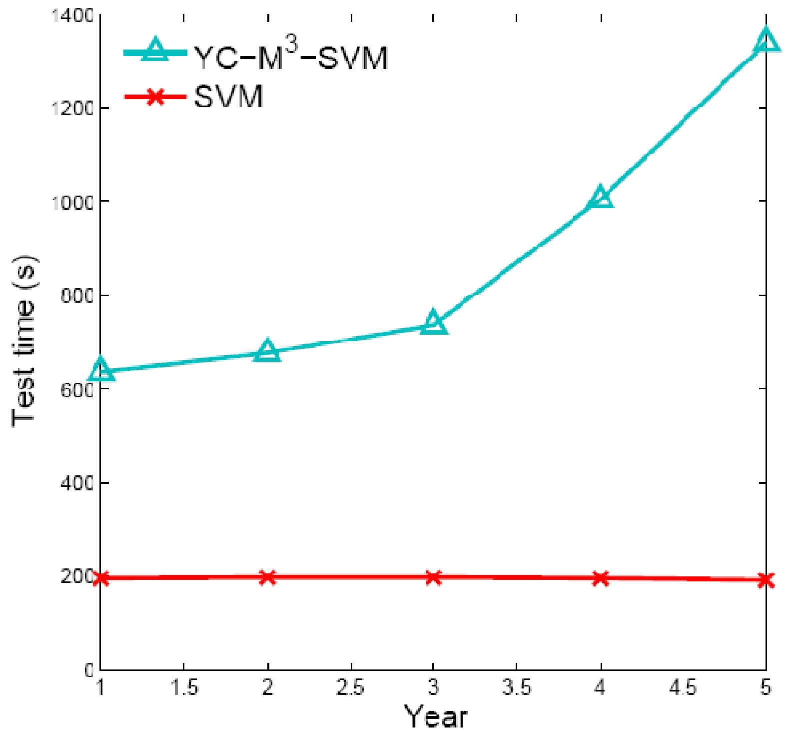


Performance variation with changes of C



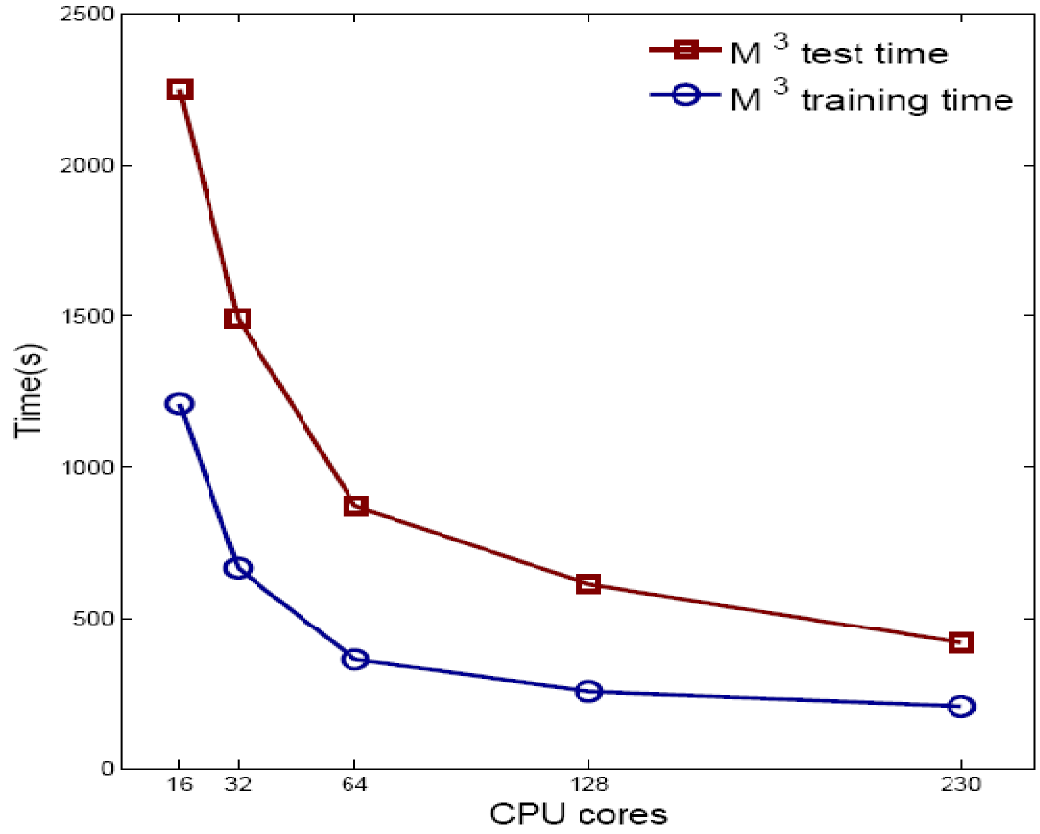
Comparison of training and test time



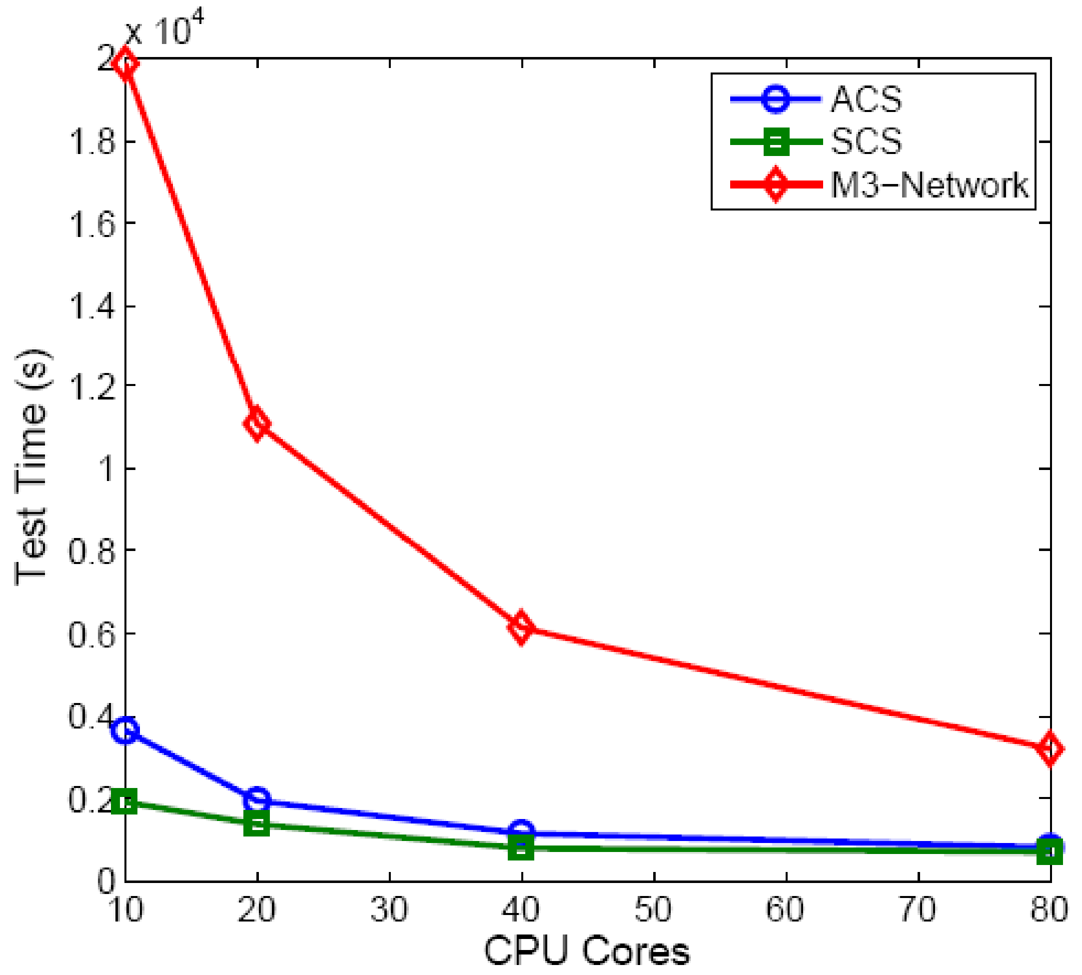


Here SVM with linear kernel was used

Scalability of our approach

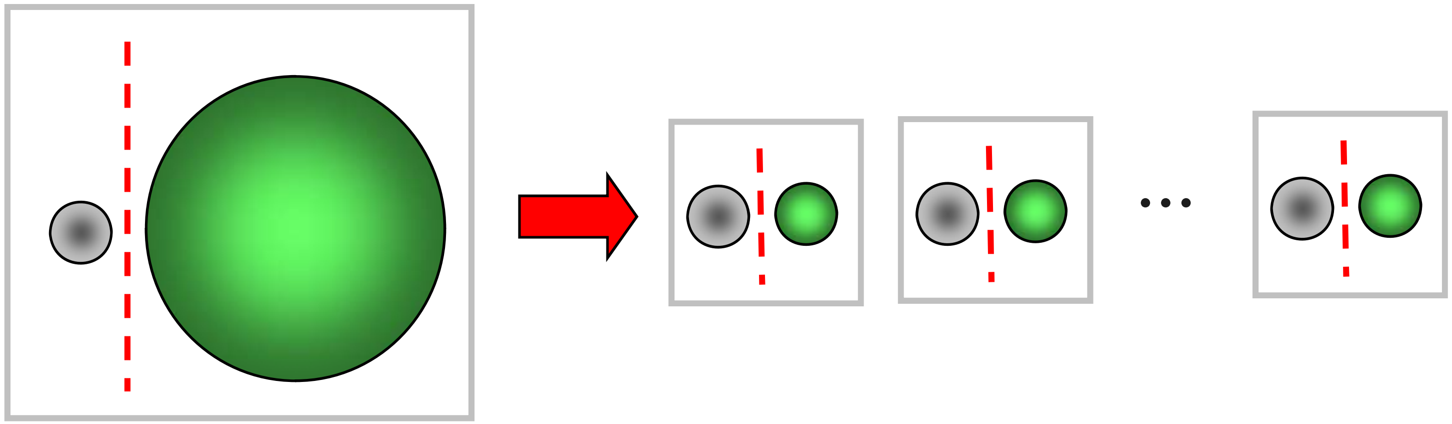


Comparison of three combination methods



Balance training data by using PVP

Divide an imbalance two-class problem into a number of relatively more balance and smaller two-class subproblems.



Data sets

- UCI benchmark
  - Abalone data : 487 vs. 3,690 (1:7.6)
  - K=29; 11 versus all
- Looftop data
  - 781 vs. 17,084 (1:21.8)
- Protein subcellular localization (Park *et al.*)
  - 861 vs. 6,718 (1:7.8)
  - K=12 ; 4 versus all

Experiment Results (1)

Abalone	TP(%)	TN(%)	AUC
C5.0	61.5	59.6	66.84
CSVM	59.0	58.8	64.25
C5.0 + SMOTE	64.5	62.4	69.53
M3SVM	67.5	66.4	72.67

(Ye et al, 2009)

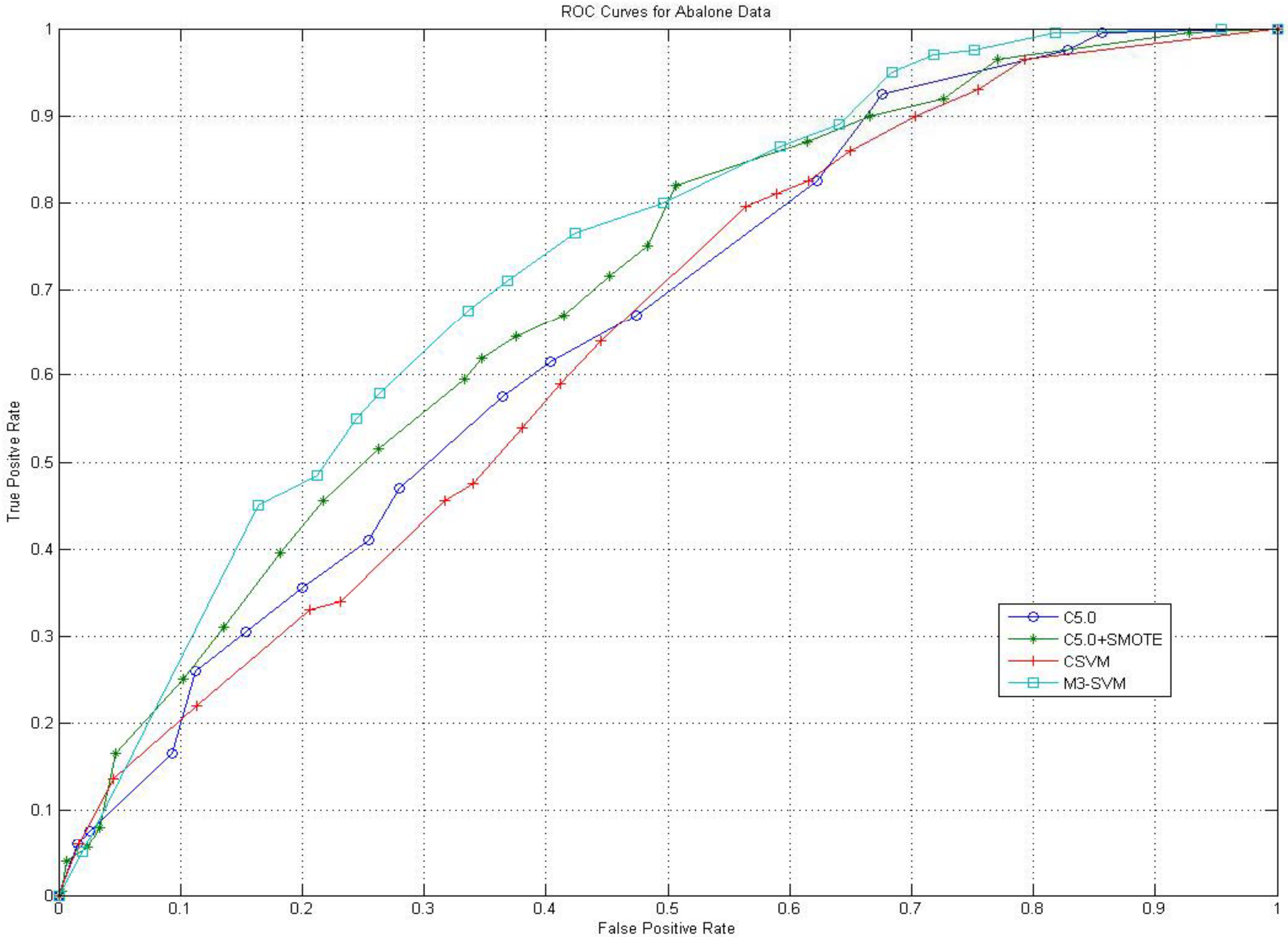
Experiment Results (2)

Rooftop	TP(%)	TN(%)	AUC
C5.0	78.5	80.2	87.43
CSVM	78.1	79.8	83.98
C5.0 + SMOTE	79.9	80.1	88.22
M3SVM	81.6	81.4	89.28

Experiment Results (3)

Park	TP(%)	TN(%)	AUC
C5.0	82.6	85.8	90.39
CSVM	84.9	85.5	93.93
C5.0 + SMOTE	84.3	83.8	90.96
M3SVM	87.2	87.7	94.54

ROC Curve for Abalone data



Conclusions

- M3-network enables us to easily incorporate prior knowledge into learning
- Incorporating time information into task decomposition can reduce the influence of time-varying features.
- Incorporating time and hierarchical structure information into learning has the best performance.
- The lower time cost of our parallel system is important for training on large data sets.

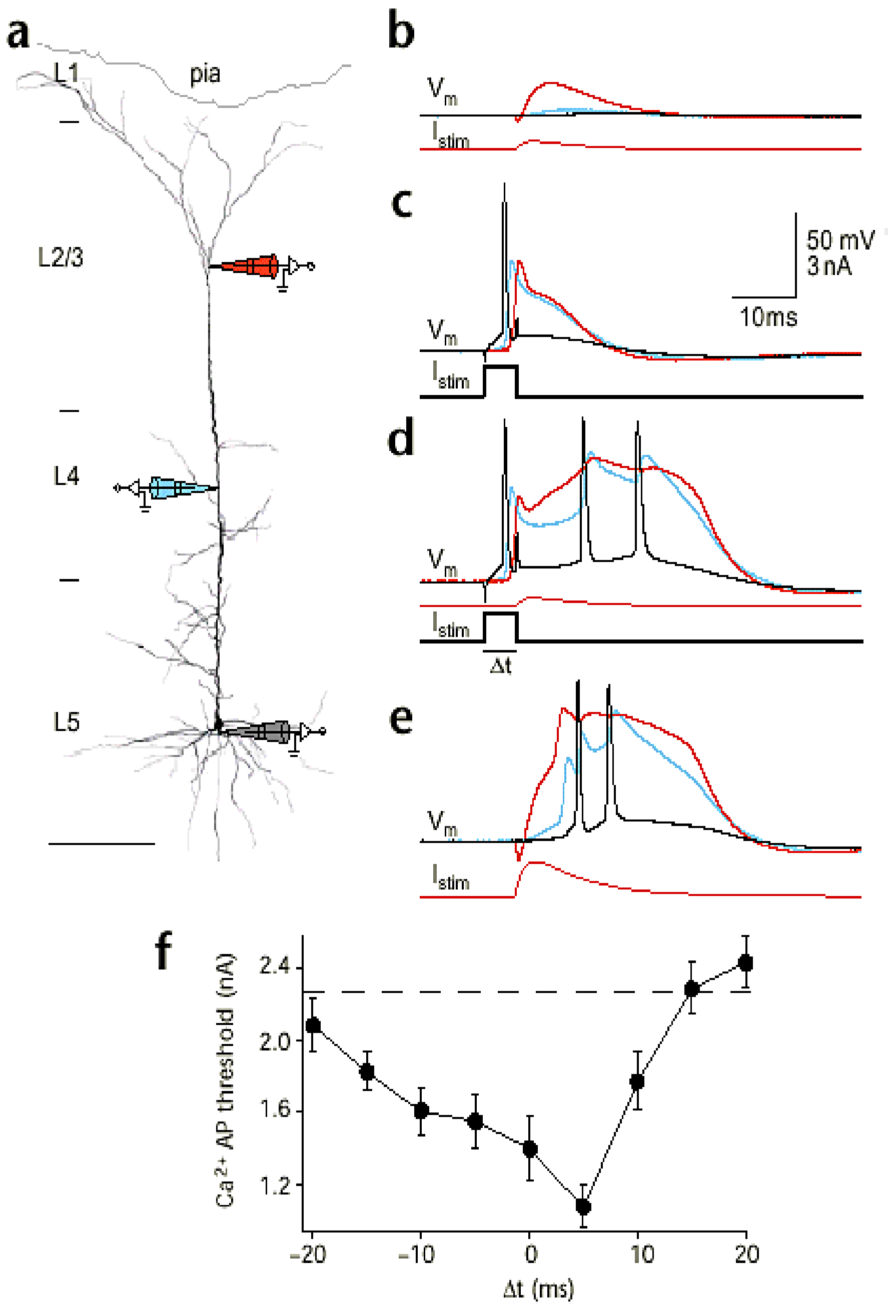
Towards Brain-Like Computing

Back-propagation (BP) algorithm in 1989

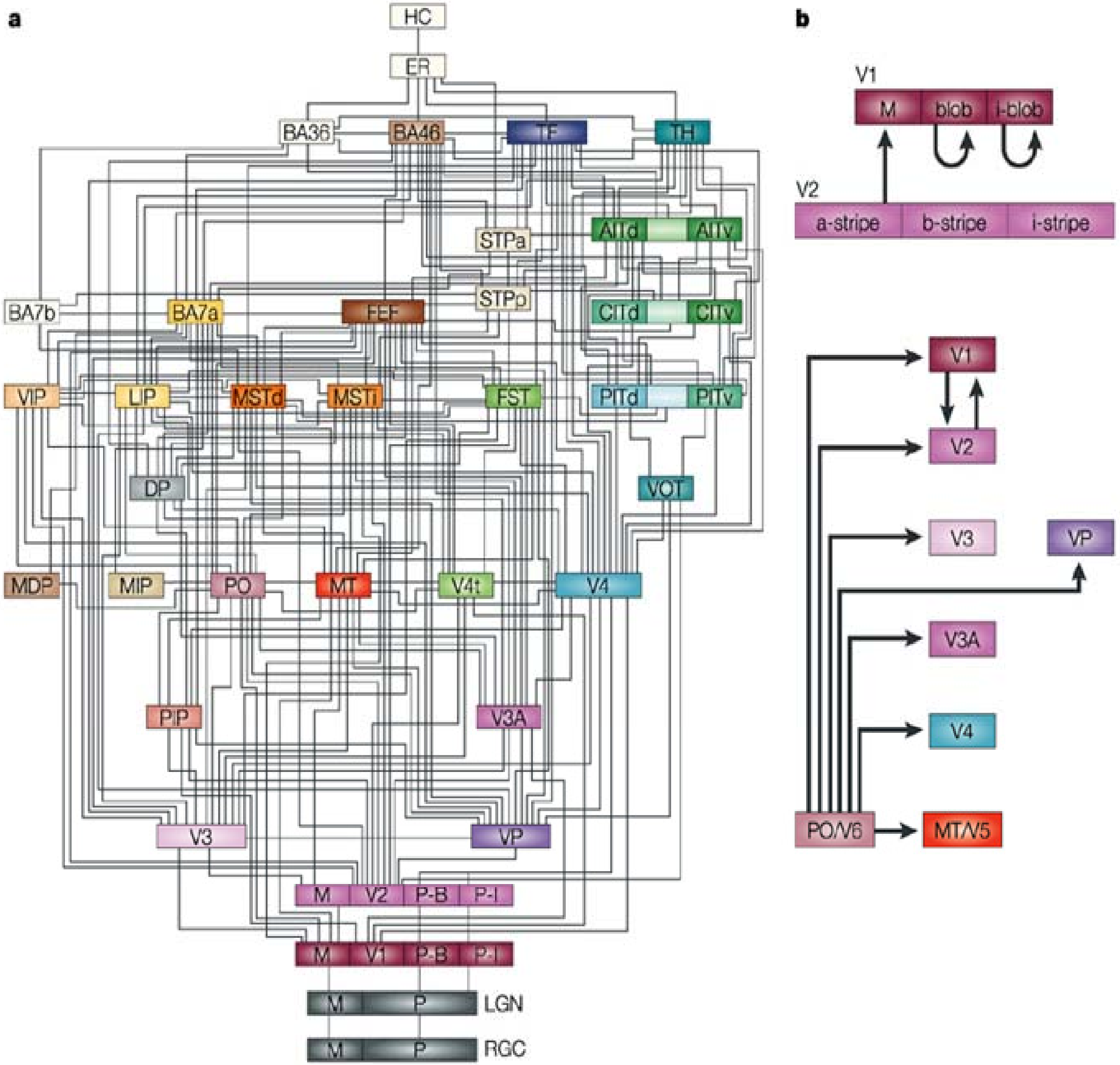
$\approx$

Support vector machine in 2009

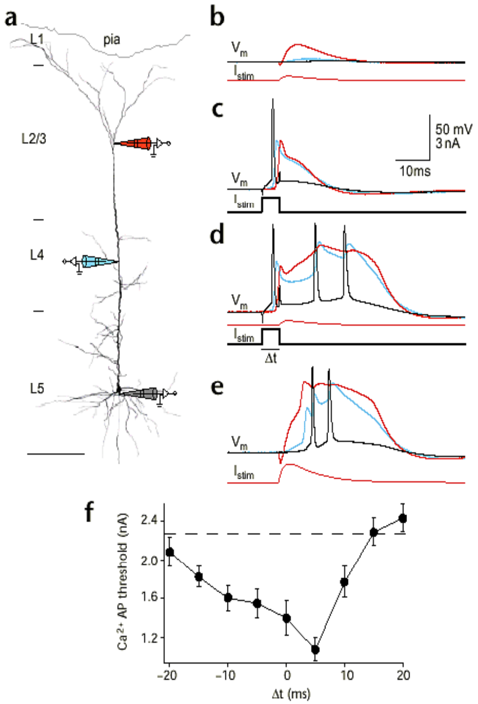
Static + Statistic → Dynamic + Domain knowledge

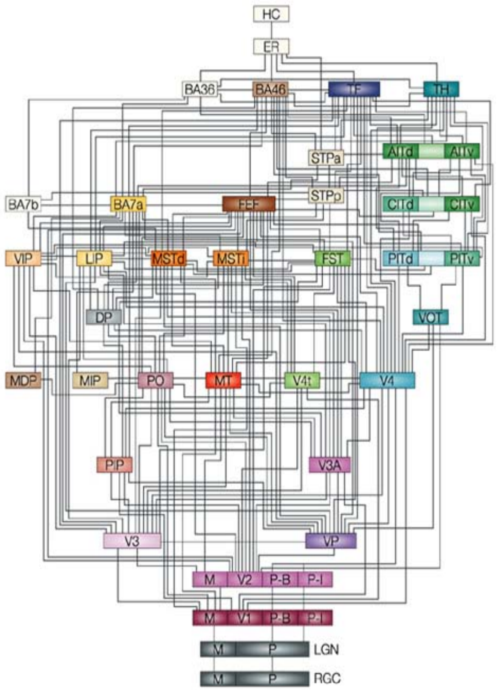


a
L1
pia
L2/3
L4
L5
b
Vm
Istim
c
50 mV
3 nA
10ms
Vm
Istim
d
Vm
Istim
Δt
e
Vm
Istim
f
Ca2+ AP threshold (nA)
2.4
2.0
1.6
1.2
-20
-10
0
10
20
Δt (ms)



Nature Reviews | Neuroscience





Our knowledge of organizing
neurons in system level
is rather poor !

Emergence: From Chaos to Order

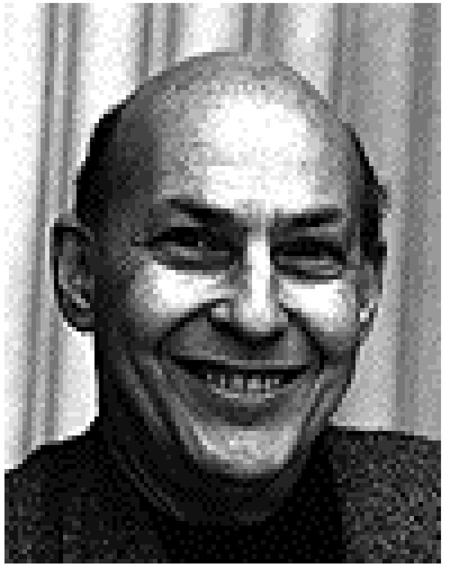
John H. Holland (1998)

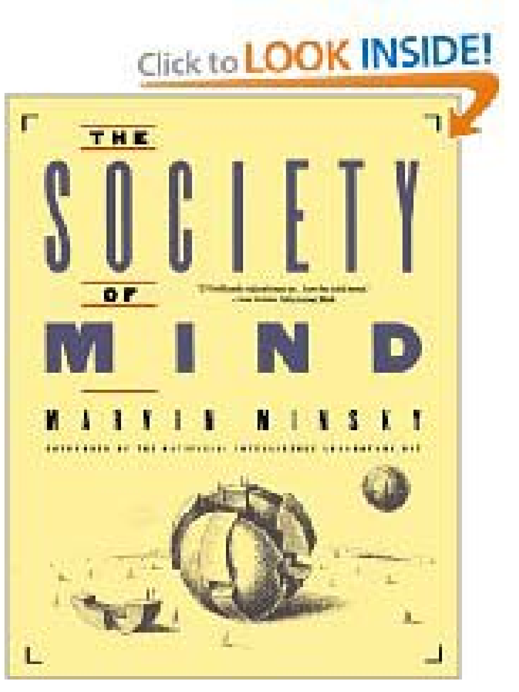
A Theory of Emergence

“We are everywhere confronted with emergence in complex adaptive systems: ant colonies, network of neurons, Internet ...., where the behavior of the whole is much more complex than the behavior of the parts.”

J. H. Holland, Emergence: From Chaos to Order (1998)

Marvin Minsky (1986)





Emergence of Intelligence

“This book tries to explain how minds work. How can intelligence emerge from non-intelligence ? To answer that, we’ll show that you can build a mind from many little parts, each mindless by itself”

Acknowledgments

- SJTU

  Hai Zhao
  Yi-Min Wen
  Hui-Cheng Lian
  Jing Li
  Yang Yang
  Jun Luo
  Zhi-Fei Ye
  Chao Ma
  Yue Wang
  Qi Kong

- Grants:

  NSFC
  973
  863

  Riken
  NICT
  Microsoft
  Hitachi
  Fujitsu
  Omron

Seventh International Symposium on Neural Networks (ISNN2010)

- Date: June 7-10, 2010
- Venue: Xinya Hotel, Nanjing Road, Shanghai
- General Chairs: Jun Wang and Bao-Liang LU
- Program Chairs: Li-Qing Zhang, James Kwok, and Zhi-Gang Zeng
- Proceedings: LNCS, Springer
- Special Issues: Neurocompuing
- Web: http://isnn2010.sjtu.edu.cn
- Email: isnn2010@sjtu.edu.cn

Deadline: December 1st, 2009
Welcome to submit your paper!

Thank You !