Support Vector Machines for Pattern Classification

Shigeo Abe

Graduate School of Science and Technology
Kobe University
Kobe, Japan

My Research History on NN, FS, and SVM

- **Neural Networks (1988-)**
  - **Convergence characteristics of Hopfield networks**
  - **Synthesis of multilayer neural networks**

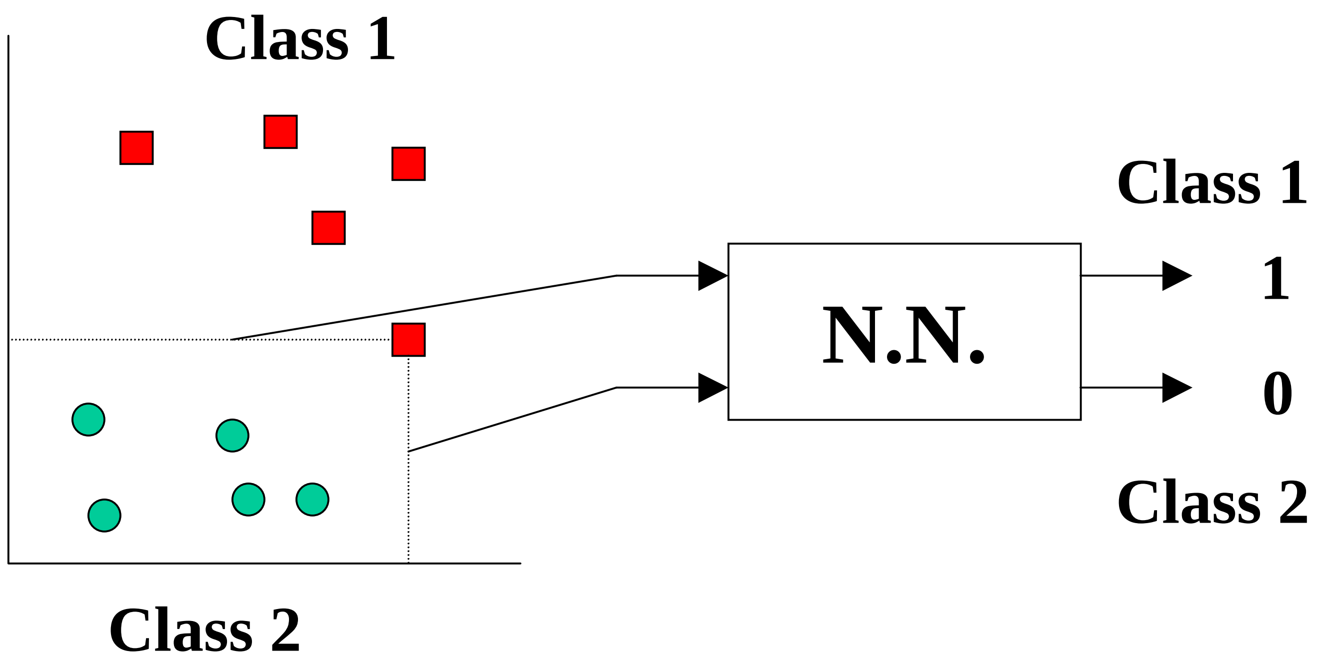
- **Fuzzy Systems (1992-)**
  - **Trainable fuzzy classifiers**
  - **Fuzzy classifiers with ellipsoidal regions**

- **Support Vector Machines (1999-)**
  - **Characteristics of solutions**
  - **Multiclass problems**

Contents

Multilayer Neural Networks

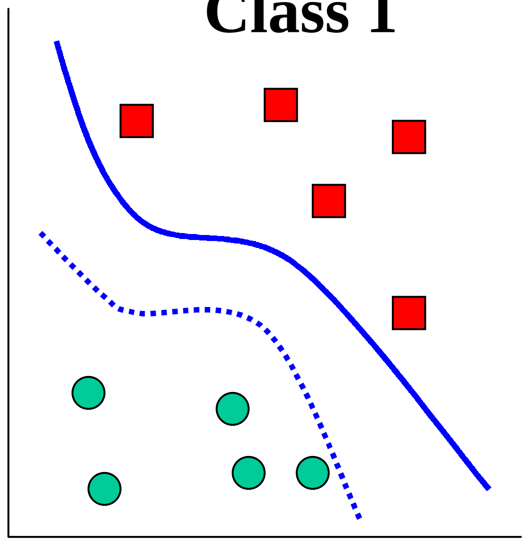
Neural networks are trained to output the target values for the given input.



Multilayer Neural Networks

Indirect decision functions: decision boundaries change as the initial weights are changed.

Class 1



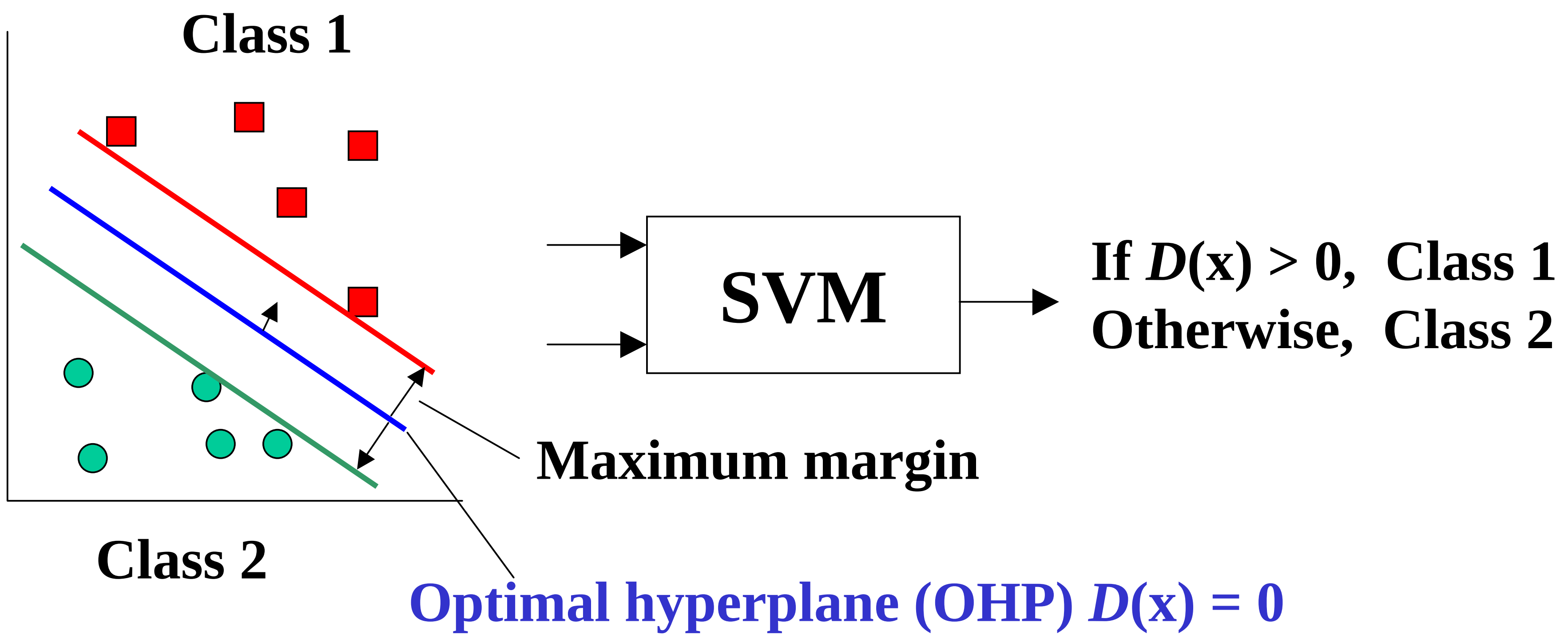
Class 2

Class 1

Class 2

Support Vector Machines

Direct decision functions: decision boundaries are determined to minimize the classification error of both training data and unknown data.



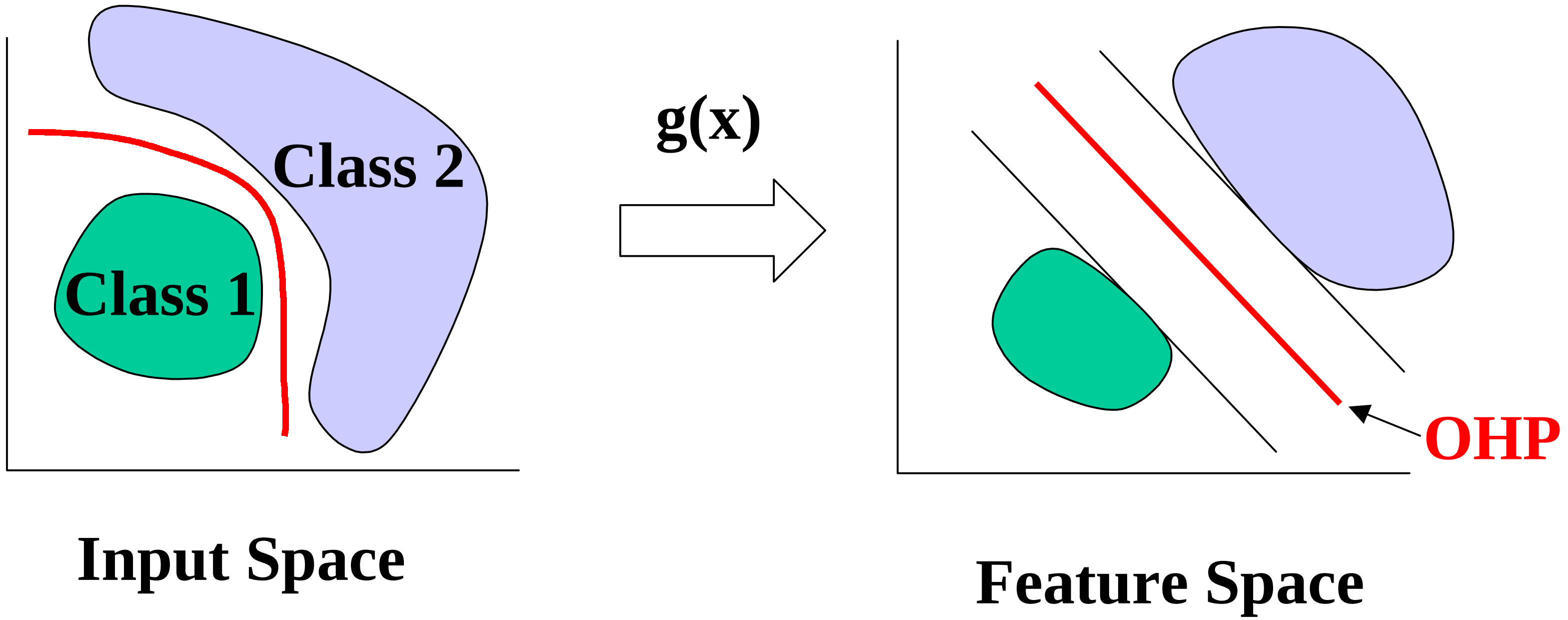
Summary

- When the number of training data is small, SVMs outperform conventional classifiers.
- By maximizing margins performance of conventional classifiers can be improved.

Contents

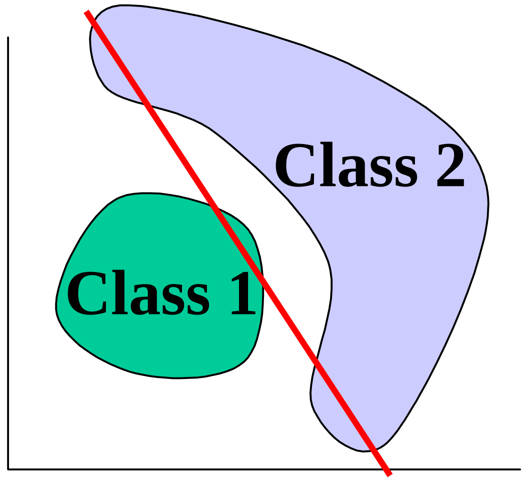
Architecture of SVMs

- Formulated for two-class classification problems
- Map the input space into the feature space
- Determine the optimal hyperplane in the feature space



Types of SVMs

- **Hard margin SVMs**
  - linearly separable in the feature space
  - maximize generalization ability
- **Soft margin SVMs**
  - Not separable in the feature space
  - minimize classification error and maximize generalization ability
    - L1 soft margin SVMs (commonly used)
    - L2 soft margin SVMs



Feature Space

Hard Margin SVMs

We determine OHP so that the generalization region is maximized:

maximize δ

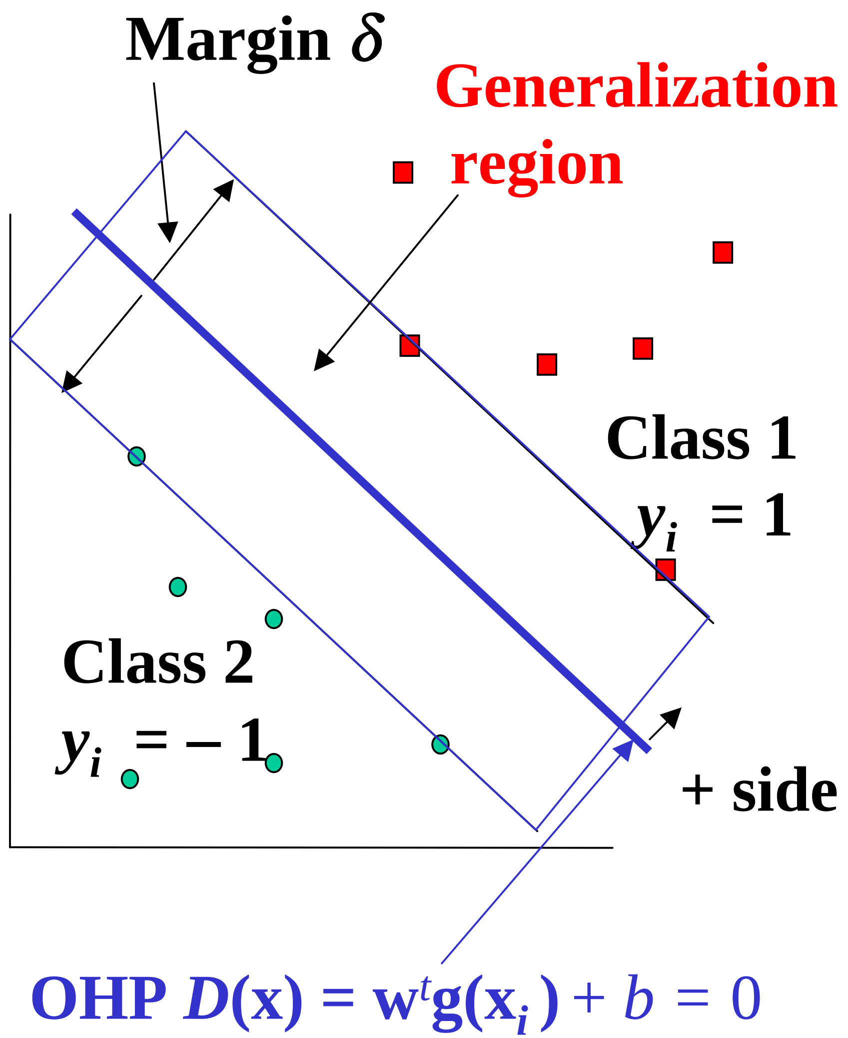
subject to

$$\mathbf{w}^t \mathbf{g}(\mathbf{x}_i) + b \geqq 1 \text{ for Class 1}$$

$$-\mathbf{w}^t \mathbf{g}(\mathbf{x}_i) - b \geqq 1 \text{ for Class 2}$$

Combining the two:

$$y_i\,(\mathbf{w}^t \mathbf{g}(\mathbf{x}_i) + b) \geqq 1$$



Hard Margin SVM (2)

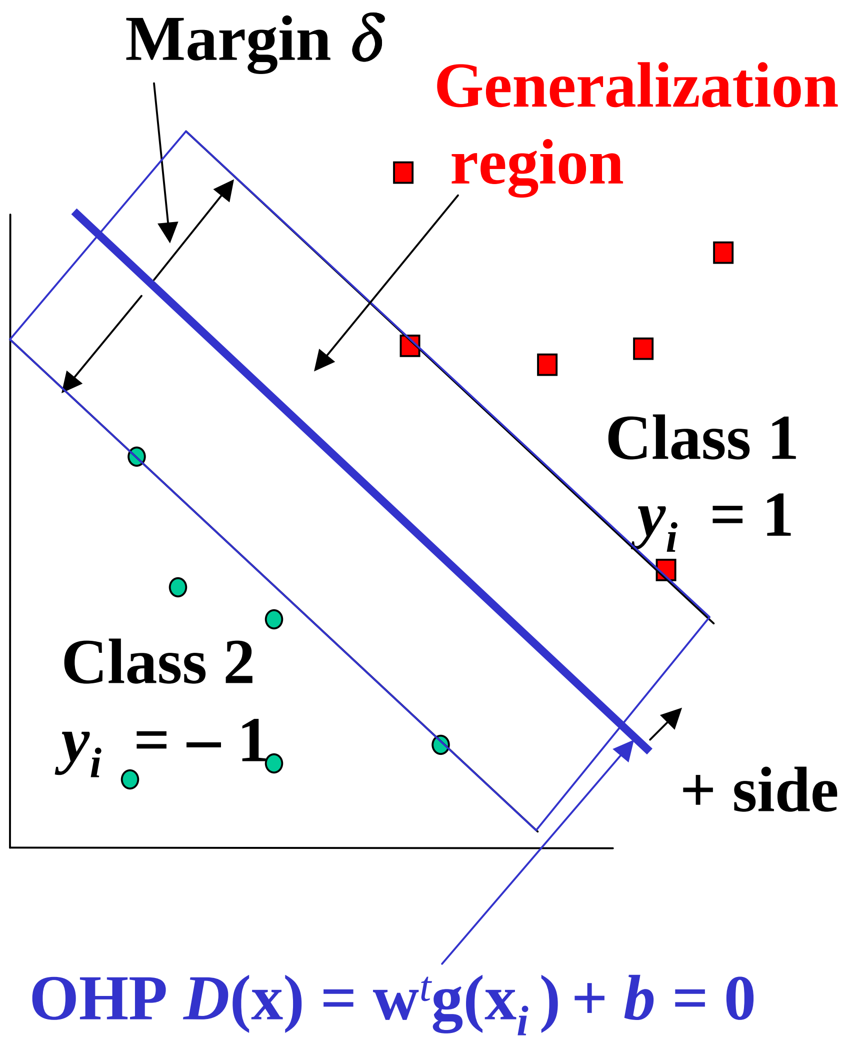
The distance from x to OHP is given by $y_i\, D(\mathbf{x}) / \|\mathbf{w}\|$. Thus all the training data must satisfy

$$y_i\, D(\mathbf{x}) / \|\mathbf{w}\| \geqq \delta .$$

Imposing $\|\mathbf{w}\|\, \delta = 1$, the problem is to minimize $\|\mathbf{w}\|^2/2$

subject to

$$y_i\, D(\mathbf{x}_i) \geqq 1 \quad \text{for } i = 1,\ldots,M.$$



Soft Margin SVMs

If the problem is non-separable, we introduce slack variables ξ_i.

Minimize

$$\|w\|^2/2 + C/p \, \Sigma_{i=1,M} \, \xi_i^p$$

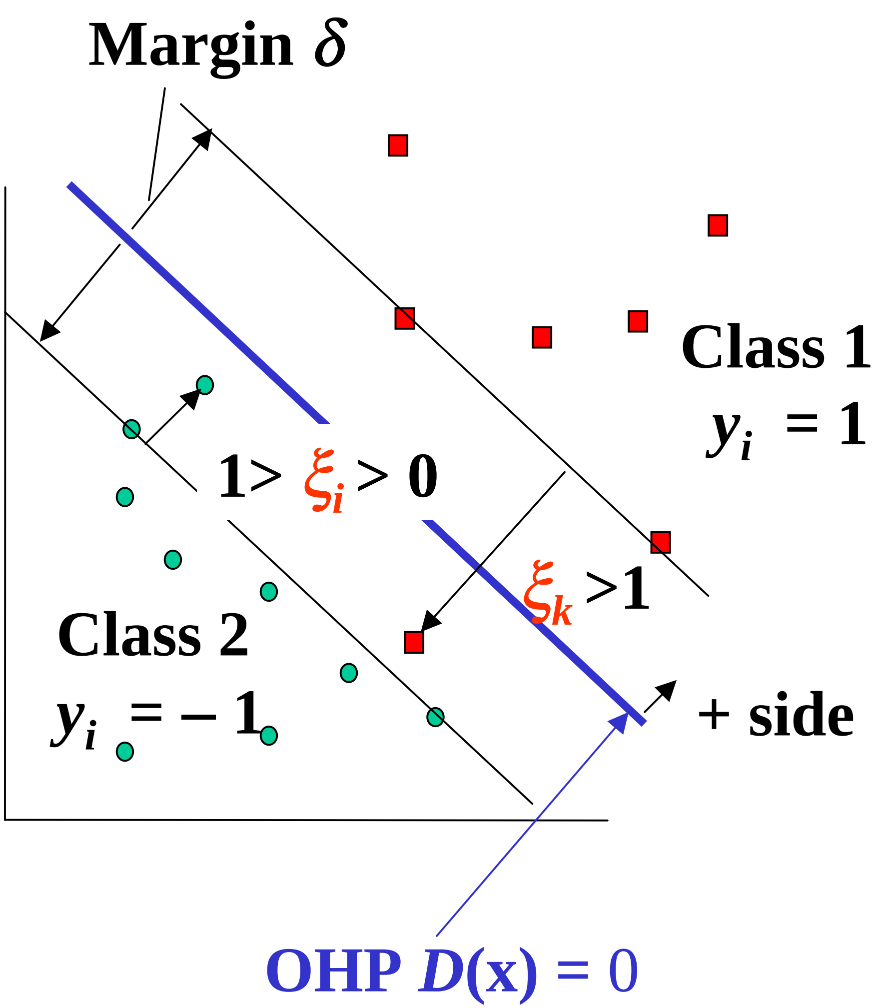
subject to

$$y_i \, D(\mathbf{x}_i) \geqq 1 - \xi_i$$

where C: margin parameter,

p = 1: L1 SVM,

= 2: L2 SVM



Conversion to Dual Problems

Introducing the Lagrange multiplies α_i and β_i,

$$Q = \|\mathrm{w}\|^2/2 + C/p \, \Sigma_i \, \xi_i^p - \Sigma_i \, \alpha_i(y_i(\mathrm{w}^t \mathrm{g}(\mathrm{x}_i) + b) - 1 + \xi_i)$$

$$[-\Sigma_i \, \beta_i \xi_i \,]$$

The Karush-Kuhn-Tacker (KKT) optimality conditions:

$$\partial \, Q / \partial \, \mathrm{w} = 0, \; \partial \, Q / \partial \, b = 0, \; \partial \, Q / \partial \, \xi_i = 0,$$

$$\alpha_i > 0, \; \beta_i > 0$$

KKT complementarity conditions

$$\alpha_i(y_i(\mathrm{w}^t \mathrm{g}(\mathrm{x}_i) + b) - 1 + \xi_i) = 0,$$

$$[(\beta_i \xi_i = 0 \,].$$

When $p = 2$, terms in [ ] are not necessary.

Dual Problems of SVMs

L1 SVM

Maximize

$$\Sigma_i \alpha_i - C/2\, \Sigma_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{g}(\mathbf{x}_i)^t \mathbf{g}(\mathbf{x}_j)$$

subject to

$$\Sigma_i y_i \alpha_i = 0, \quad C \geqq \alpha_i \geqq 0.$$

L2 SVM

Maximize

$$\Sigma_i \alpha_i - C/2\, \Sigma_{i,j} \alpha_i \alpha_j y_i y_j (\mathbf{g}(\mathbf{x}_i)^t \mathbf{g}(\mathbf{x}_j) + \delta_{ij} / C)$$

subject to

$$\Sigma_i y_i \alpha_i = 0, \quad \alpha_i \geqq 0,$$

where δ_{ij} **: 1 for** $i = j$ **and 0 for** $i \neq j$.

KKT Complementarity Condition

For L1 SVMs, from $\alpha_i(y_i(\mathbf{w}^t\mathbf{g}(\mathbf{x}_i) + b) - 1 + \xi_i) = 0$, $\beta_i \xi_i = (C - \alpha_i)\, \xi_i = 0$, there are three cases for α_i :

1. $\alpha_i = 0$. Then $\xi_i = 0$. Thus $\mathbf{x}_i$ is correctly classified,
2. $0 < \alpha_i < C$. Then $\xi_i = 0$, and $\mathbf{w}^t\mathbf{g}(\mathbf{x}_i) + b = y_i$,
3. $\alpha_i = C$. Then $\xi_i \geqq 0$.

Training data $\mathbf{x}_i$ with $\alpha_i > 0$ are called support vectors and those with $\alpha_i = C$ are called bounded support vectors.

The resulting decision function is given by

$$D(\mathbf{x}) = \Sigma_i\, \alpha_i y_i \mathbf{g}(\mathbf{x}_i)^t\, \mathbf{g}(\mathbf{x}) + b.$$

Kernel Trick

Since mapping function g(x) appears in the form of $g(x)^t g(x')$, we can avoid treating the variables in the feature space by introducing the kernel:

$$H(x,x') = g(x)^t g(x').$$

The following kernels are commonly used:

1. **Dot product kernels:** $H(x,x') = x^t x'$
2. **Polynomial kernels:** $H(x,x') = (x^t x'+1)^d$
3. **RBF kernels :** $H(x,x') = \exp(-\gamma \|x - x'\|^2)$

Summary

- The global optimum solution by quadratic programming (no local minima).
- Robust classification for outliers is possible by proper value selection of *C*.
- Adaptable to problems by proper selection of kernels.

Contents

Hessian Matrix

Substituting $\alpha_s = - y_s \Sigma y_i \alpha_i$ into the objective function,

$Q = \alpha^t 1 - 1/2\, \alpha^t H \alpha$

we derive the Hessian matrix.

L1 SVM

$$H_{L1} = (\cdots y_i (\mathbf{g}(\mathbf{x}_i) - \mathbf{g}(\mathbf{x}_s)) \cdots)^t (\cdots y_i (\mathbf{g}(\mathbf{x}_i) - \mathbf{g}(\mathbf{x}_s)) \cdots)$$

H_{L1}: **positive semidefinite**

L2 SVM

$$H_{L2} = H_{L1} + \{(y_i \; y_j + \delta_{ij})/C\}$$

H_{L2}: **positive definite, which results in stabler training.**

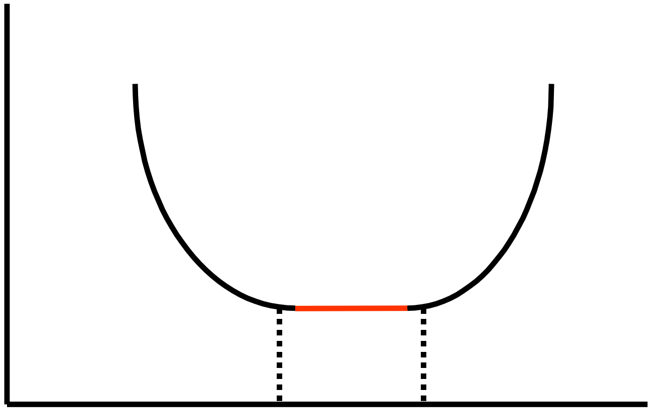
Non-unique Solutions

Strictly convex functions give unique solutions.

Table Uniqueness

	L1 SVM	L2 SVM
Primal	Non-unique*	Unique*
Dual	Non-unique	Unique*

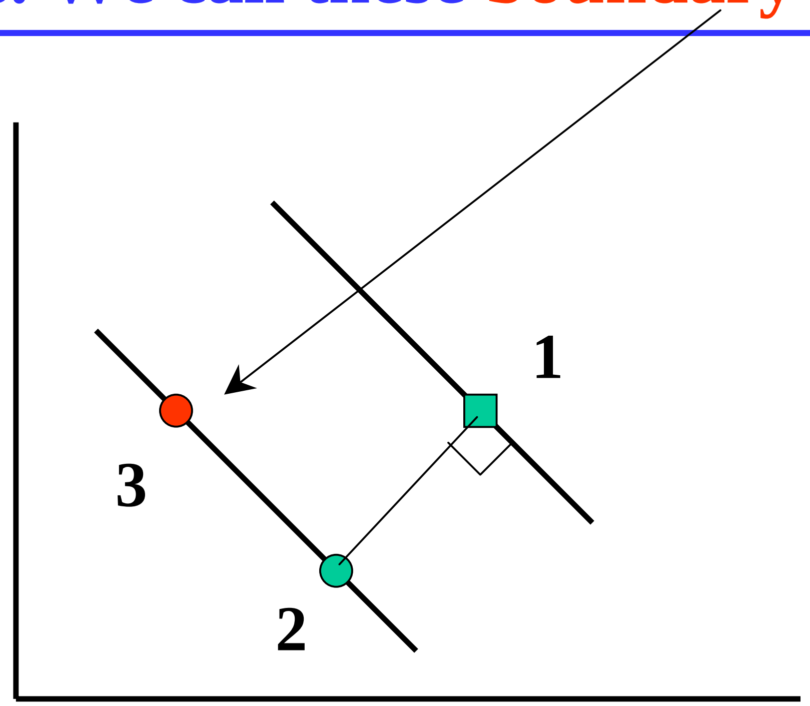
***: Burges and Crisp (2000)**



Convex objective function

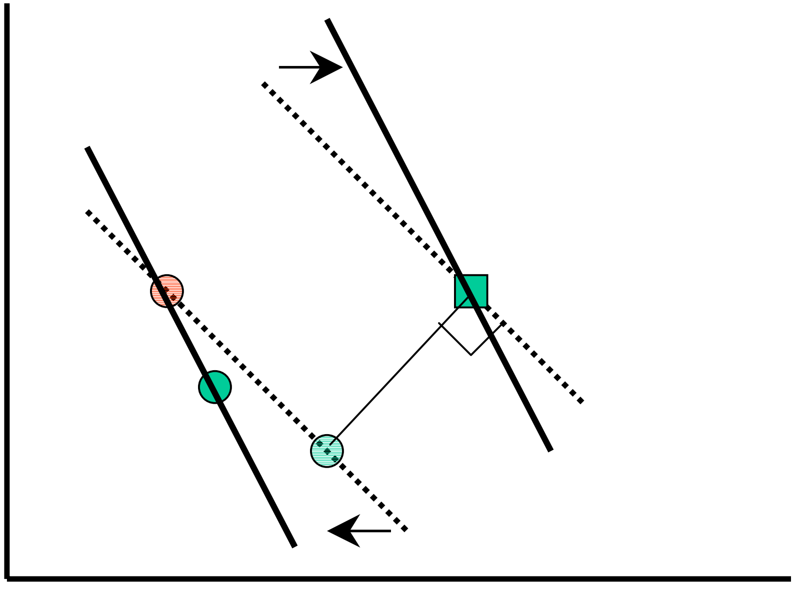
Property 1

For the L1 SVM, the vectors that satisfy $y_i(w^t g(x_i) + b) = 1$ are not always support vectors. We call these boundary vectors.



Irreducible Set of Support Vectors

A set of support vectors is irreducible if deletion of boundary vectors and any support vectors result in the change of the optimal hyperplane.

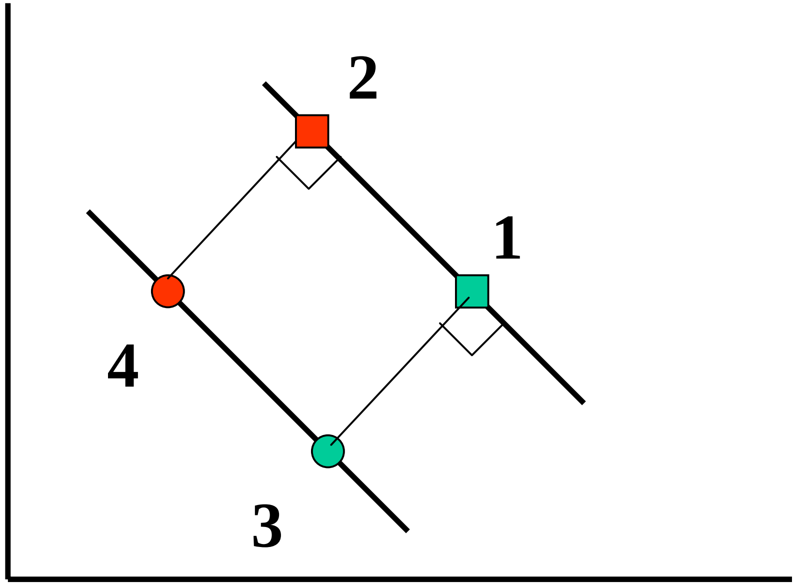


Property 2

For the L1 SVM, let all the support vectors be unbounded. Then the Hessian matrix associated with the irreducible set is positive definite.

Property 3

For the L1 SVM, if there is only one irreducible set, and support vectors are all unbounded, the solution is unique.



Irreducible sets

{1, 3}, {2, 4}

The dual problem is non-unique, but the primal problem is unique.

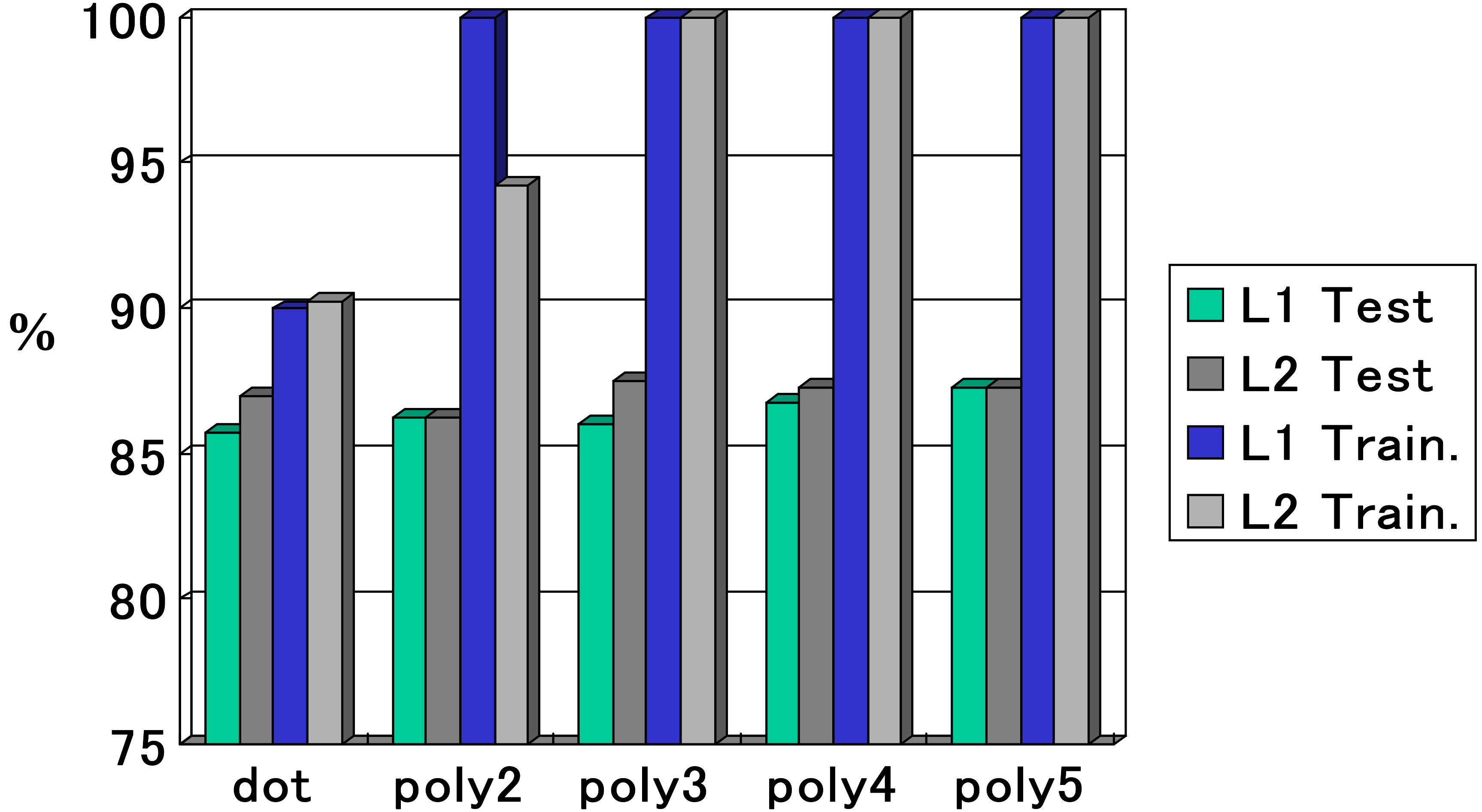
In general the number of support vectors of L2 SVM is larger than that of L1 SVM.

Computer Simulation

- **We used white blood cell data with 13 inputs and 2 classes, each class having app. 400 data for training and testing.**
- **We trained SVMs using the steepest ascent method (Abe et al. (2002))**
- **We used a personal computer (Athlon 1.6Ghz) for training.**

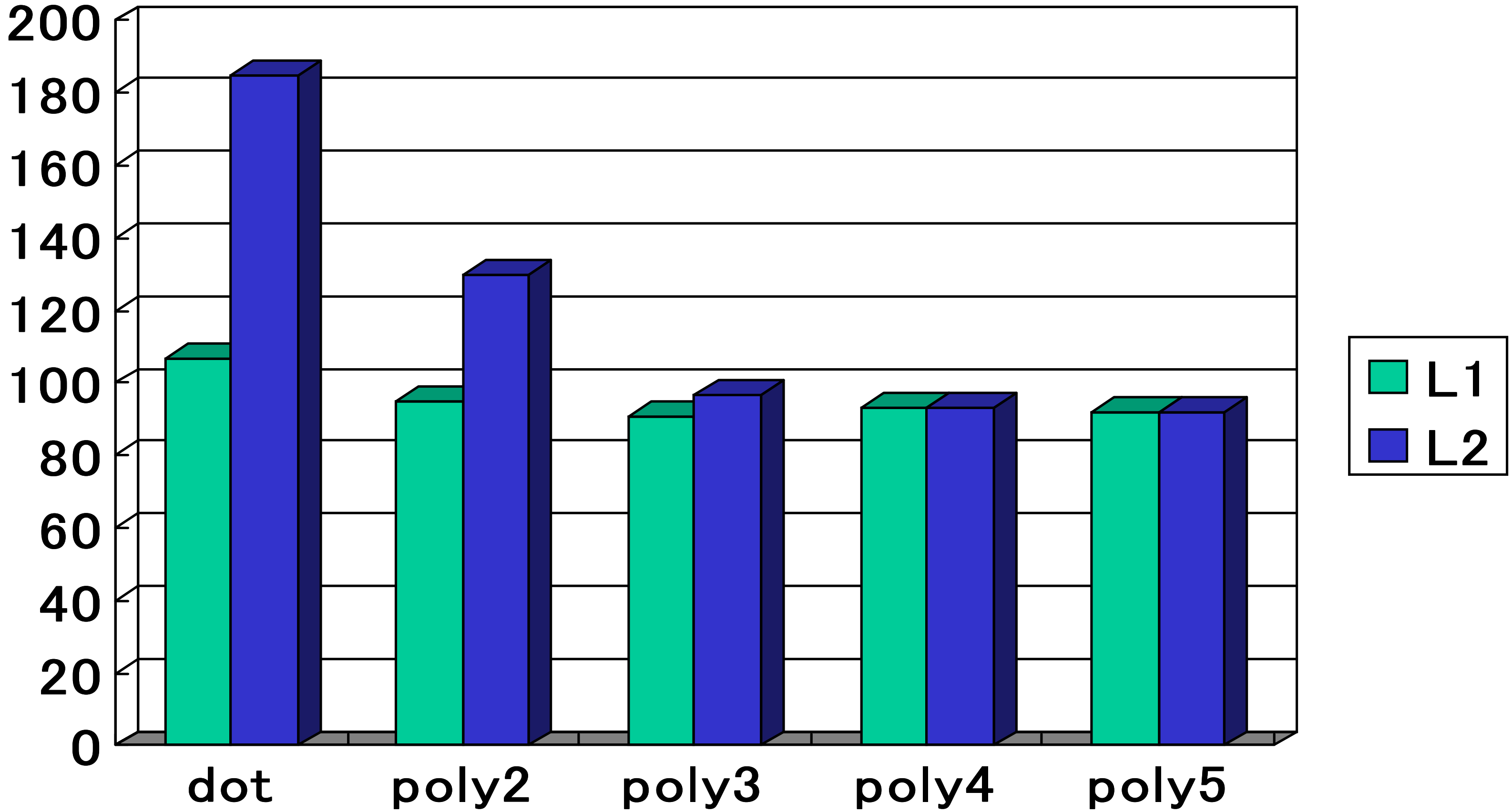
Recognition Rates for Polynomial Kernels

The difference is small.



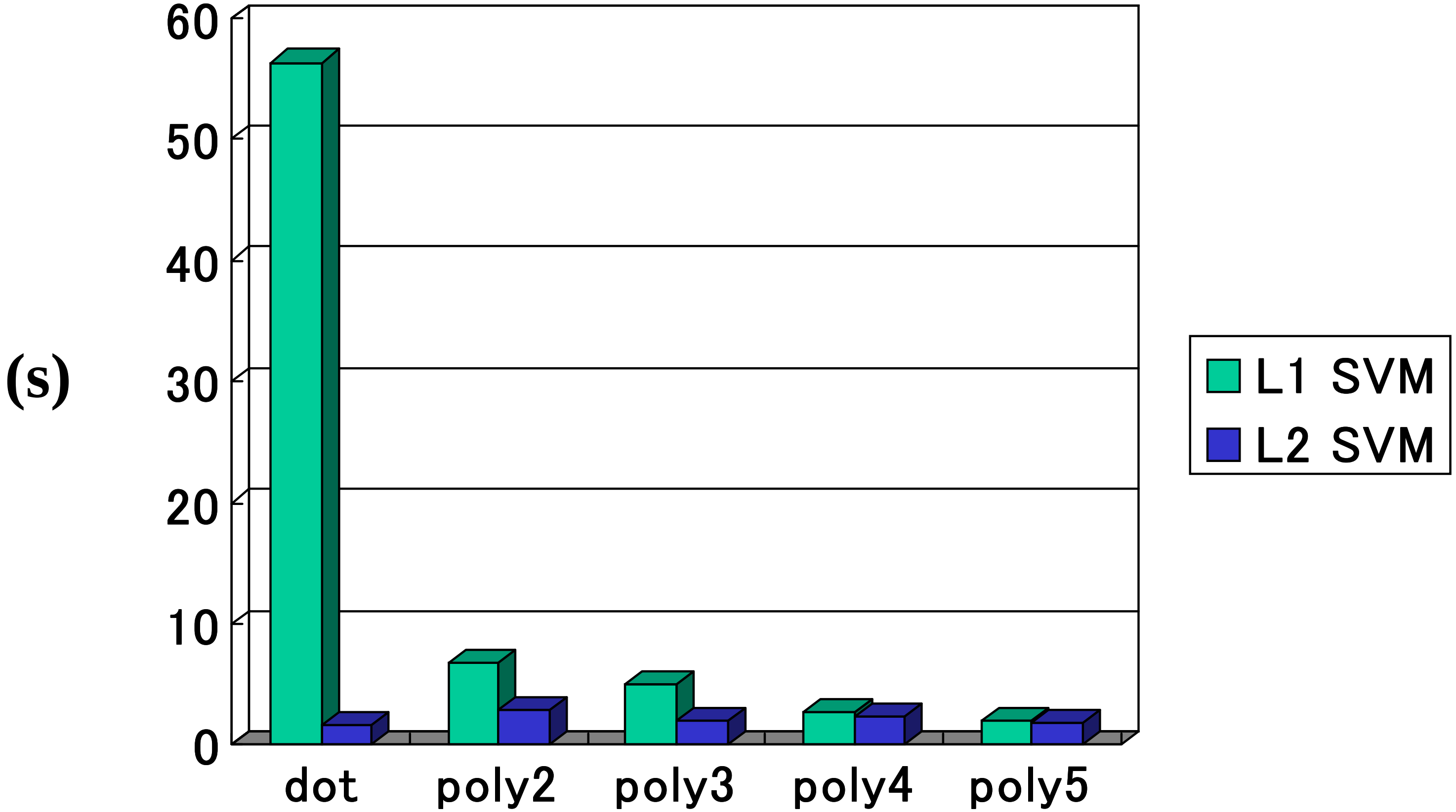
Support Vectors for Polynomial Kernels

For poly4 and poly5 the numbers are the same.



Training Time Comparison

As the polynomial degree increases, the difference becomes smaller.



Summary

- The Hessian matrix of an L1 SVM is positive semi-definite, but that of an L2 SVM is always positive definite.
- Thus dual solutions of an L1 SVM are non-unique.
- When non-critical the difference between L1 and L2 SVMs is small.

Contents

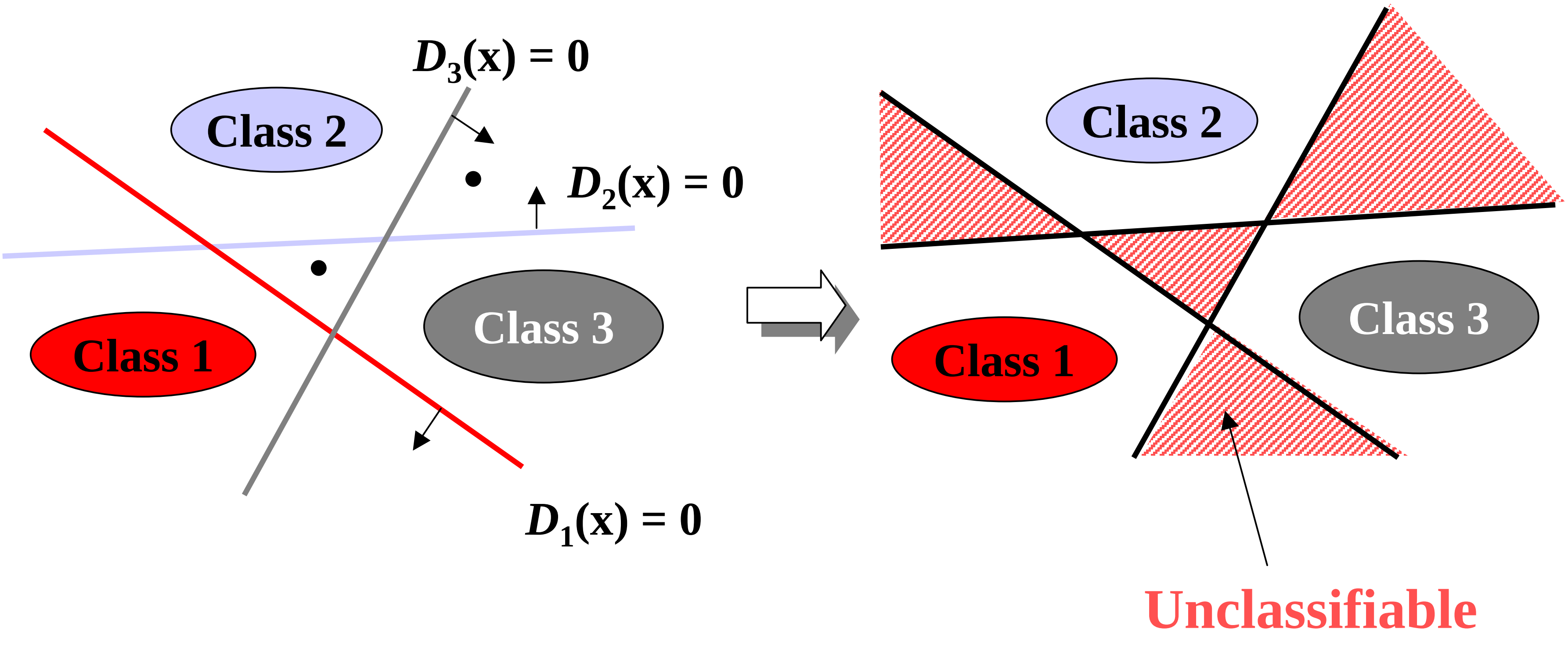
Multiclass SVMs

- One-against-all SVMs
  - Continuous decision functions
  - Fuzzy SVMs
  - Decision-tree-based SVMs
- Pairwise SVMs
  - Decision-tree-based SVMs (DDAGs, ADAGs)
  - Fuzzy SVMs
- ECOC SVMs
- All-at-once SVMs

One-against-all SVMs

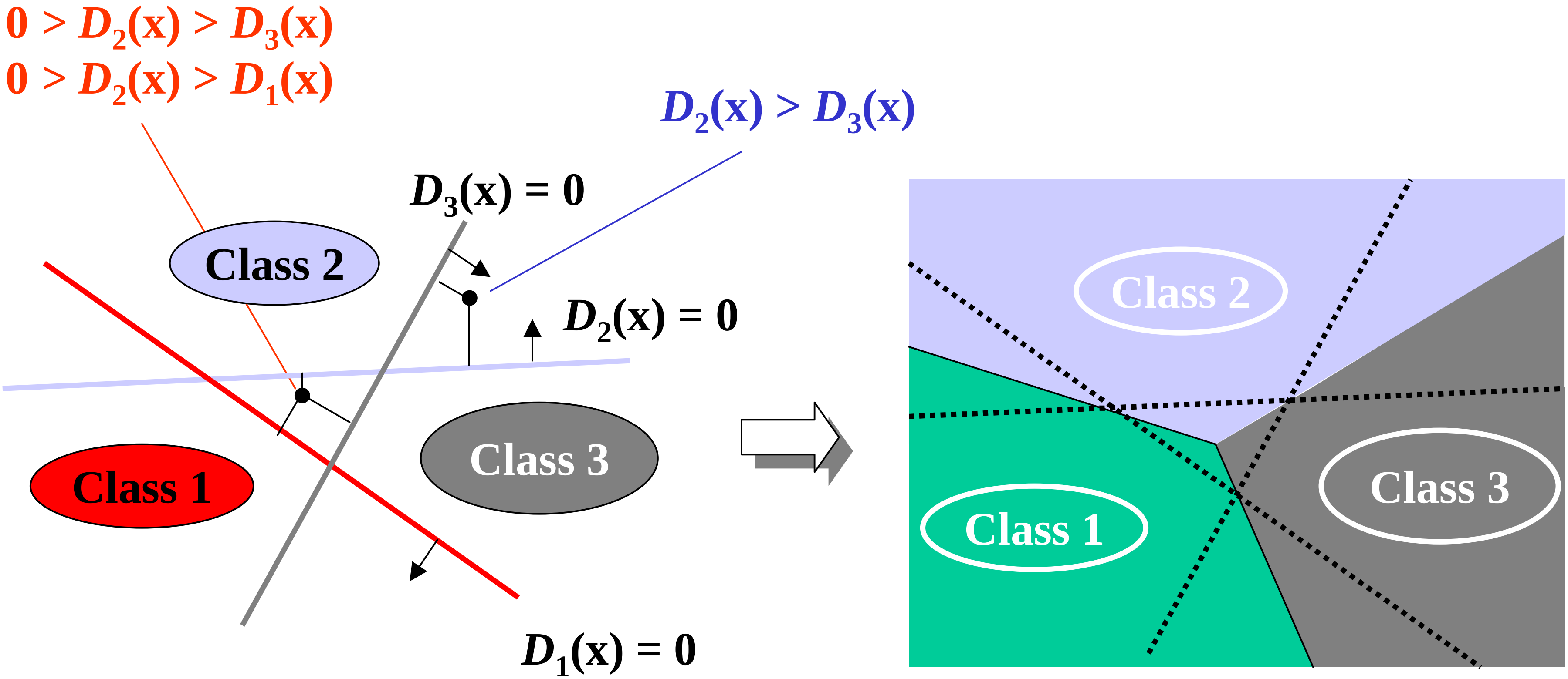
Determine the *i*th decision function $D_i(\mathbf{x})$ so that class *i* is separated from the remaining classes.

Classify x into the class with $D_i(\mathbf{x}) > 0$.

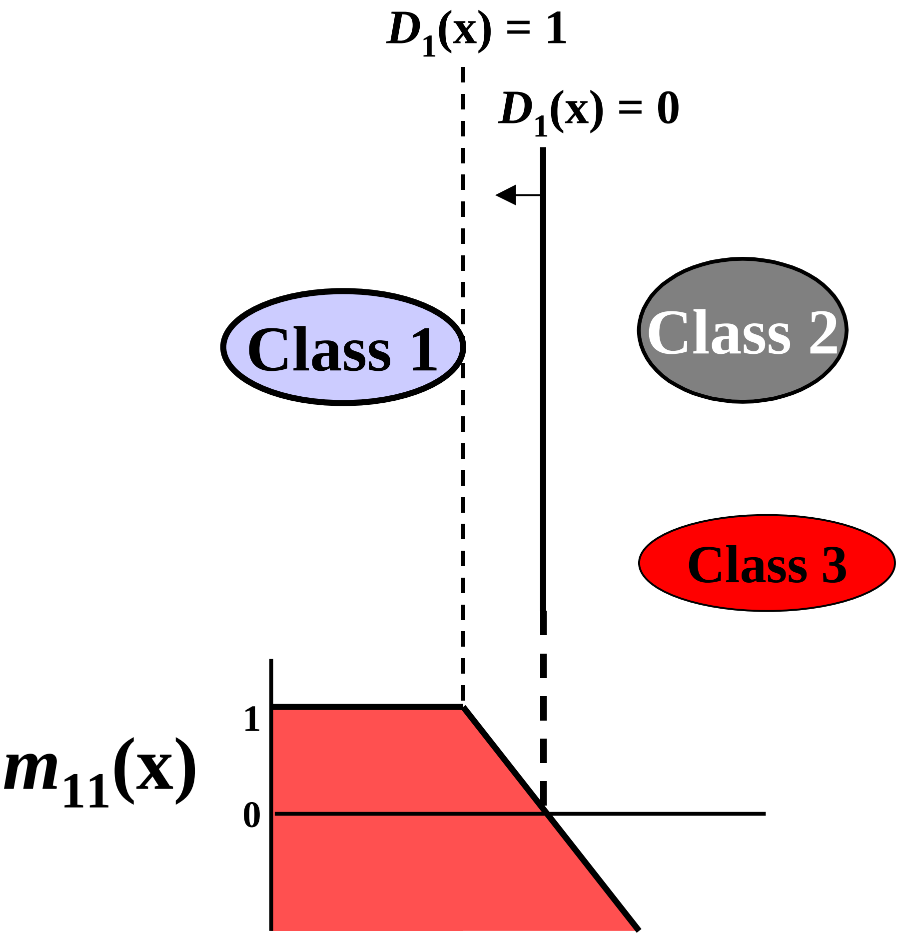


Continuous SVMs

Classify x into the class with max $D_i(\mathbf{x})$.



Fuzzy SVMs



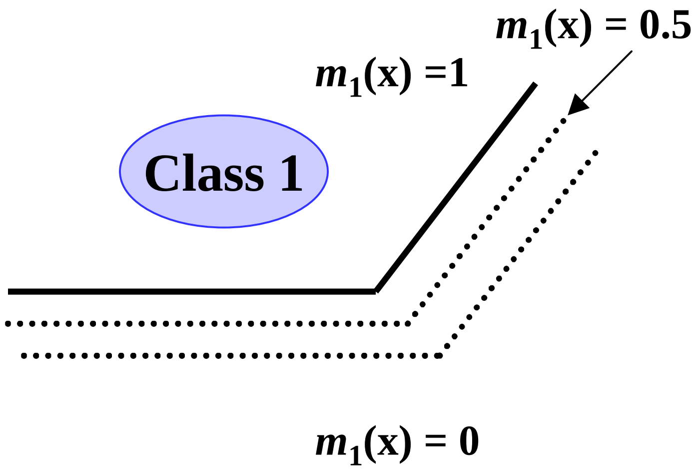
We define a one-dimensional membership function in the direction orthogonal to the decision function D_{ij}(x).

Membership function

Class i Membership

Class i membership function

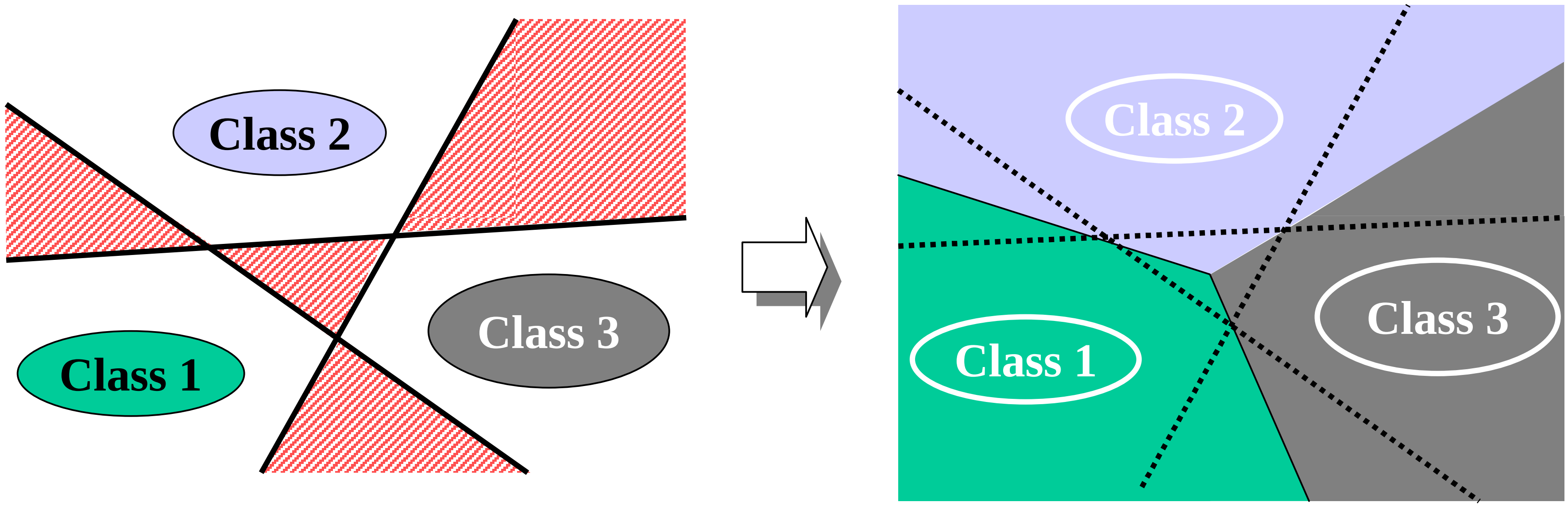
$$m_i(\mathbf{x}) = \min_{j=1,\ldots,n} m_{ij}(\mathbf{x}).$$



The region that satisfies $m_i(\mathbf{x}) > 0$ corresponds to the classifiable region for class i.

Resulting Class Boundaries by Fuzzy SVMs

The generalization regions are the same with those by continuous SVMs.



Decision-tree-based SVMs

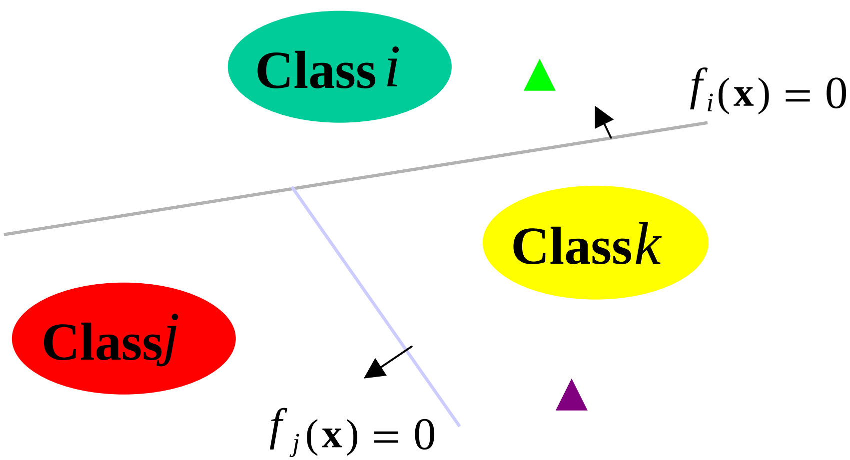
Each node corresponds to the hyperplane.

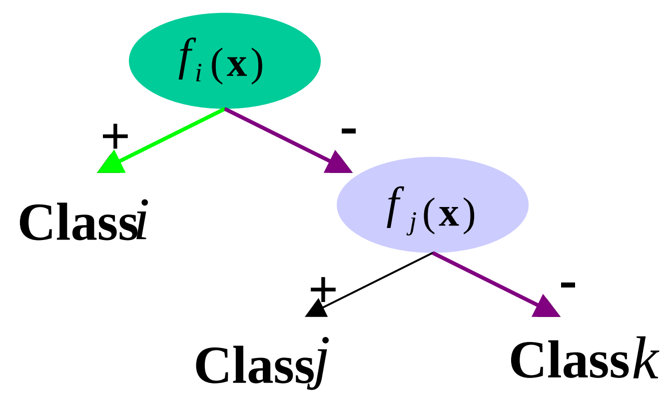
Training step

At each node, determine OHP.

Classification step

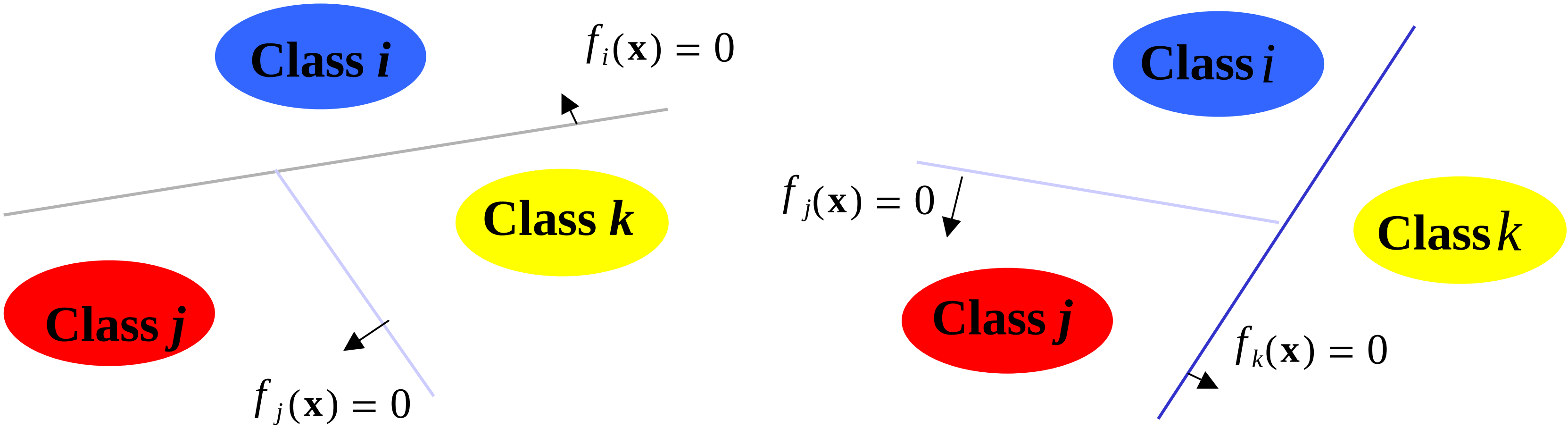
Starting from the top node, calculate the value until a leaf node is reached.





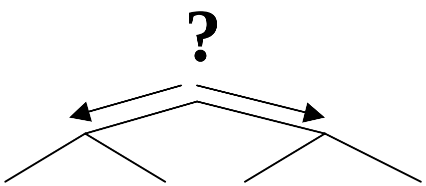
The problem of the decision tree

- **Unclassifiable region can be resolved.**
- **The region of each class depends on the structure of a decision tree.**



How do we determine the decision tree?

Determination of the decision tree



⇨ **The overall classification performance becomes worse.**

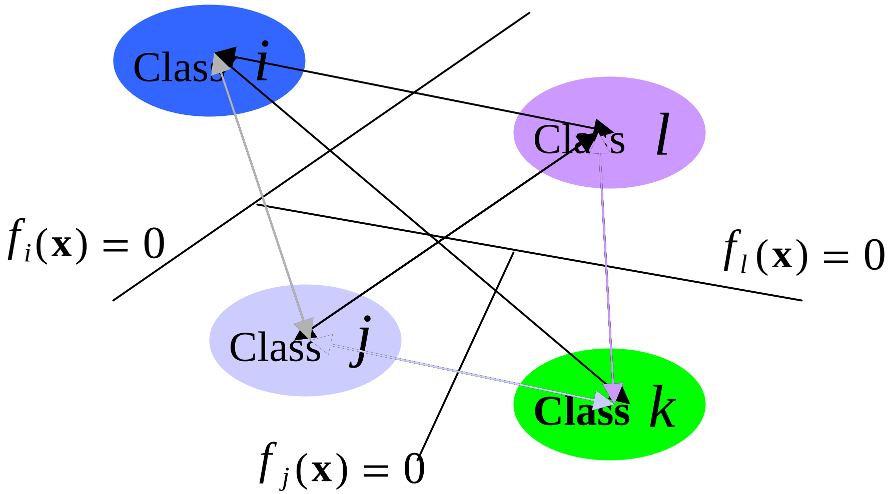
The separable classes need to be separated at the upper node.

The separability measures:

- **Euclidean distances between class centers**
- **Classification errors by the Mahalanobis distance**
  - **1 vs. remaining classes**
  - **Some vs. remaining classes**

1 Class vs. Remaining Classes Using Distances between Class Centers

Separate the farthest class from remaining classes.

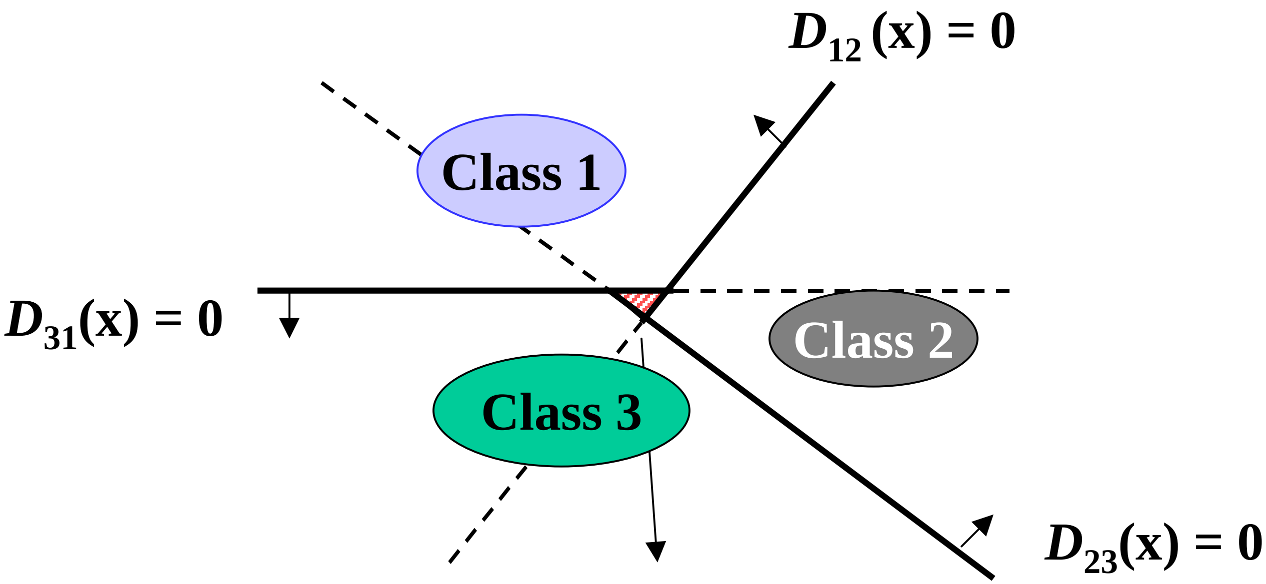


- **Calculate distances between class centers.**
- **For each class, find the smallest value.**
- **Separate the class with the largest value in step 2.**
- **Repeat for remaining classes.**

Pairwise SVMs

For all pairs of classes *i*, *j*, we define *n*(*n*-1)/2 decision functions and classify x into class

$$\arg\max_i D_i(\mathbf{x}) \text{ where } D_i(\mathbf{x}) = \sum_j \text{sign } D_{ij}(\mathbf{x})$$

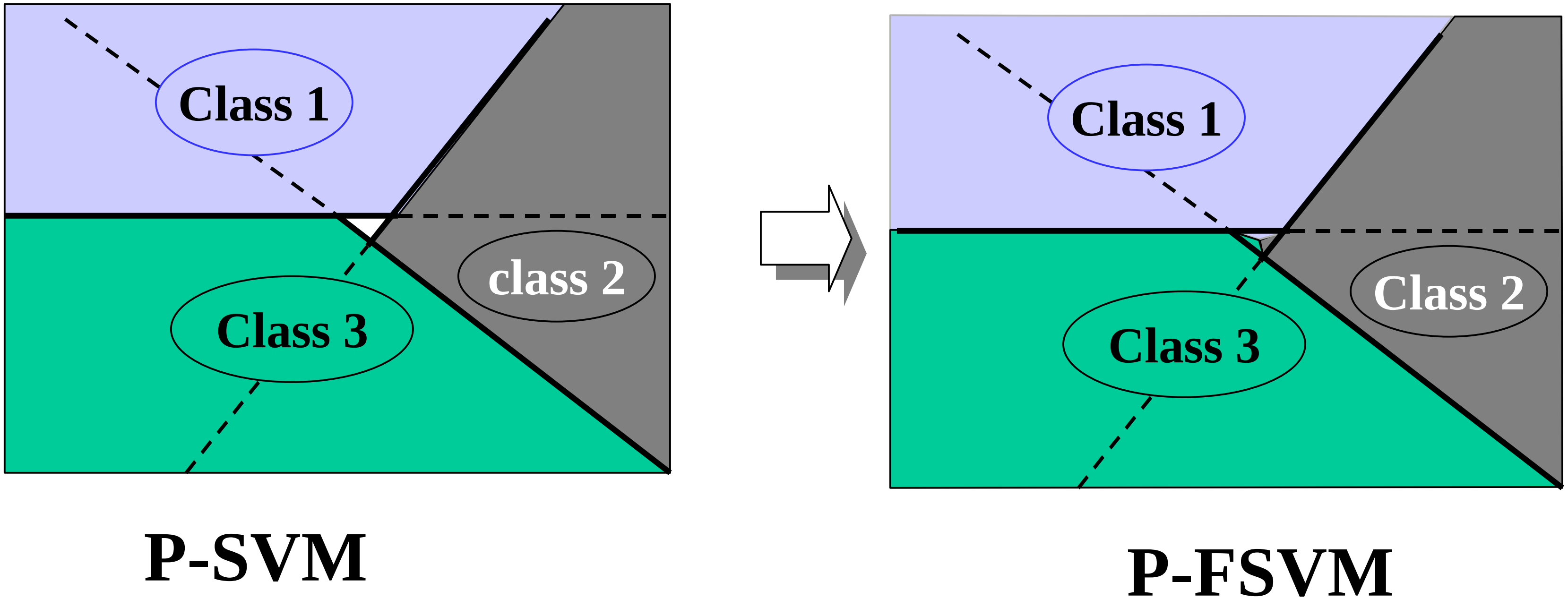


Unclassifiable regions still exist.

$D_1(\mathbf{x}) = D_2(\mathbf{x}) = D_3(\mathbf{x}) = 1$

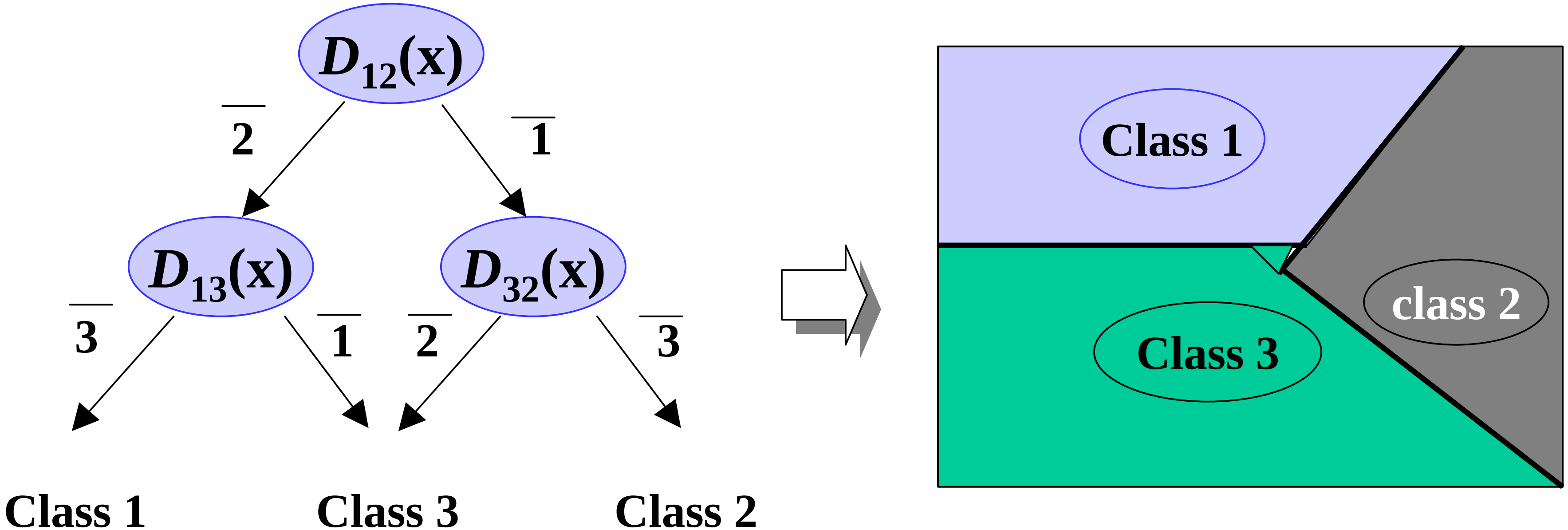
Pairwise Fuzzy SVMs

The generalization ability of the P-FSVM is better than the P-SVM.



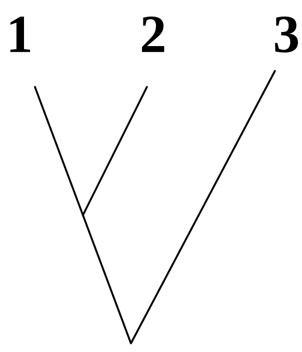
Decision-tree-based SVMs (DDAGs)

Generalization regions change according to the tree structure.

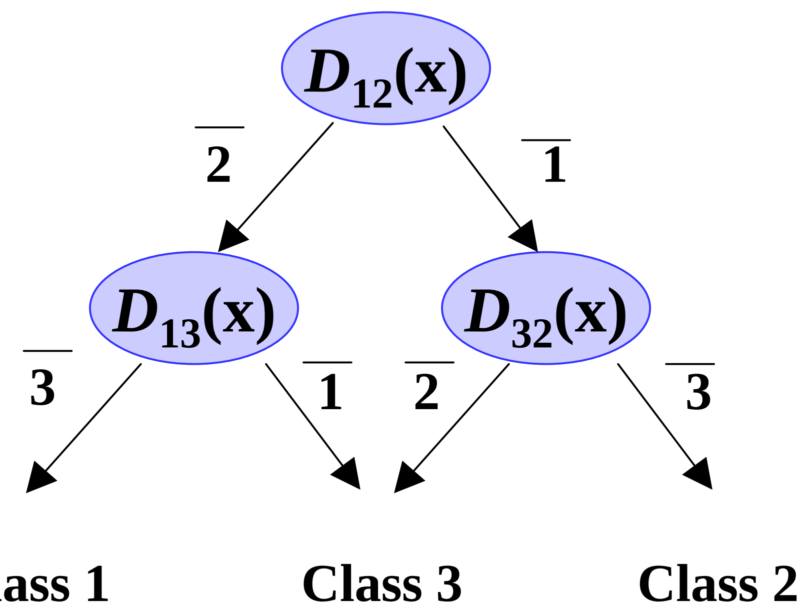


Decision-tree-based SVMs (ADAGs)

For three-class problems ADAGs and DDAGs are equivalent. In general, DDAGs include ADAGs.



ADAG (Tennis Tournament)



Equivalent DDAG

ECC Capability

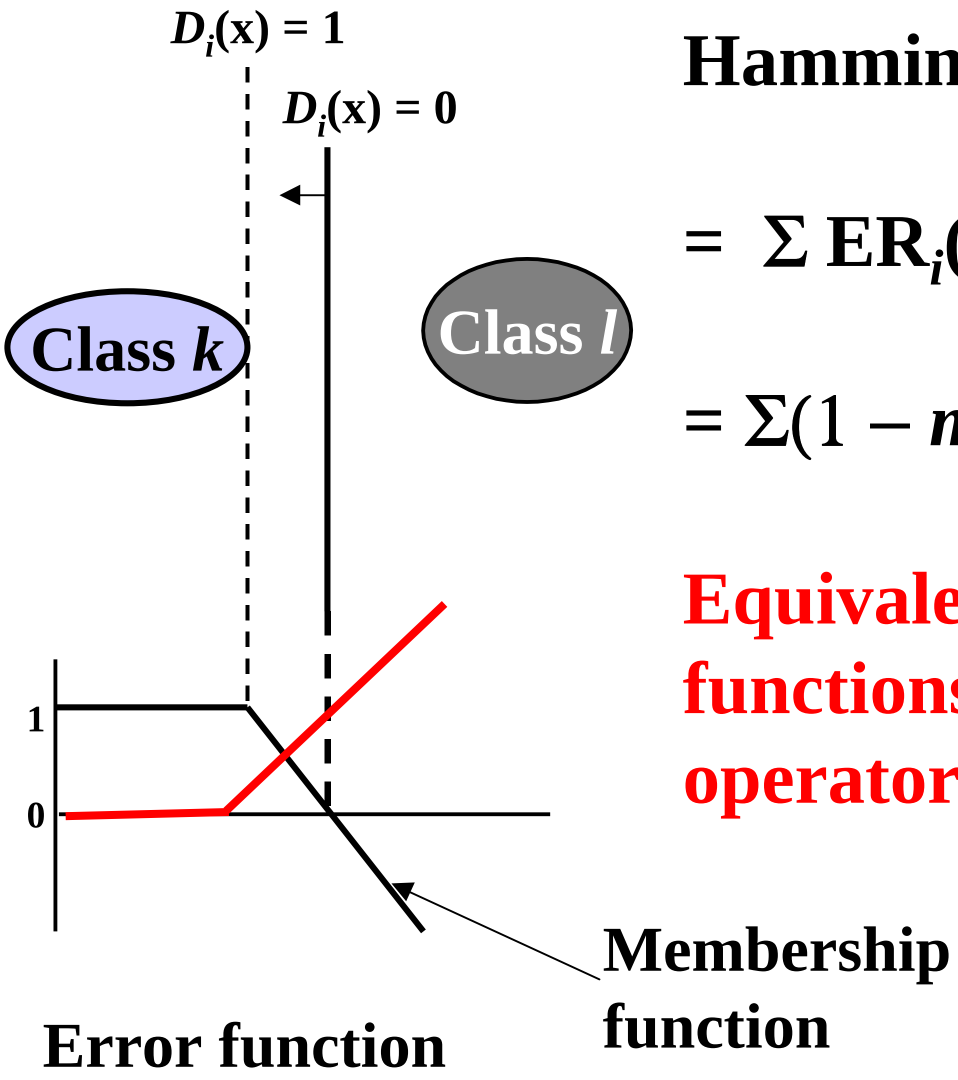
The maximum number of decision functions = $2^{n-1} - 1$, where *n* is the number of classes.

Error correcting capability = (*h*-1)/2, where *h* is the Hamming distance.

For *n* = 4, *h* = 4 and ecc of 1.

Class 1	1	1	1	-1	1	1	-1
Class 2	1	1	-1	1	1	-1	1
Class 3	1	-1	1	1	-1	1	1
Class 4	-1	1	1	1	-1	-1	-1

Continuous Hamming Distance



Hamming Distance

$$= \Sigma\, ER_i(\mathbf{x})$$

$$= \Sigma(1 - m_i(\mathbf{x}))$$

Equivalent to membership functions with sum operators

All-at-once SVMs

Decision functions

$$\mathbf{w}_i\, \mathbf{g}(\mathbf{x}) + b_i > \mathbf{w}_j \mathbf{g}(\mathbf{x}) + b_j$$

$$\text{for } j \neq i,\ j=1,\ldots,n$$

Original formulation

$n \times M$ **variables**

where n**: number of classes**

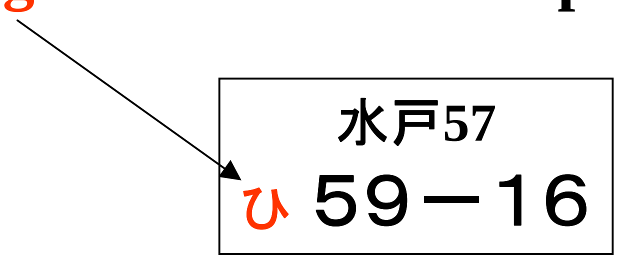
M**: number of training data**

Crammer and Singer (2000)'s

M **variables**

Performance Evaluation

- Compare recognition rates of test data for one-against-all and pairwise SVMs.
- Data sets used:
  - white blood cell data
  - thyroid data
  - hiragana data with 50 inputs
  - hiragana data with 13 inputs

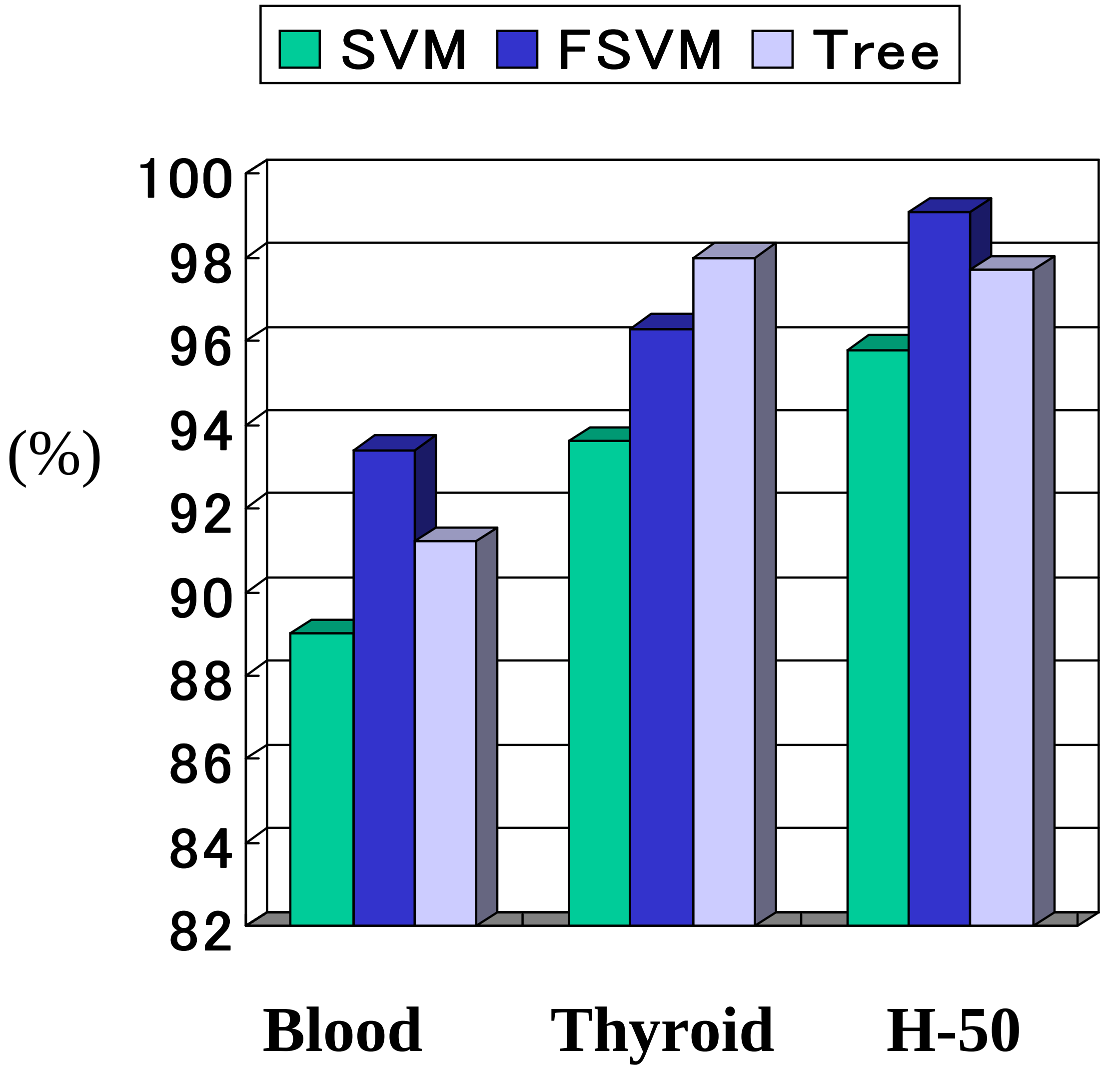


Japanese License Plate

Data Sets Used for Evaluation

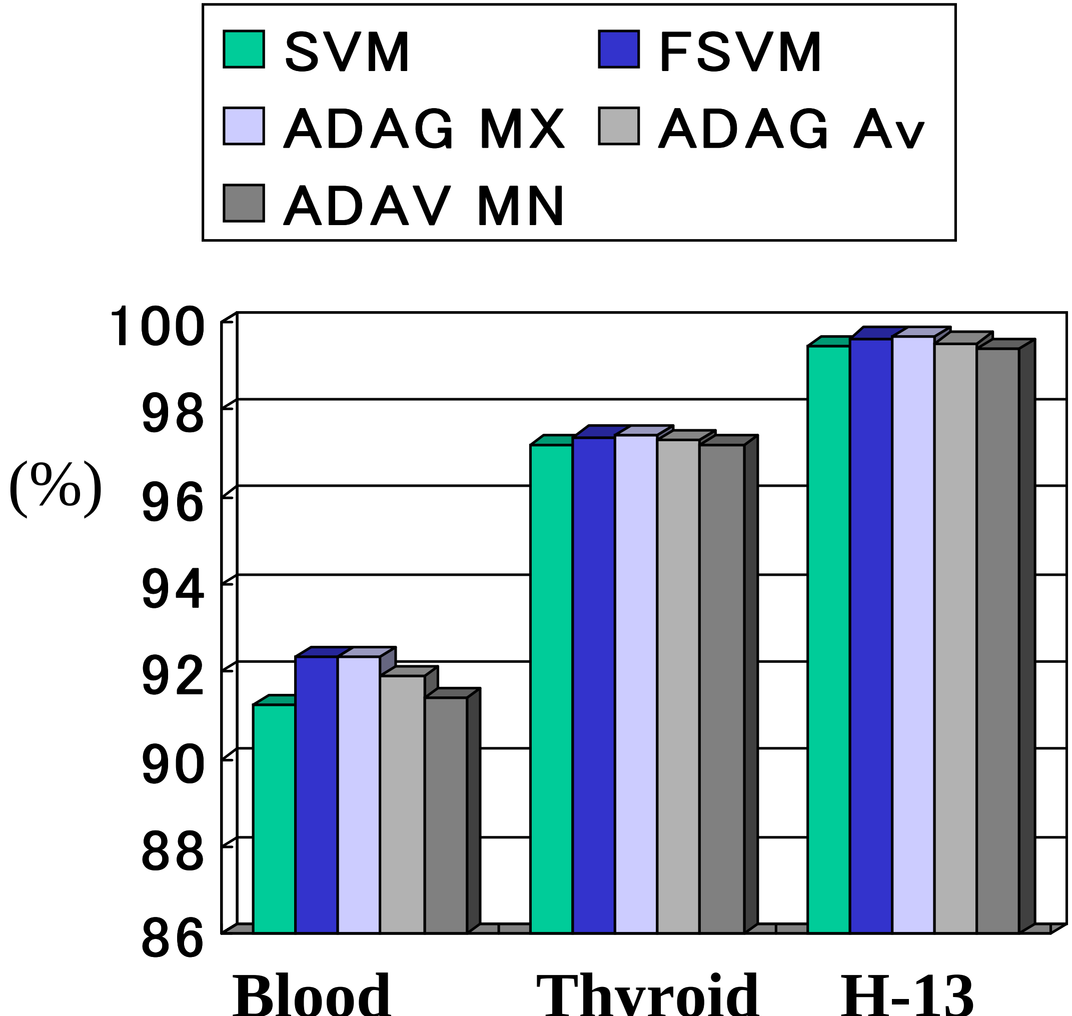
Data	Inputs	Classes	Train.	Test
Blood Cell	13	12	3097	3100
Thyroid	21	3	3772	3428
H-50	50	39	4610	4610
H-13	13	38	8375	8356

Performance Improvement for 1-all SVMs



Performance improves by fuzzy SVMs and decision trees.

Performance Improvement for Pairwise Classification



FSVMs are comparable with ADAG MX.

Summary

- One-against-all SVMs with continuous decision functions are equivalent to 1-all fuzzy SVMs.
- Performance of pairwise fuzzy SVMs is comparable to that of ADAGs with maximum recognition rates.
- There is no so much difference between one-against-all and pairwise SVMs.

Contents

Research Status

- **Too slow to train by quadratic programming for a large number of training data.**
- **Several training methods have been developed:**
  - **decomposition technique by Osuna (1997)**
  - **Kernel-Adatron by Friess et al. (1998)**
  - **SMO (Sequential Minimum Optimization) by Platt (1999)**
  - **Steepest Ascent Training by Abe et al. (2002)**

Decomposition Technique

Decompose the index set into two: *B* and *N*.

Maximize

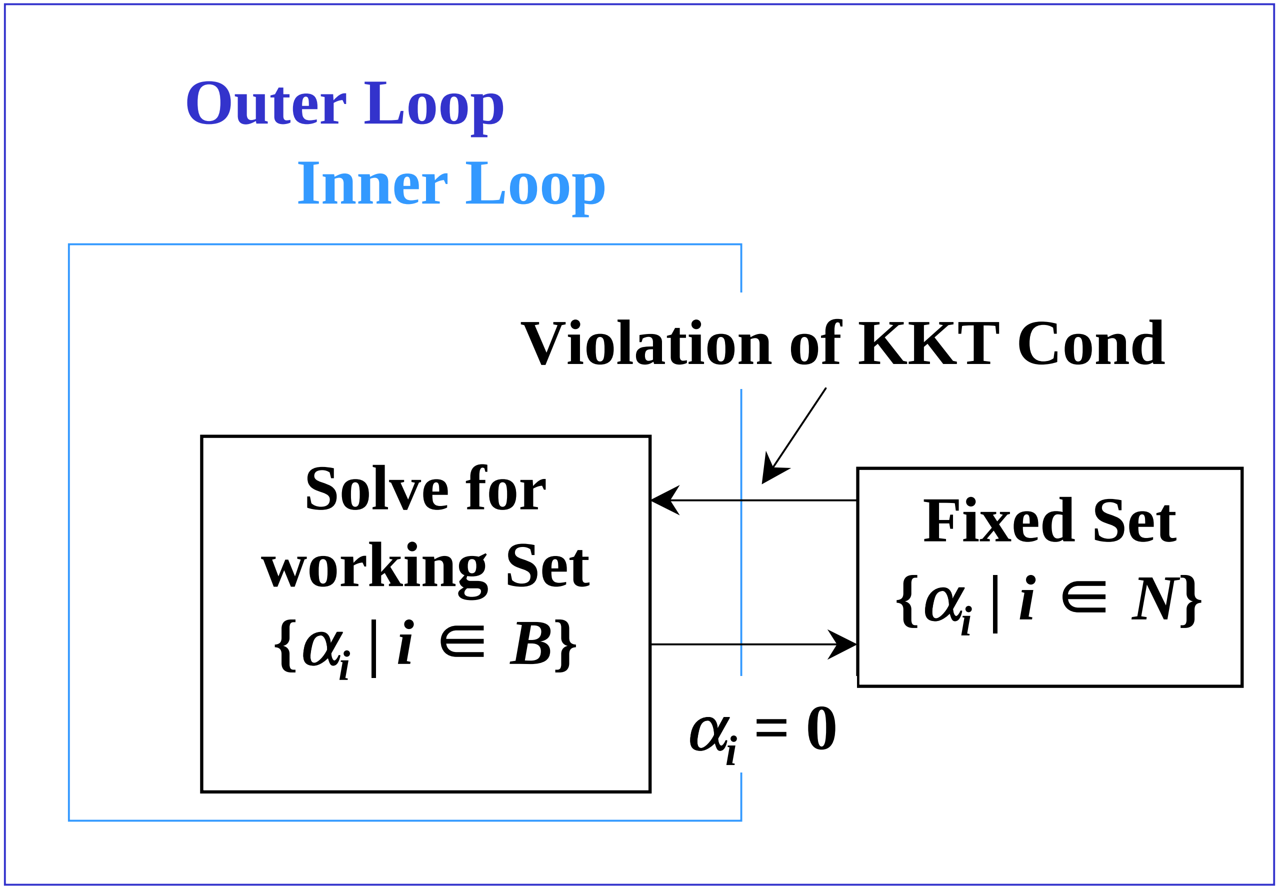
$$Q(\alpha) = \Sigma_{i\in B}\, \alpha_i - C/2\, \Sigma_{i,j\in B}\, \alpha_i\, \alpha_j\, y_i\, y_j\, H(\mathbf{x}_i, \mathbf{x}_j)$$
$$- C\, \Sigma_{i\in B,\, j\in N}\, \alpha_i\, \alpha_j\, y_i\, y_j\, H(\mathbf{x}_i, \mathbf{x}_j)$$
$$- C/2\, \Sigma_{i,j\in N}\, \alpha_i\, \alpha_j\, y_i\, y_j\, H(\mathbf{x}_i, \mathbf{x}_j) + \Sigma_{i\in N}\, \alpha_i$$

subject to

$$\Sigma_{i\in B}\, y_i\, \alpha_i = -\, \Sigma_{i\in N}\, y_i\, \alpha_i\, ,\ C \geqq \alpha_i \geqq 0 \text{ for } i \in B$$

fixing α_i for $i \in N$.

Solution Framework



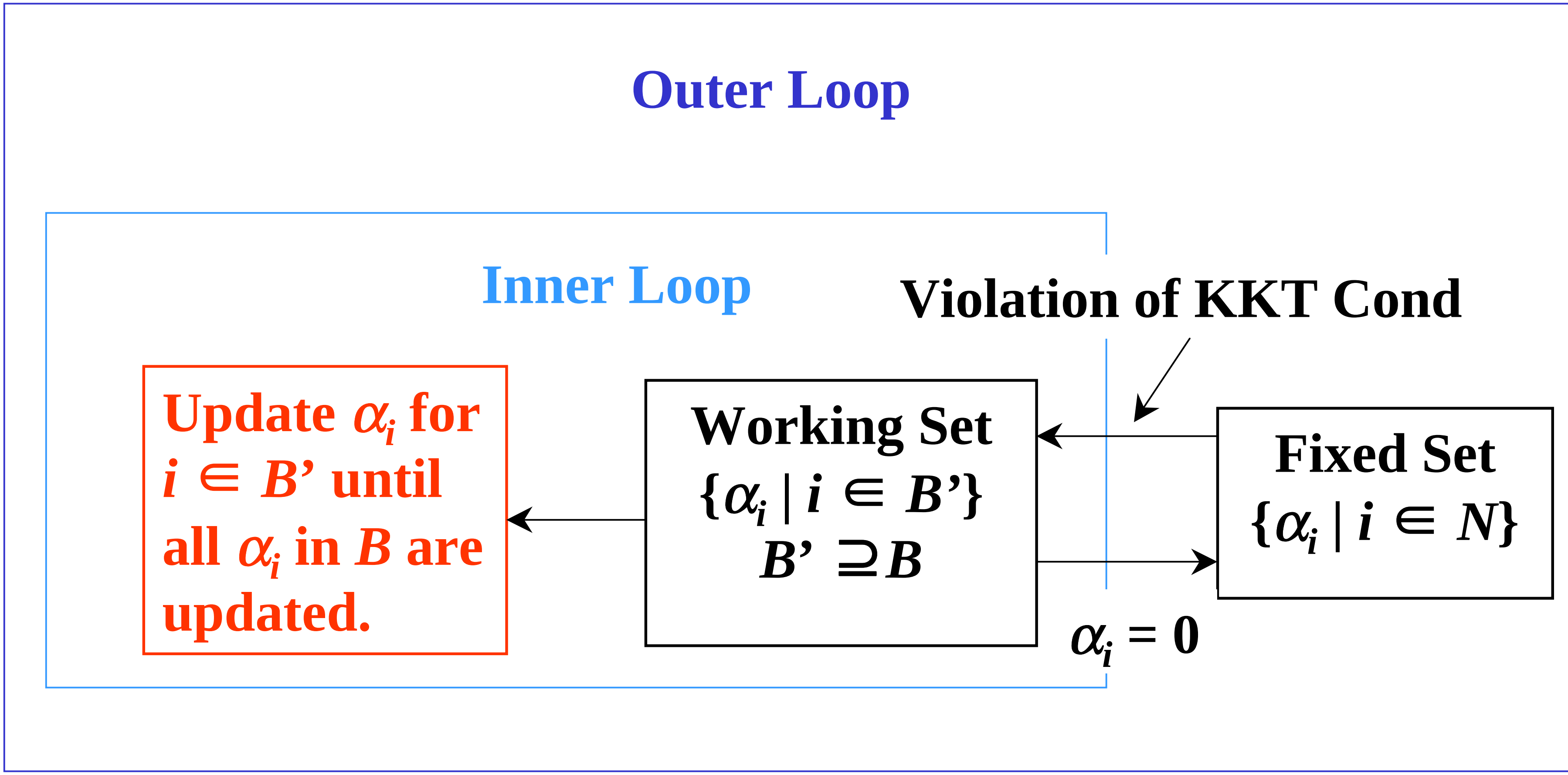
Use of QP Package e.g., LOQO (1998)

Solution without Using QP Package

- **SMO**
  - **|B| = 2, i.e., solve the problem for two variables**
  - **The subproblem is solvable without matrix calculations**

- **Steepest Ascent Training**
  - **Speedup SMO by solving subproblems with variables more than two**

Solution Framework for Steepest Ascent Training

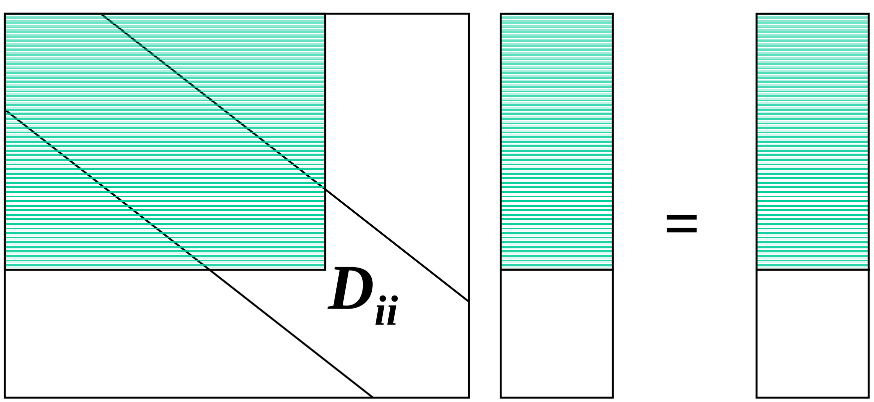


Solution Procedure

- **Set the index set *B*.**
- **Select α_s and eliminate the equality constraint by substituting $\alpha_s = -\, y_s \Sigma y_j\, \alpha_j$ into the objective function.**
- **Calculate corrections by**
  $\alpha_{B''} = -(\partial^2 Q / \partial \alpha^2_{B''})^{-1}\ \partial Q / \partial \alpha_{B''}$
  where $B'' = B' - \{s\}$.
- **Correct variables if possible.**

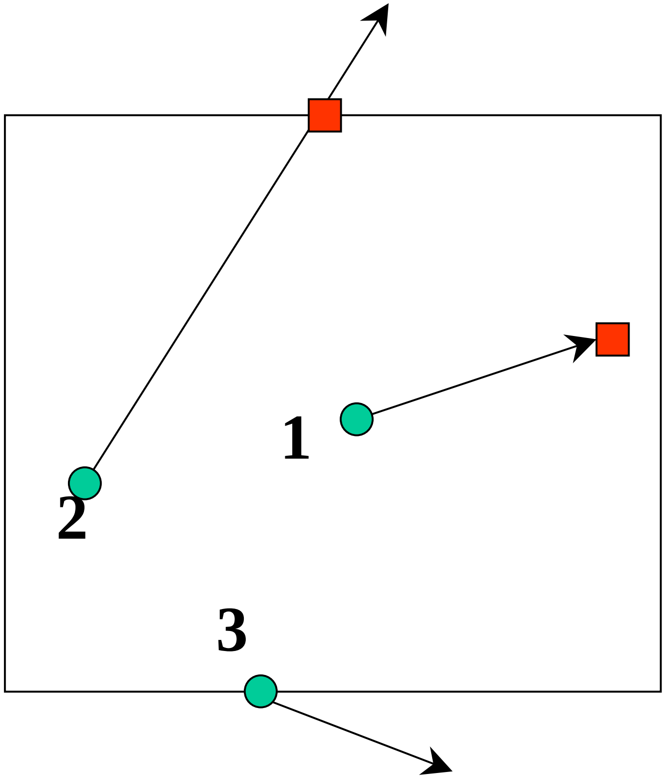
Calculation of corrections

- We calculate corrections by the Cholesky decomposition.
- For the L2 SVM, since the Hessian matrix is positive definite, it is regular.
- For the L1 SVM, since the Hessian matrix is positive semi-definite, it may be singular.



If $D_{ii} < \eta$, we discard the variables α_j, $j \geqq i$.

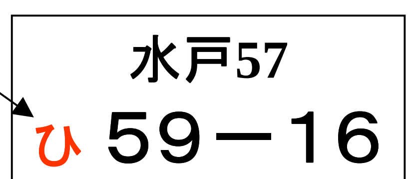
Update of Variables



- **Case 1**
  Variables are updated.
- **Case 2**
  Corrections are reduced to satisfy constraints.
- **Case 3**
  Variables are not corrected.

Performance Evaluation

- **Compare training time by the steepest ascent method, SMO, and the interior point method.**
- **Data sets used:**
  - **white blood cell data**
  - **thyroid data**
  - **hiragana data with 50 inputs**
  - **hiragana data with 13 inputs**

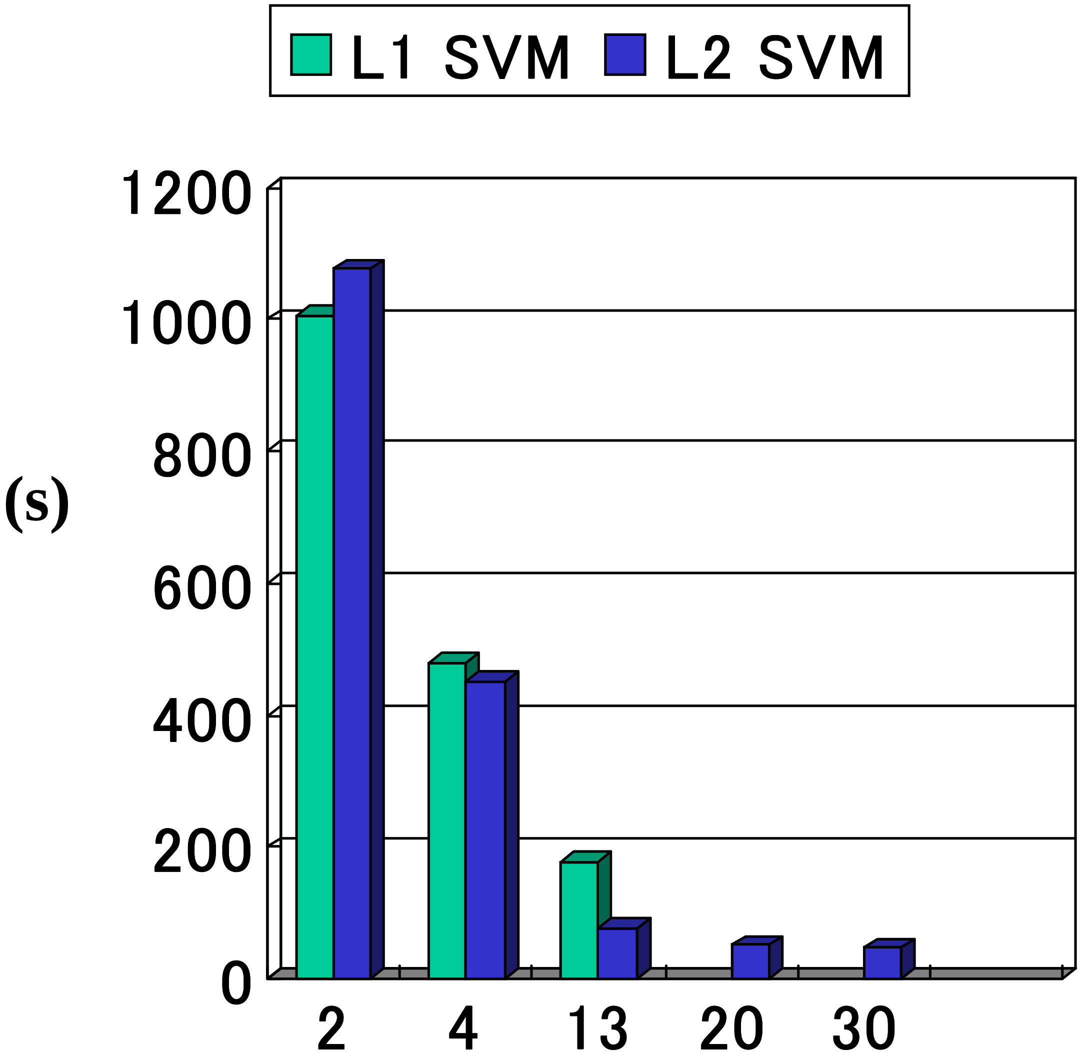


Japanese License Plate

Data Sets Used for Evaluation

Data	Inputs	Classes	Train.	Test
Blood Cell	13	12	3097	3100
Thyroid	21	3	3772	3428
H-50	50	39	4610	4610
H-13	13	38	8375	8356

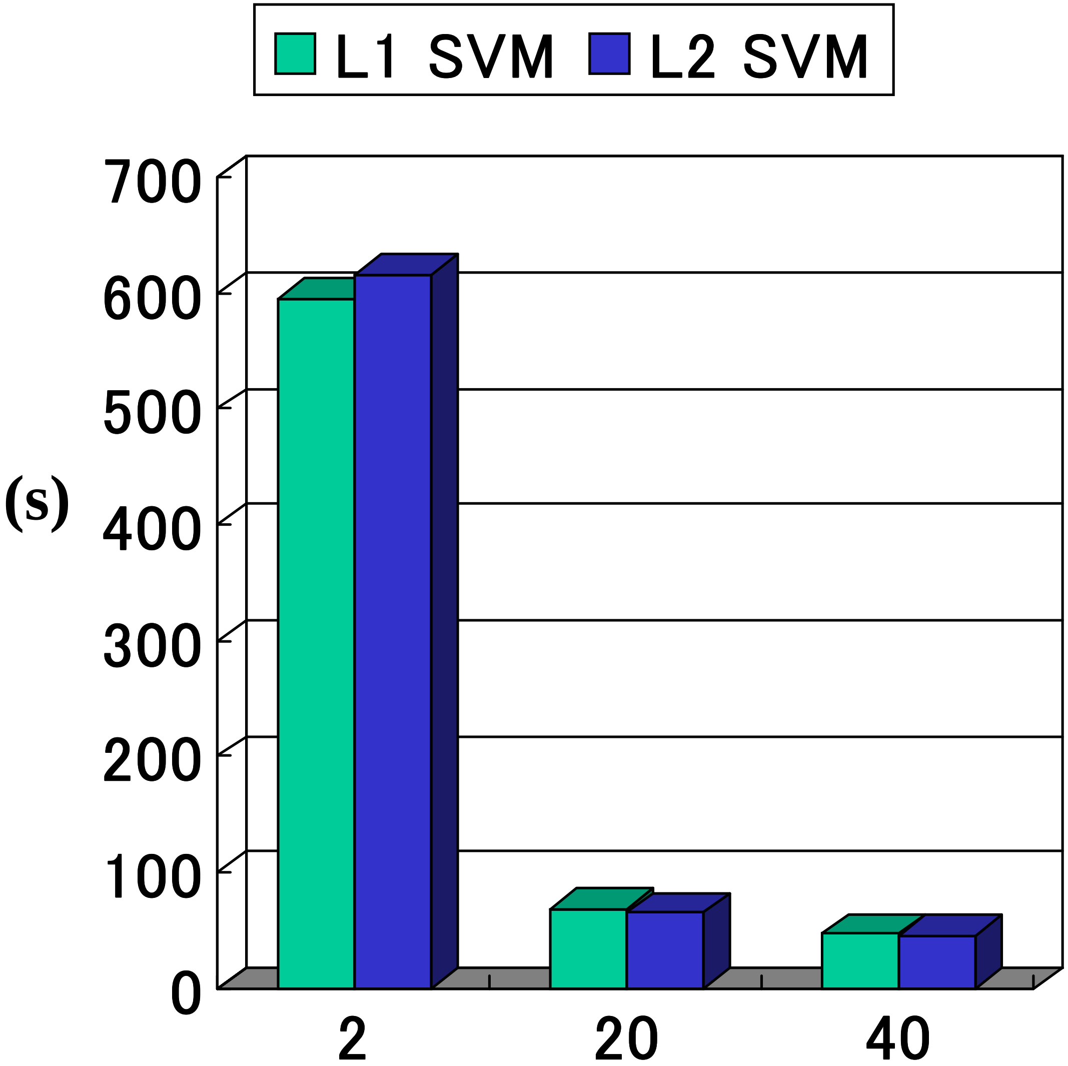
Effect of Working Set Size for Blood Cell Data



Dot product kernels are used.

For a larger size, L2 SVMs are trained faster.

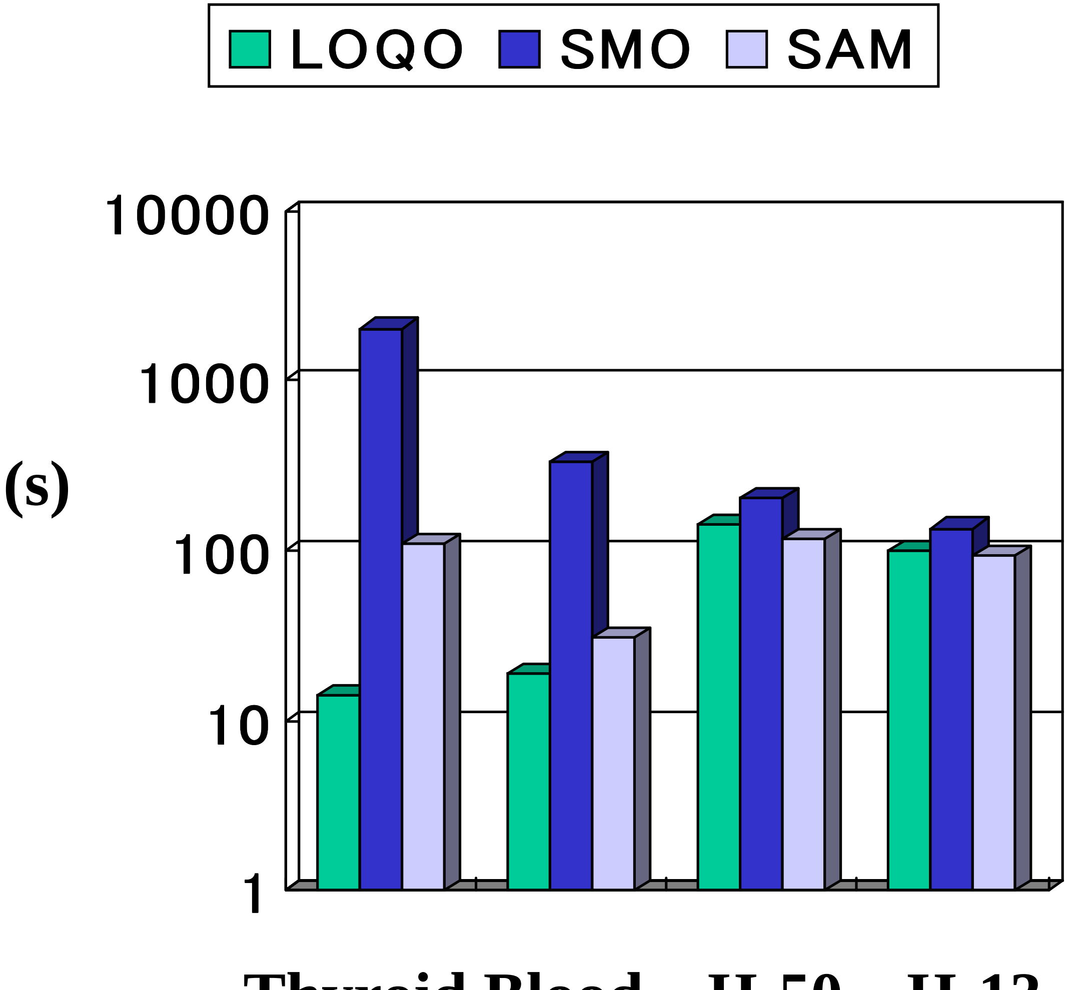
Effect of Working Set Size for Blood Cell Data (Cond.)



Polynomial kernels with degree 4 are used.

No much difference for L1 and L2 SVMs.

Training Time Comparison



LOQO: LOQO is combined with decomposition technique.

SMO is the slowest. LOQO and SAM are comparable for hiragana data.

Summary

- The steepest ascent method is faster than SMO and comparable for some cases with the interior-point method combined with decomposition technique.
- For the critical cases, L2 SVMs are trained faster than L1 SVMs, but for normal cases they are almost the same.

Contents

SVM-inspired Methods

- Kernel-based Methods
  - Kernel Perceptron
  - Kernel Least Squares
  - Kernel Mahalanobis Distance
  - Kernel Principal Component Analysis
- Maximum Margin Classifiers
  - Fuzzy Classifiers
  - Neural Networks

Contents

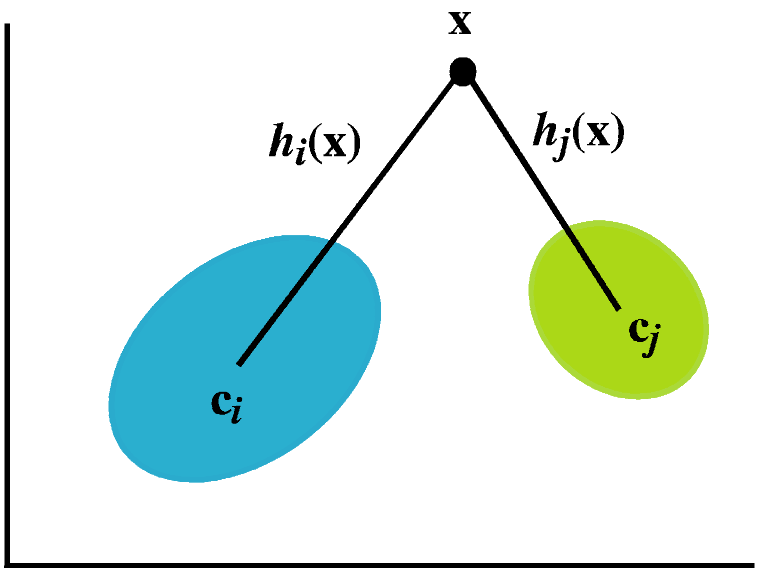
Fuzzy Classifier with Ellipsoidal Regions

Membership function

$m_i(\mathbf{x}) = \exp(- h_i^2(\mathbf{x}))$

$h_i^2(\mathbf{x}) = d_i^2(\mathbf{x})/\alpha_i$

$d_i^2(\mathbf{x}) = (\mathbf{x} - \mathbf{c}_i)^t Q_i^{-1} (\mathbf{x} - \mathbf{c}_i)$

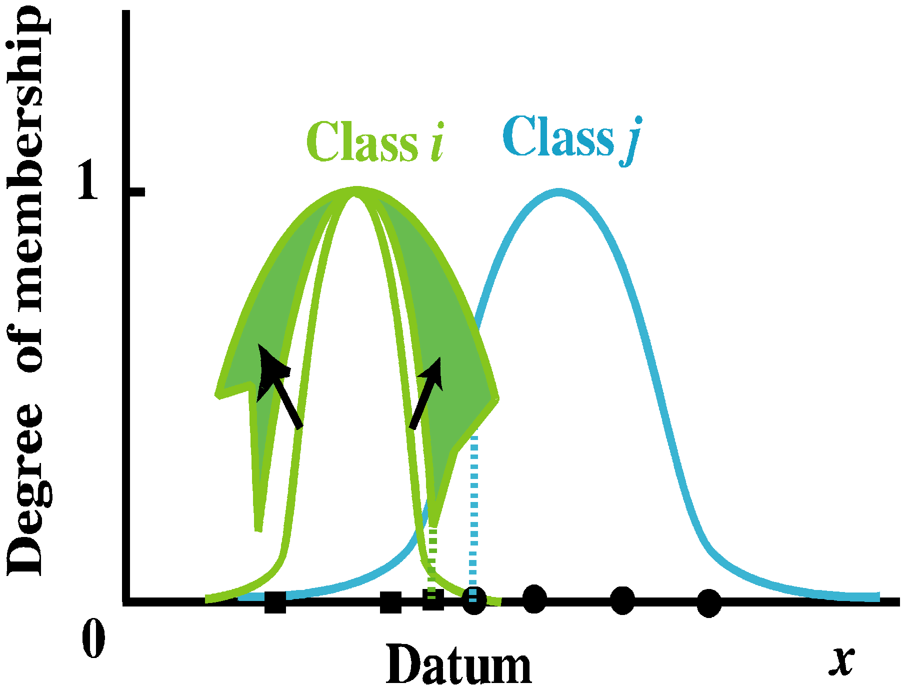


where α_i**: a tuning parameter,**

$d_i^2(\mathbf{x})$**: a Mahalanobis distance.**

Training

1. For each class calculate the center and the covariance matrix.
2. Tune the membership function so that misclassification is resolved.



Comparison of Fuzzy Classifiers with Support Vector Machines

- **Training of fuzzy classifiers is faster than support vector machines.**
- **Comparable performance for the overlapping classes.**
- **Inferior performance when data are scarce since the covariance matrix Q_i becomes singular.**

Improvement of Generalization Ability When Data Are Scarce

- **by Symmetric Cholesky factorization,**
- **by maximizing margins.**

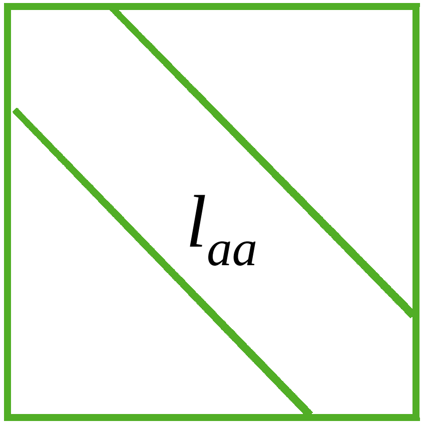
Symmetric Cholesky Factorization

In factorizing Q_i into two triangular matrices:

$Q_i = L_i L_i^t$,

if the diagonal element l_{aa} is

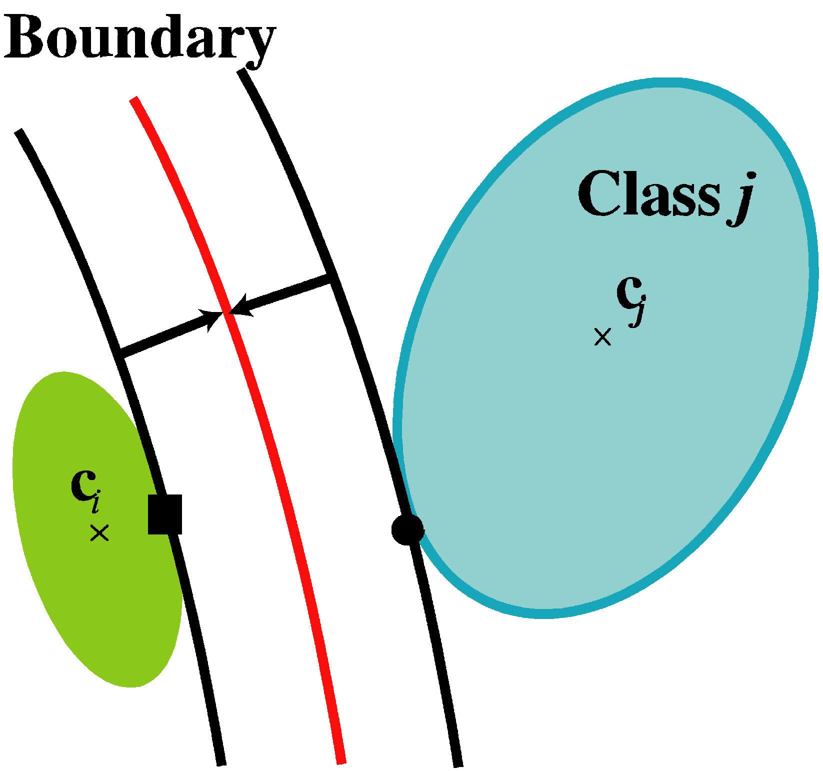
$l_{aa} < \eta$



namely, Q_i is singular, we set $l_{aa} = \eta$.

Concept of Maximizing Margins

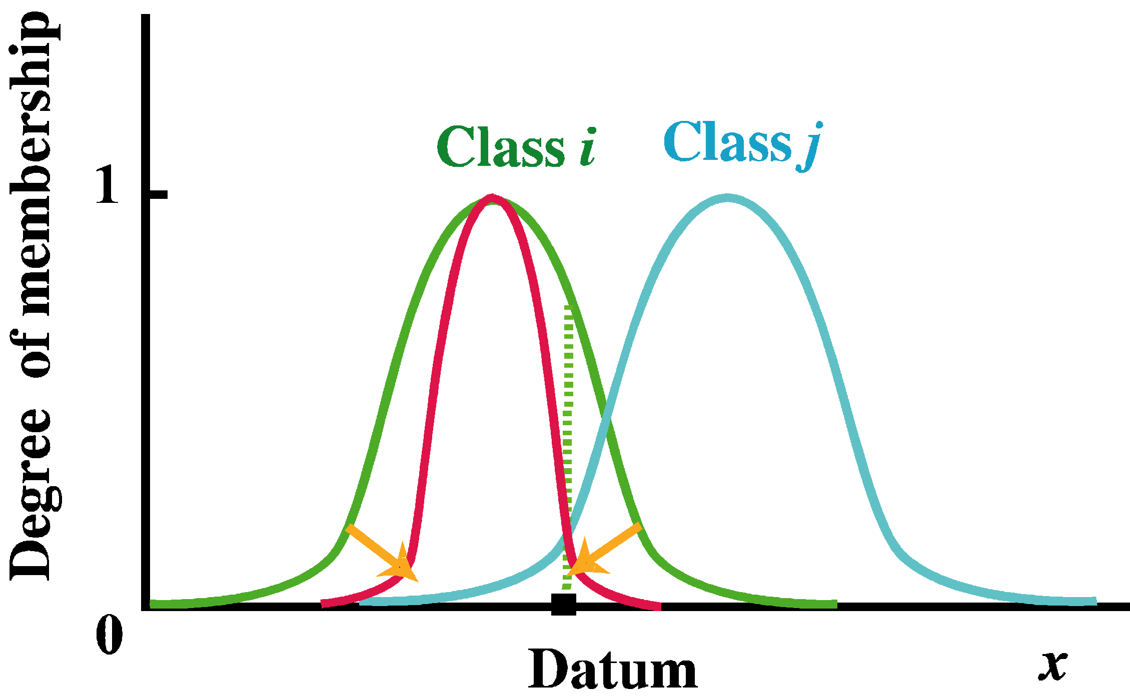
If there are no overlap between classes, we set the boundary at the middle of two classes, by tuning α_i.



Upper Bound of α_i

The class *i* datum remains correctly classified for the increase of α_i.

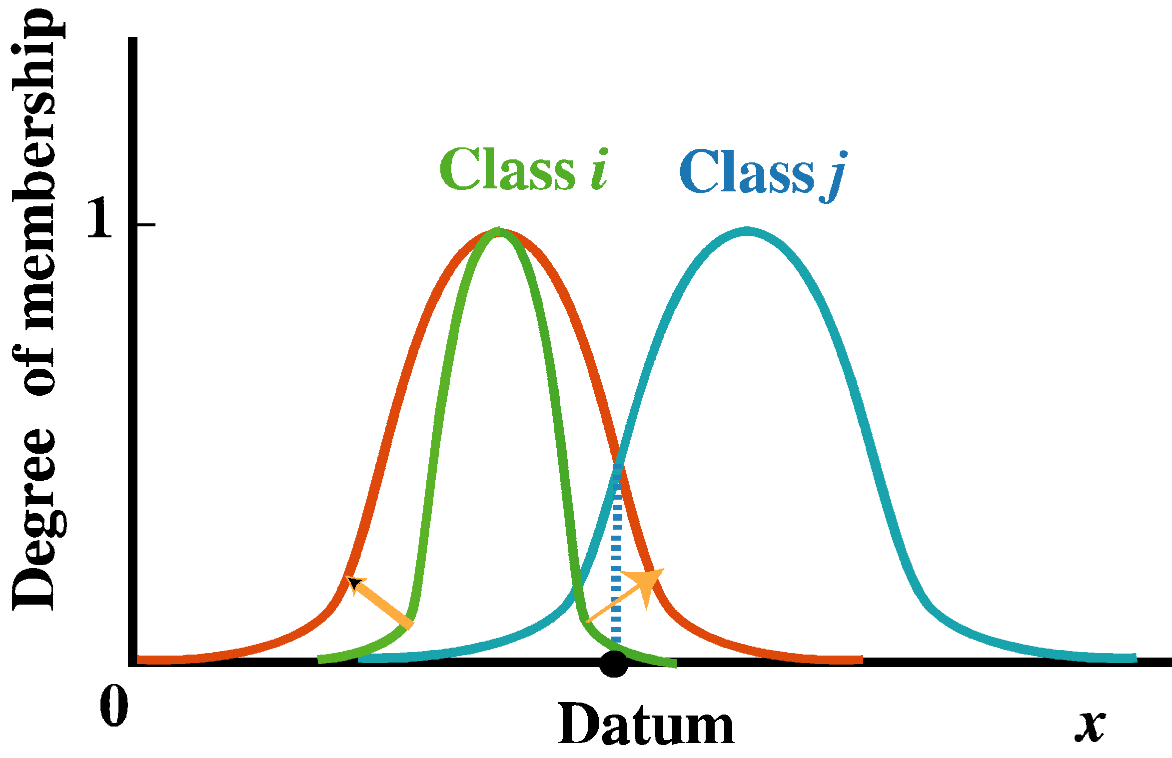
We calculate the upper bound for all the class *i* data.



Lower Bound of α_i

The class *j* datum remains correctly classified for the decrease of α_i.

We calculate the lower bound for all the data not belonging to class *i*.



Tuning Procedure

1. Calculate the upper bound L_i and lower bound U_i of α_i.

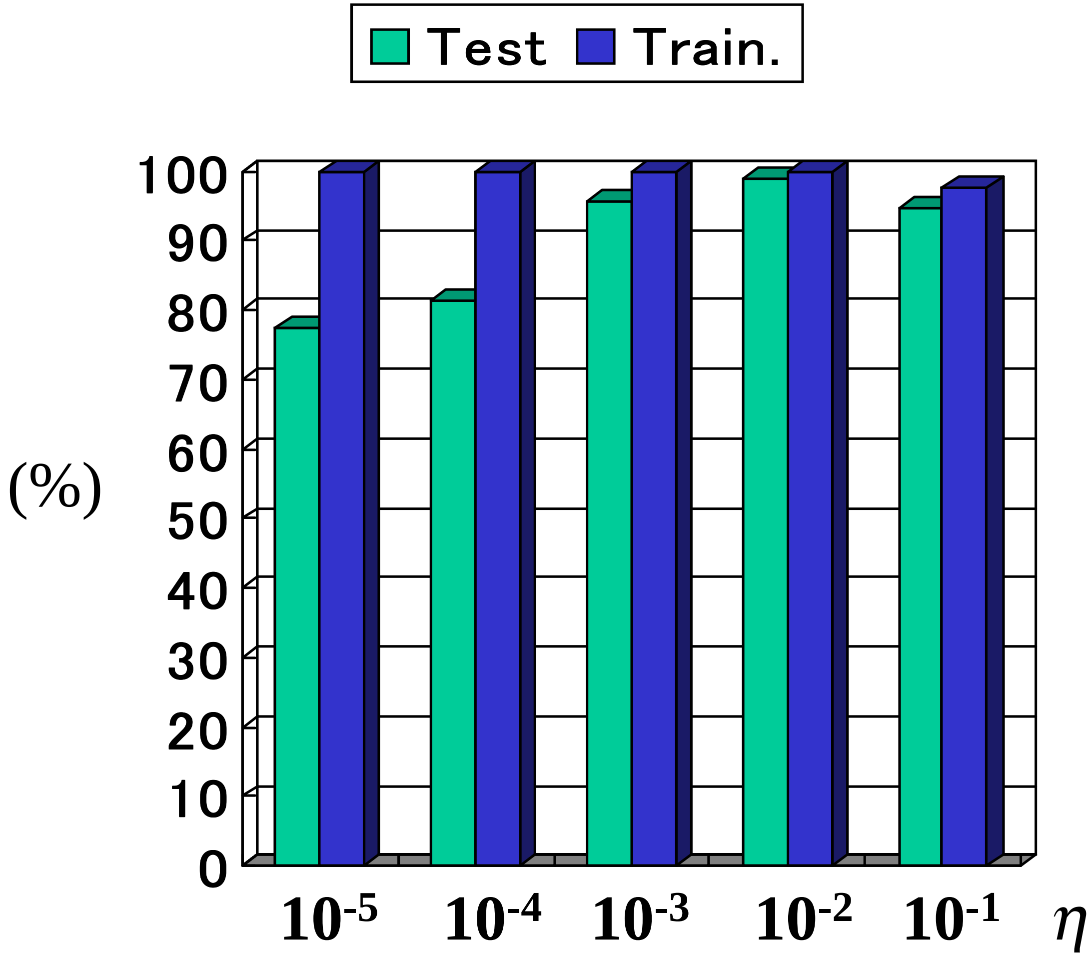
2. Set $\alpha_i = (L_i + U_i)/2$.

3. Iterate the above procedure for all the classes.

Data Sets Used for Evaluation

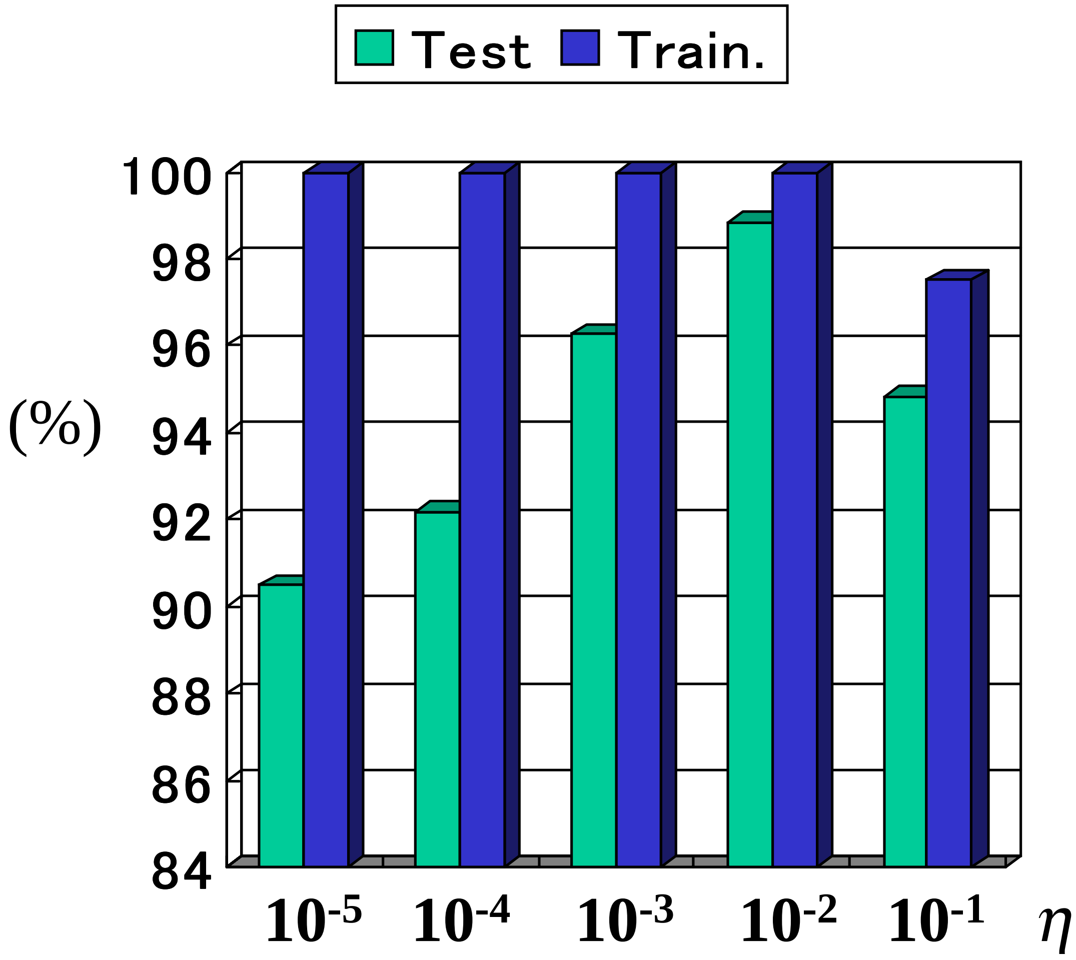
Data	Inputs	Classes	Train.	Test
H-50	50	39	4610	4610
H-105	105	38	8375	8356
H-13	13	38	8375	8356

Recognition Rate of Hiragana-50 Data
(Cholesky Factorization)



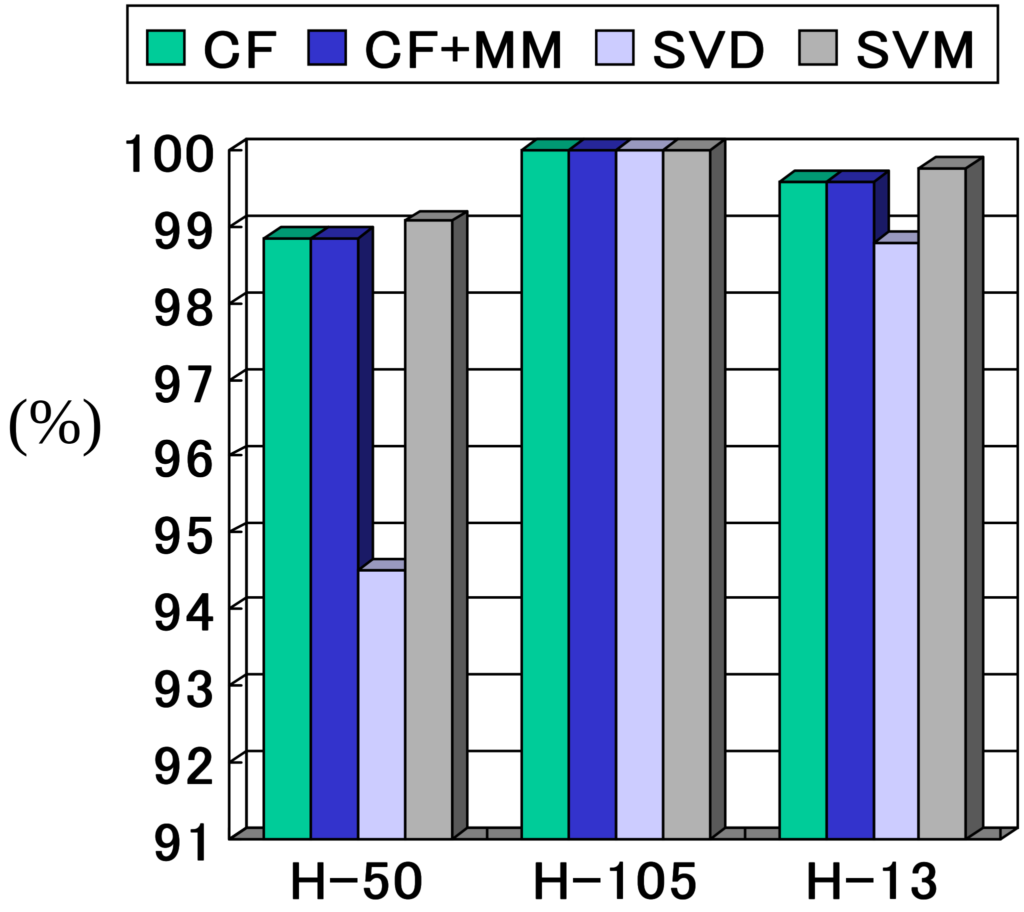
The recognition rate improved as η was increased.

Recognition Rate of Hiragana-50 Data
(Maximizing Margins)



The better recognition rate for small η.

Performance Comparison



Performance excluding SVD is comparable.

Summary

- **When the number of training data is small, the generalization ability of the fuzzy classifier with ellipsoidal regions is improved by**
  - **the Cholesky factorization with singular avoidance,**
  - **tuning membership functions when there is no overlap between classes.**
- **Simulation results show the improvement of the generalization ability.**

Contents

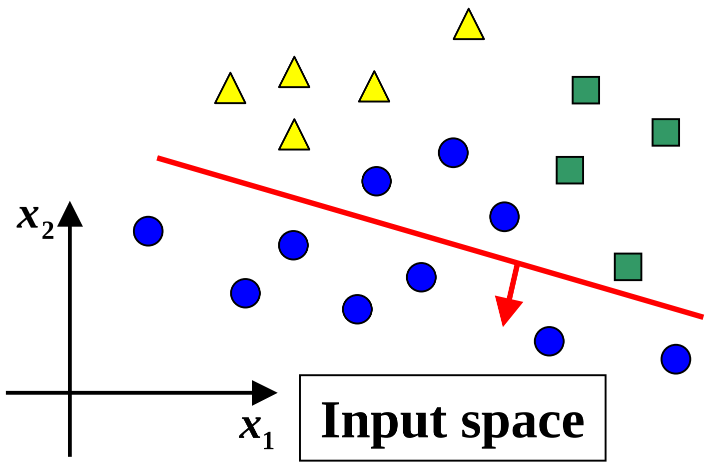
Maximum Margin Neural Networks

- **Training of multilayer neural networks (NNs)**
  - **Back propagation algorithm (BP)**
    - **- Slow training**
  - **Support vector machine (SVM) with sigmoid kernels**
    - **- High generalization ability by maximizing margins**
    - **- Restriction to parameter values**
  - **CARVE Algorithm(Young & Downs 1998)**
    - **- Efficient training method not developed yet**

CARVE Algorithm

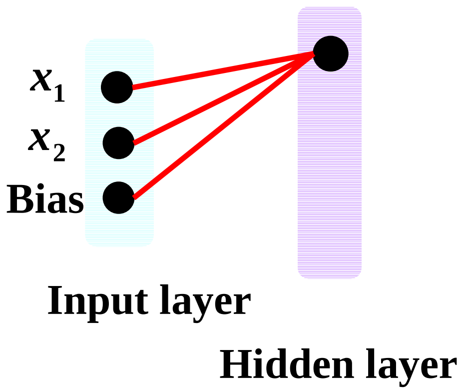
- **A constructive method of training NNs**
- **Any pattern classification problems can be synthesized in 3-layers (input layer included)**
- **Needs to find hyperplanes that separate data of one class from the others**

CARVE Algorithm (hidden layer)



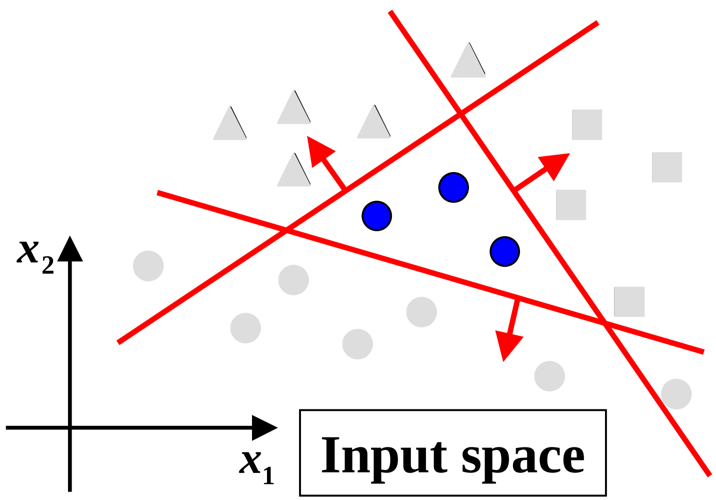
● : The class for separation

- Only ● data are separated



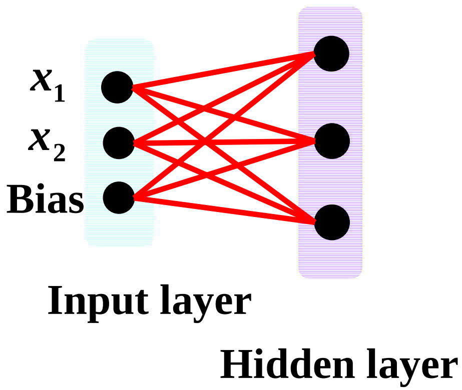
The weights between input layer and hidden layer represent the hyperplane

CARVE Algorithm (hidden layer)



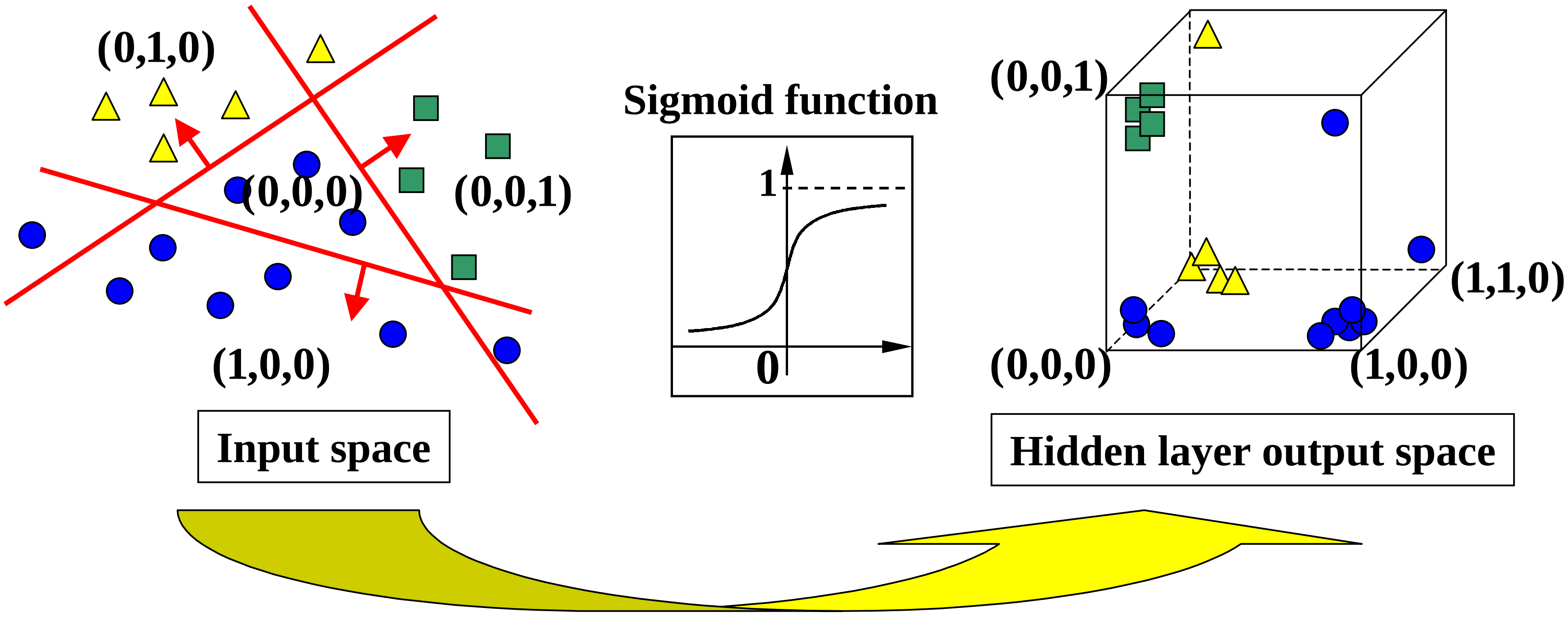
■ : The class for separation

- Only ● data are separated
- Separated data are not used in next training
- When only data of one class remain, the hidden layer training is finished



The weights between input layer and hidden layer represent the hyperplane

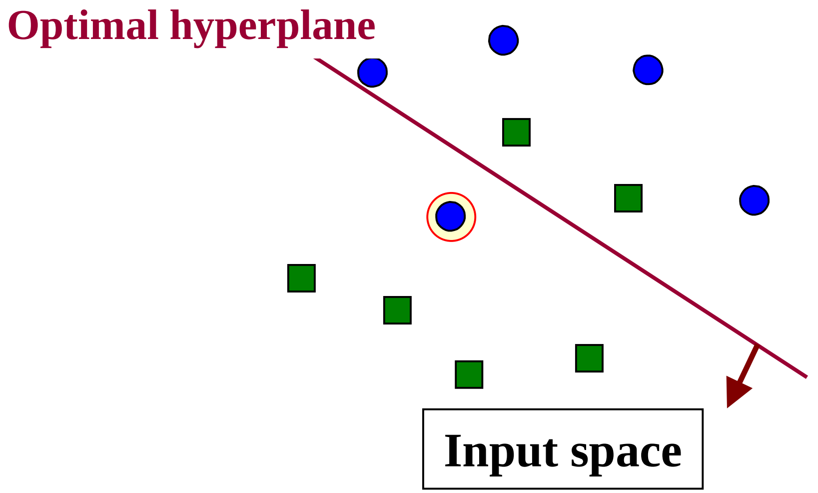
CARVE Algorithm (output layer)



All data can be linearly separable by the output layer

Proposed Method

- NN training based on CARVE algorithm
- Maximizing margins at each layer



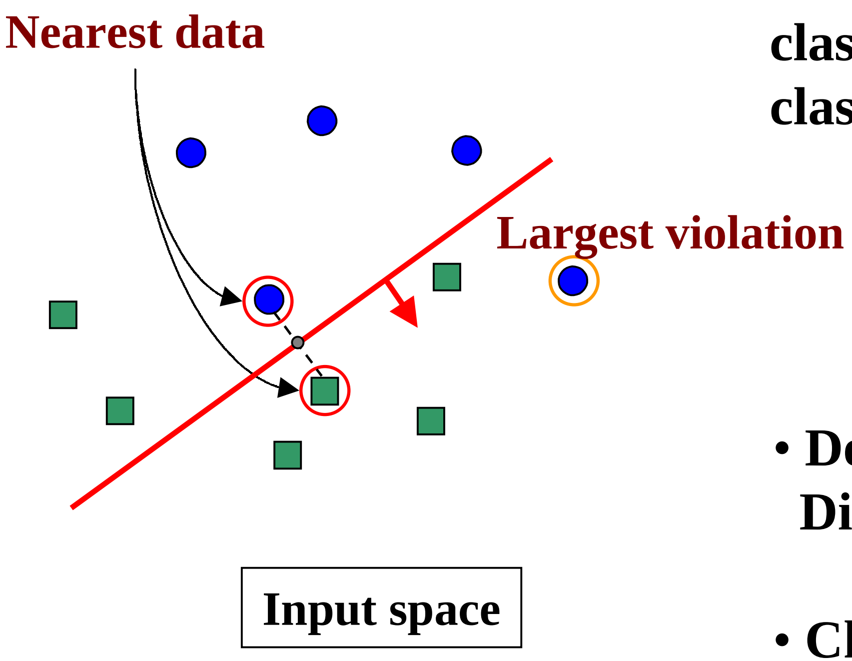
- On the positive side, data of other classes may exist by SVM training

⇩

Not appropriate hyperplanes for CARVE algorithm

- **Extend DirectSVM method in hidden layer training and use conventional SVM training in output layer**

Extension of DirectSVM



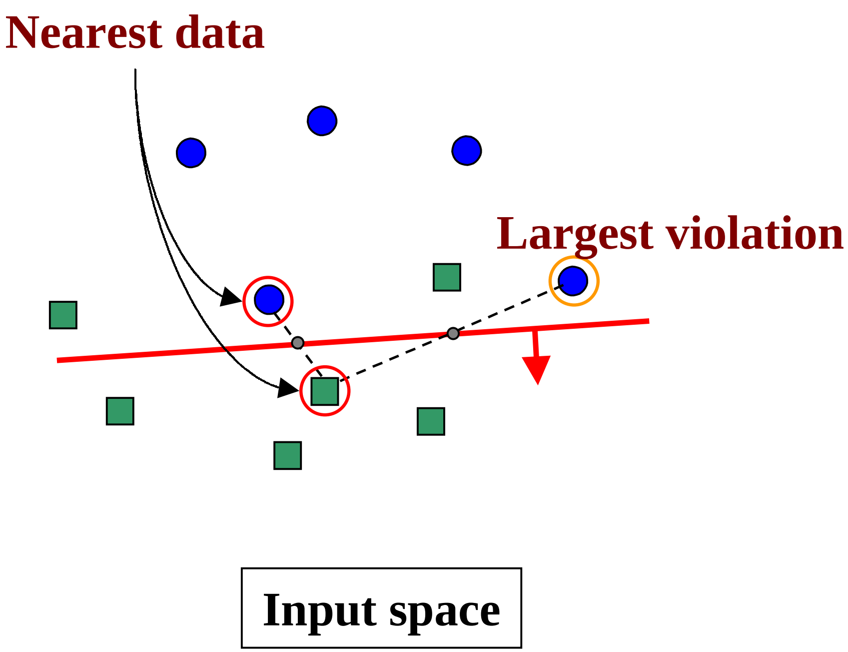
• The class labels are set so that the class for separation are +1, and other classes are -1

ex. target values { ■ : +1, ● : -1 }

• Determine the initial hyperplane by DirectSVM method

• Check the violation of the data with label –1

Extension of DirectSVM

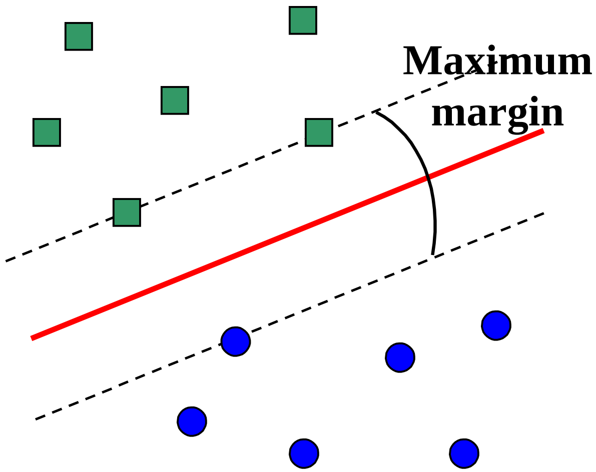


• Update the hyperplane so that the misclassified datum with -1 is classified correctly

If there are no violating data with label –1, we stop updating the hyperplane

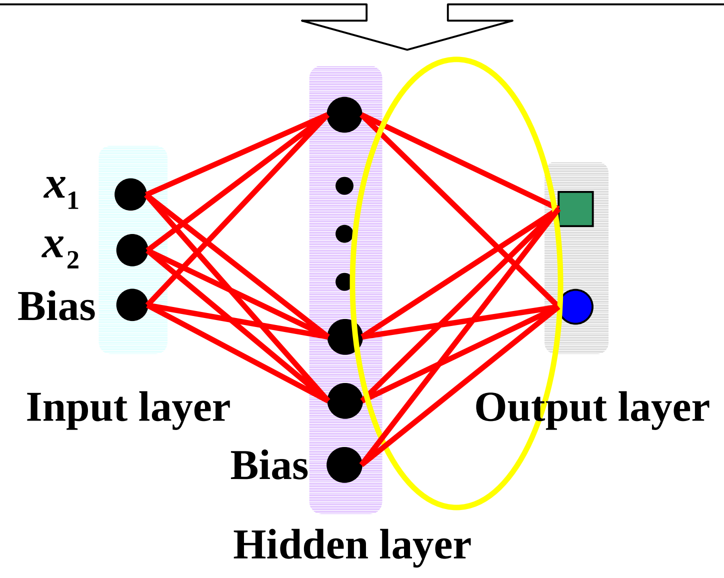
Training of output layer

- **Apply conventional SVM training**



Hidden layer output space

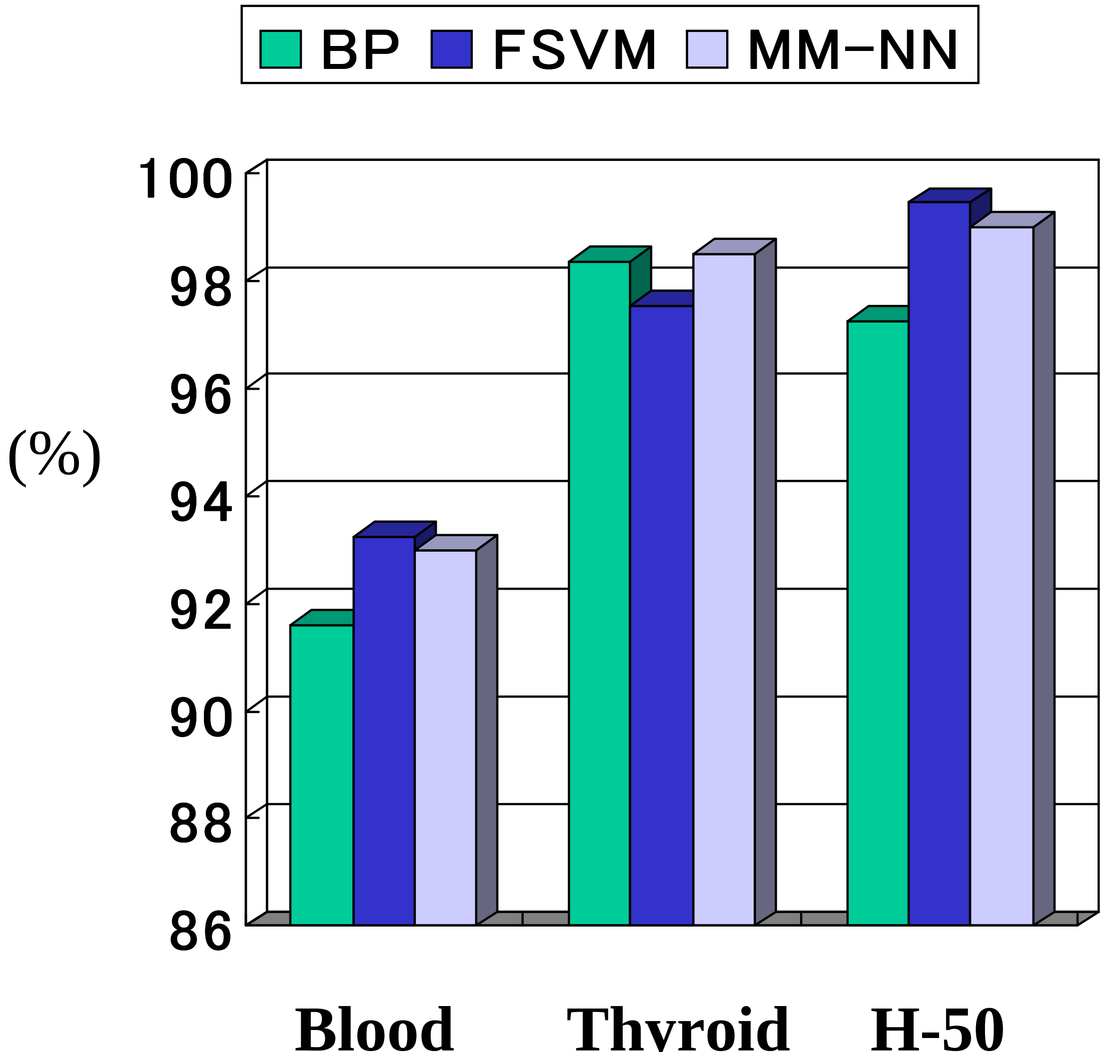
Set weights by SVM with dot product kernels



Data Sets Used for Evaluation

Data	Inputs	Classes	Train.	Test
Blood Cell	13	12	3097	3100
Thyroid	21	3	3772	3428
H-50	50	39	4610	4610
H-13	13	38	8375	8356

Performance Comparison



FSVM: 1 vs. all

MM-NN is better than BP and comparable to FSVMs.

Summary

- **NNs are generated layer by layer by the CARVE algorithm and by maximizing margins.**
- **Generalization ability is better than that of BP NN and comparable to that of SVMs.**