



An Introduction to Support Vector Machines and Multiclass Classification

Boonserm Kijsirikul
Department of Computer Engineering,
Chulalongkorn University

Outline

- Introduction to Support Vector Machines (SVMs)
- Maximum Margin Hyperplane
- Linear SVMs
- Non-Linear SVMs
- Feature Spaces and Kernels
- Multiclass Support Vector Machines

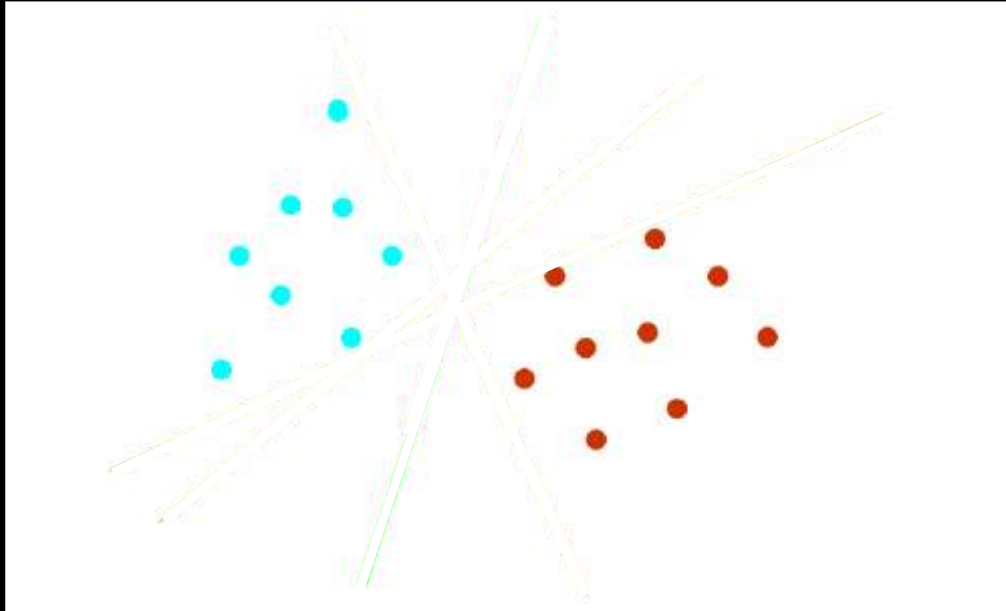
Introduction to SVMs

- Problem:
 - Find a hyperplane that correctly classifies data points from two different classes.
- Solutions: Perceptron, Neural Networks
- SVM constructs an optimal hyperplane that separates the data points of two classes as far as possible [Cortes and Vapnik, 1995].
- Issues: Linear separable, Feature space, Multiclass classification

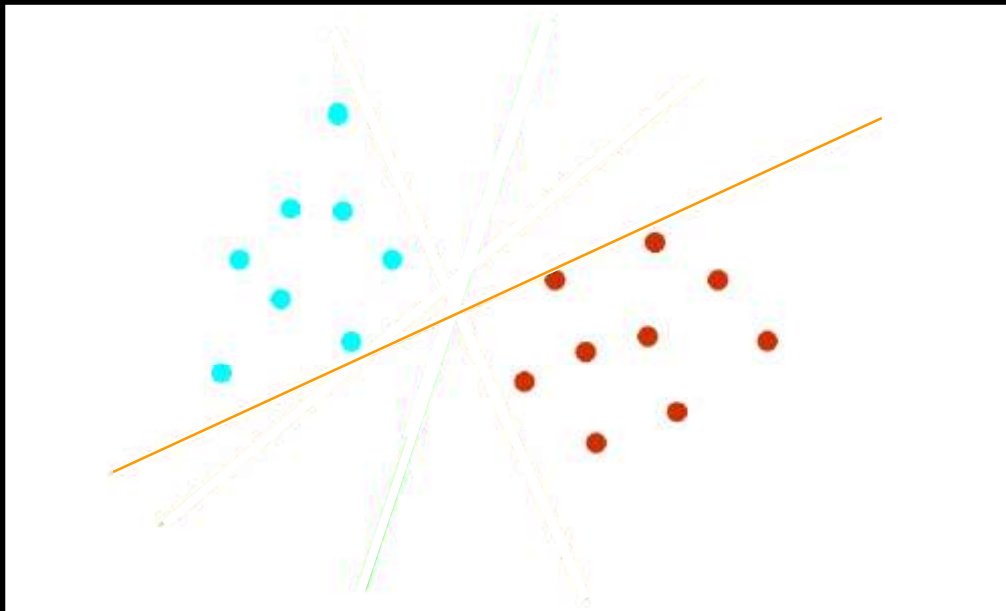
Classification of Fat Child

No	Height (cm.)	Weight (kg.)	Fat / Not fat	No.	Height (cm.)	Weight (kg.)	Fat / Not fat
1	100.0	20.0	-1	10	104.9	22.0	-1
2	100.0	26.0	1	11	105.2	20.0	-1
3	100.0	30.4	1	12	105.6	34.4	1
4	100.0	32.4	1	13	107.2	32.4	1
5	101.6	27.0	1	14	109.9	34.9	1
6	101.6	32.0	1	15	111.0	25.4	-1
7	102.0	21.0	-1	16	114.2	23.5	-1
8	103.6	29.6	1	17	115.5	36.3	1
9	104.4	30.4	1	18	117.8	26.9	-1

Maximum Margin Hyperplane

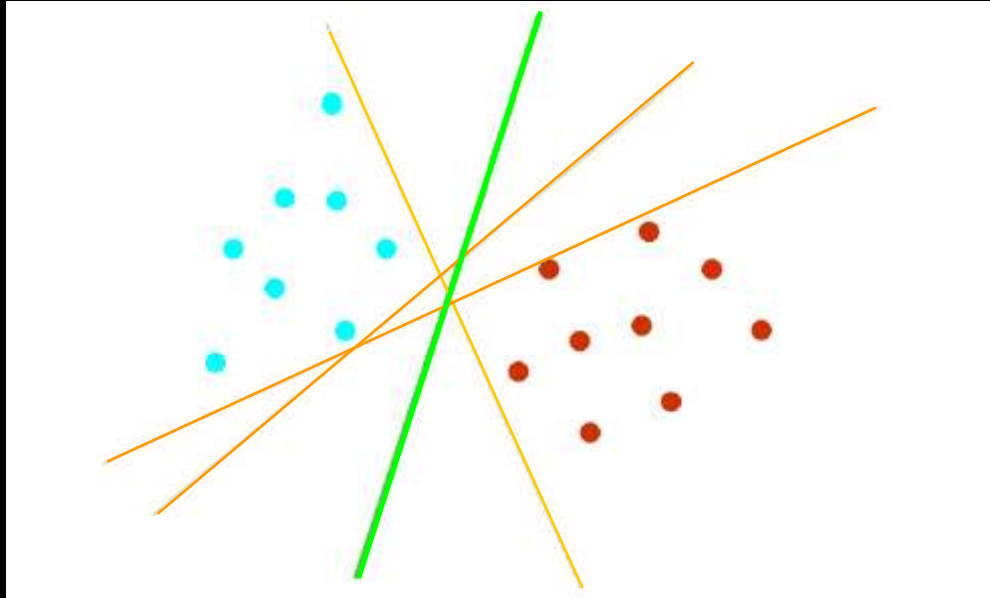


Maximum Margin Hyperplane



- A Yellow hyperplane with small margin.

Maximum Margin Hyperplane

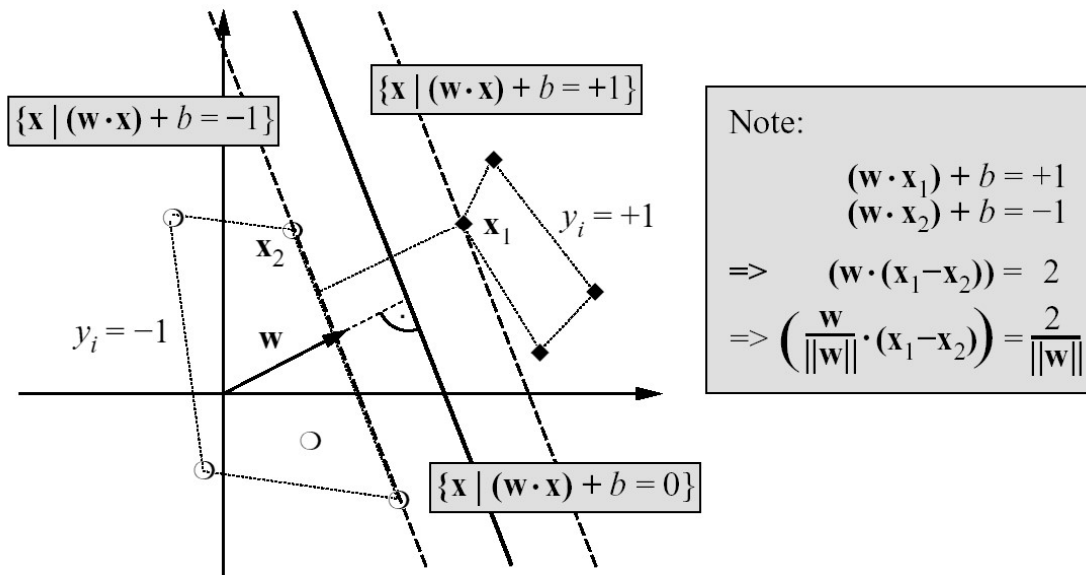


- Yellow hyperplanes with small margin.
- A Green hyperplanes with large margin.
- A better generalization is expected from the green hyperplane.

Linear SVMs

- Given a training data set
$$(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_\ell, y_\ell) \in \mathbf{R}^N \times \{\pm 1\},$$
Find $f(\mathbf{x})$ such that $f(\mathbf{x}_i) = y_i$ for all $i=1, \dots, \ell$
- Consider a hyperplane
$$(\mathbf{w} \cdot \mathbf{x}) + b = 0, \quad \mathbf{w} \in \mathbf{R}^N, b \in \mathbf{R},$$
- If we additionally require
$$\min_{i=1, \dots, \ell} |(\mathbf{w} \cdot \mathbf{x}_i) + b| = 1,$$
$$\mathbf{w} \text{ and } b \text{ be such that the point closest to the hyperplane has a distance of } 1/\|\mathbf{w}\|$$
- Therefore, $(\mathbf{w} \cdot \mathbf{x}_i) + b \geq +1$ if $y_i = +1$
$$(\mathbf{w} \cdot \mathbf{x}_i) + b \leq -1 \text{ if } y_i = -1,$$
or
$$y_i[(\mathbf{w} \cdot \mathbf{x}_i) + b] \geq +1, \quad \forall i$$

Maximum Margin



- Therefore, we want to maximize $2/\|\mathbf{w}\|$,
or minimize $\|\mathbf{w}\|/2$

Solving SVMs by Quadratic Programming

- Minimize $\frac{1}{2} \|\mathbf{w}\|^2$
subject to $y_i[(\mathbf{w} \cdot \mathbf{x}_i) + b] \geq +1$, for $i=1, \dots, \ell$
- This constrained optimization problem is dealt with by introducing Lagrange multipliers $\alpha_i \geq 0$ and a Lagrangian

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^{\ell} \alpha_i ((y_i((\mathbf{w} \cdot \mathbf{x}_i) + b) - 1))$$

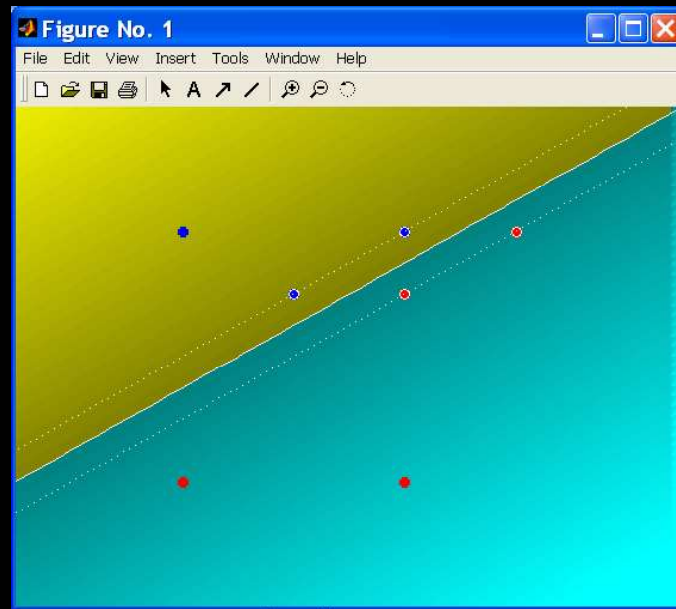
- The solution is

$$\sum_{i=1}^{\ell} \alpha_i y_i = 0 \quad \text{and} \quad \mathbf{w} = \sum_{i=1}^{\ell} \alpha_i y_i \mathbf{x}_i$$

- The solution vector thus has an expansion in terms of a subset of the training data, namely those patterns whose α_i is non-zero, called *support vectors*.
- The hyperplane decision function can be written as

$$f(\mathbf{x}) = \text{sgn}\left(\sum_{i=1}^{\ell} y_i \alpha_i \cdot (\mathbf{x} \cdot \mathbf{x}_i) + b\right)$$

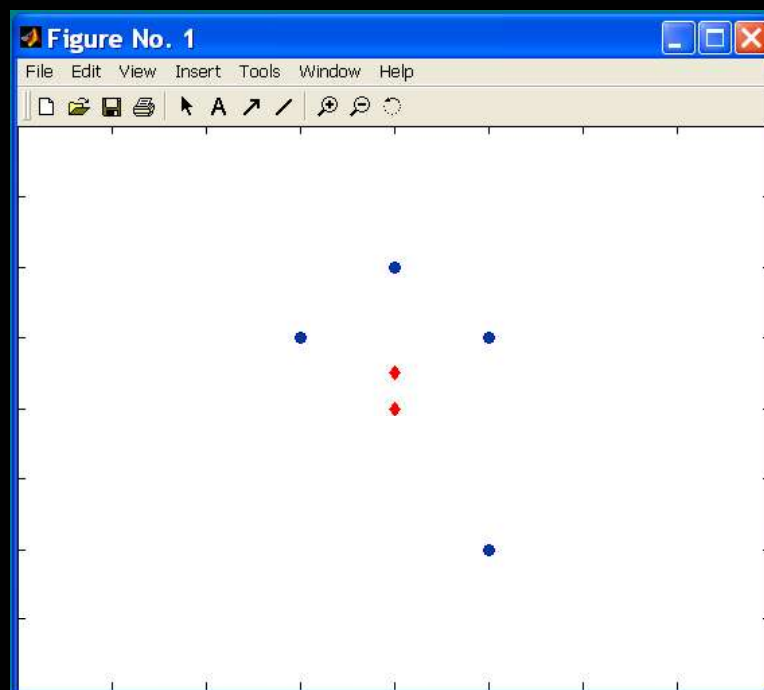
An Example of Linearly Separable Functions



- No. of support vectors = 4

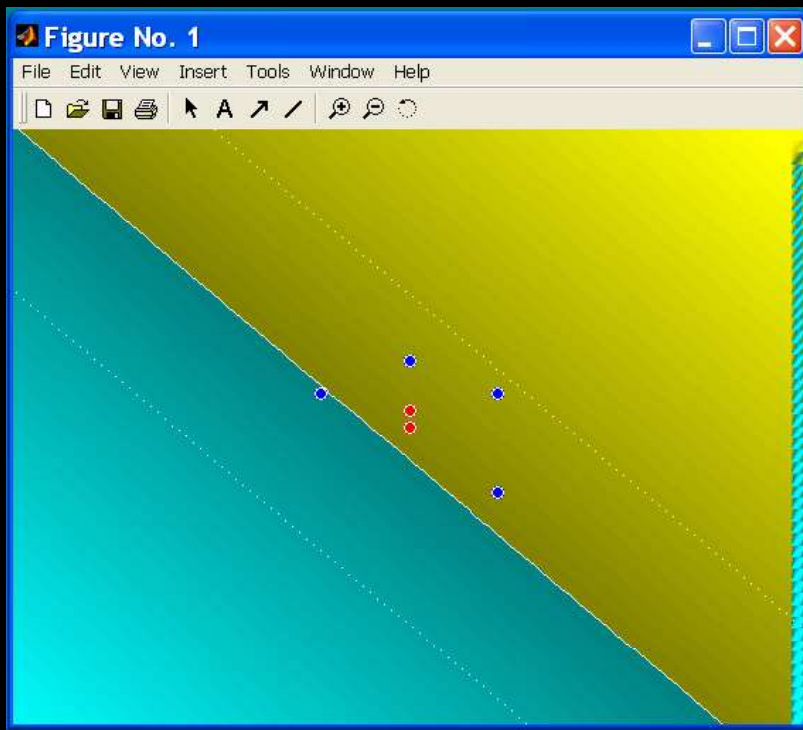
An Example of Linearly Non-Separable Functions

- An example of linearly non-separable functions



An Example of Linearly Non-Separable Functions

- In case of using the input space



Feature Spaces

- For linearly non-separable function, it is very likely that a linear separator (hyperplane) can be constructed in higher dimensional space.
- Suppose we map the data points in the input space \mathbf{R}^n into some feature space of higher dimension, \mathbf{R}^m using function Φ

$$\Phi : \mathbf{R}^n \rightarrow \mathbf{R}^m$$

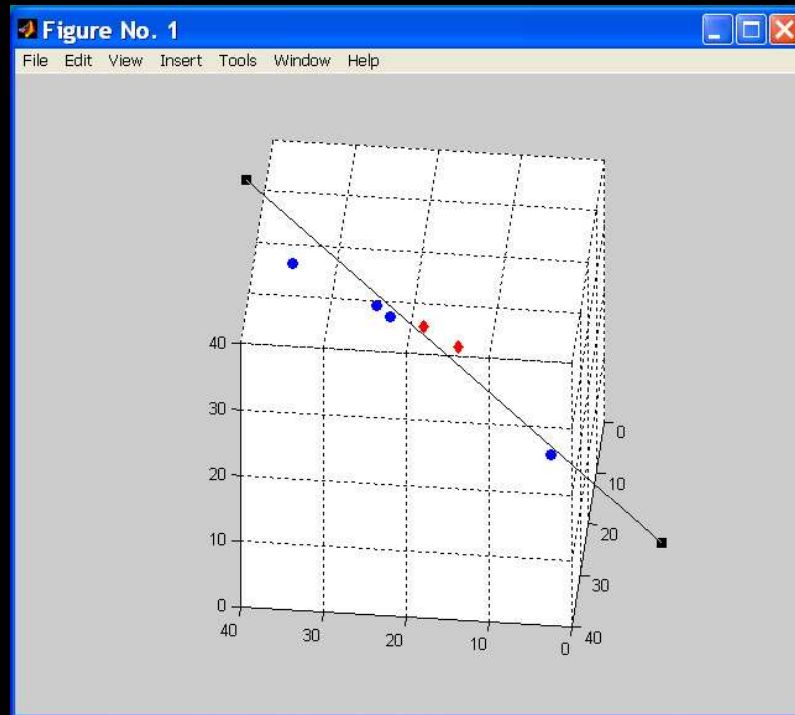
- Example:

$$\Phi : \mathbf{R}^2 \rightarrow \mathbf{R}^3$$

$$\mathbf{x} = (x_1, x_2), \quad \Phi(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2} x_1 x_2)$$

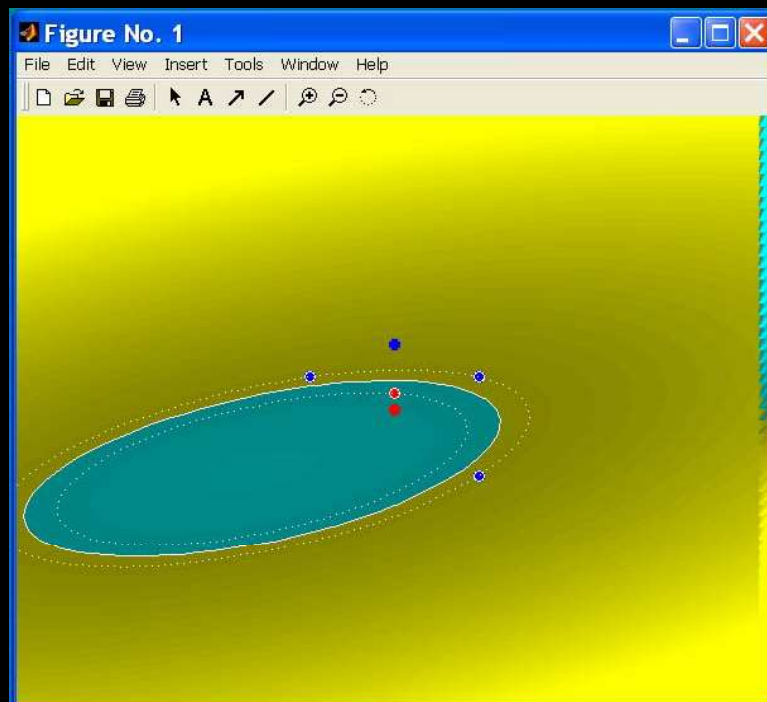
An Example of Linearly Non-Separable Functions

- In case of using the feature space: $\Phi(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2} x_1 x_2)$



An Example of Linearly Non-Separable Functions

- The corresponding non-linear function in the input space



Kernel Functions

- We need not explicitly map input data into feature space, as the construction of optimal hyperplane and the evaluation of the corresponding decision function only require the evaluation of dot products ($f(\mathbf{x}) = \text{sgn}(\sum y_i \alpha_i \cdot (\mathbf{x} \cdot \mathbf{x}_i) + b)$)
- Therefore, the dot products can be evaluated by a kernel, $k(\mathbf{x}, \mathbf{y})$, such that

$$k(\mathbf{x}, \mathbf{y}) = (\Phi(\mathbf{x}) \cdot \Phi(\mathbf{y}))$$

e.g. $k(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y})^2 = (x_1, x_2, \sqrt{2} x_1 x_2) (y_1, y_2, \sqrt{2} y_1 y_2)^T$

- Some kernel functions
 - Polynomial : $k(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + 1)^p$
 - Radial basis function : $k(\mathbf{x}, \mathbf{y}) = e^{-\|\mathbf{x} - \mathbf{y}\|^2 / 2\sigma^2}$
 - Neural network : $k(\mathbf{x}, \mathbf{y}) = \tanh(\kappa \mathbf{x} \cdot \mathbf{y} - d)$

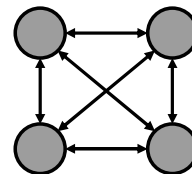
Multiclass SVMs

- SVMs are originally designed to be binary classifiers (discriminating between two classes).
- Thus, SVMs need modification in order to deal with real-world multiclass problems.
- Previous Methods for Multiclass SVMs.
 - One-against-the-Rest (1-v-R)
 - One-against-one (1-v-1)
 - Max Wins algorithm
 - Decision Directed Acyclic Graphs (DDAG)
 - Etc.

One-against-the-Rest

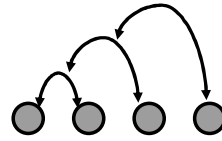
- For N -class classification, construct N binary classifiers.
- Train the i^{th} classifier with all examples in the i^{th} class as positive, and the other examples as negative labels.
- Output the class corresponding to the classifier with the highest output value.

Max Wins Algorithm



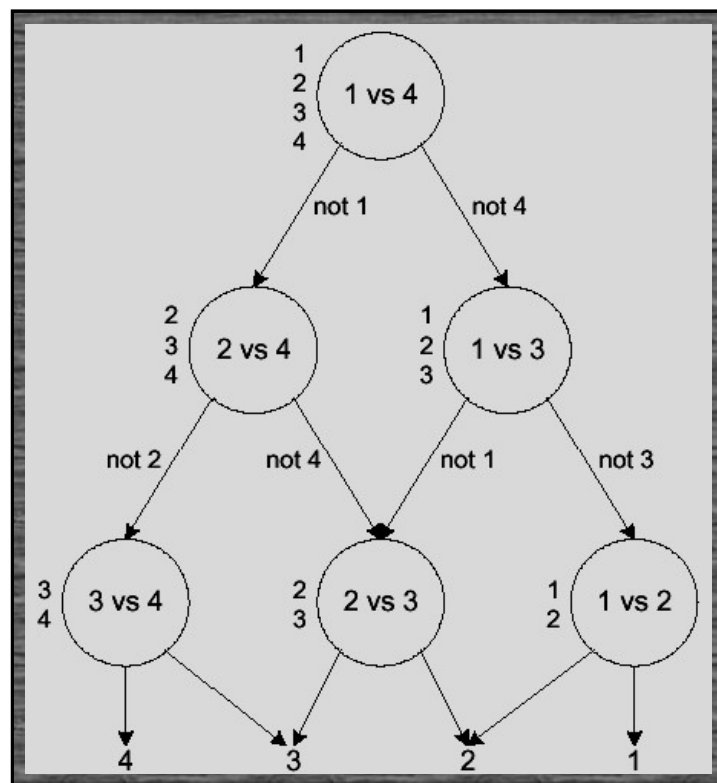
- One-against-one (1-v-1)
 - Construct all possible binary classifiers.
 - For N classes, there will be $N(N-1)/2$ classifiers.
 - Each classifier is trained on 2 out of N classes.
- Max Wins (a kind of 1-v-1)
 - A test example is classified by all classifiers.
 - Each classifier provides one vote for its preferred class.
 - The majority vote is the final output.
 - Accurate but slow.

DDAG



- Decision Directed Acyclic Graphs
- Concept – remove wrong classes one-by-one.
- Construct $N(N-1)/2$ classifiers, but require only $N-1$ times of binary classification.
($\ll N(N-1)/2$).
- Faster than Max Wins (require less evaluation time).

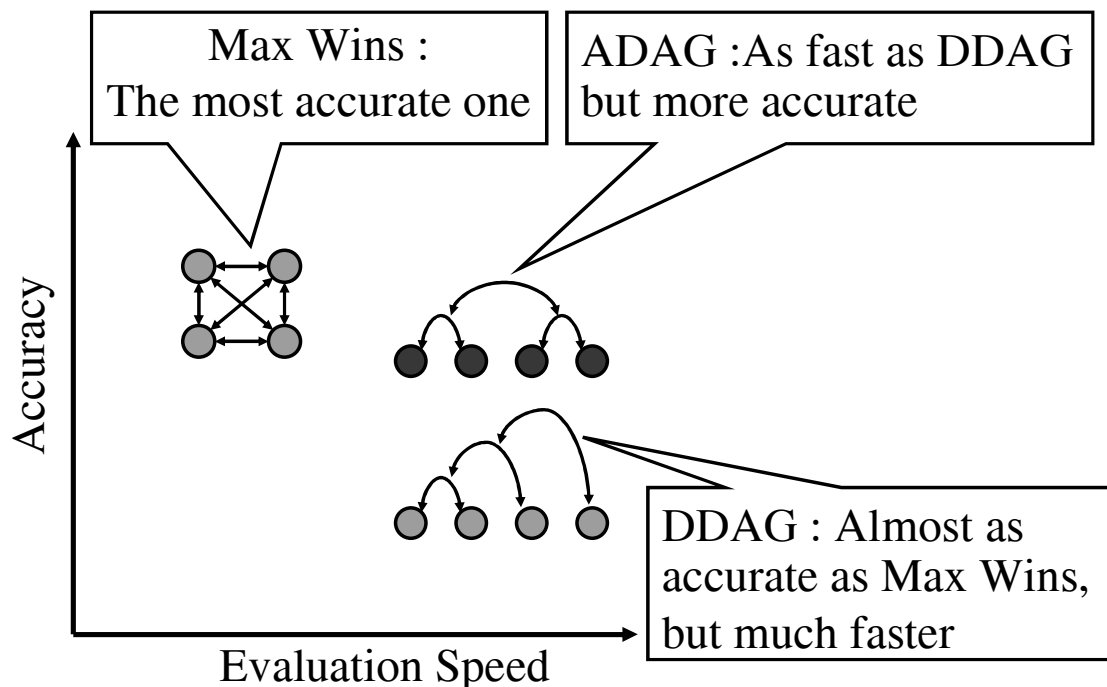
DDAG Architecture



Weakness of DDAG

- Number of node evaluations for the correct class is unnecessarily high.
- If the correct class is evaluated at the root node, it is tested against the other classes for $N-1$ times.
- The higher the number of times the correct class is tested, the higher misclassification will be.

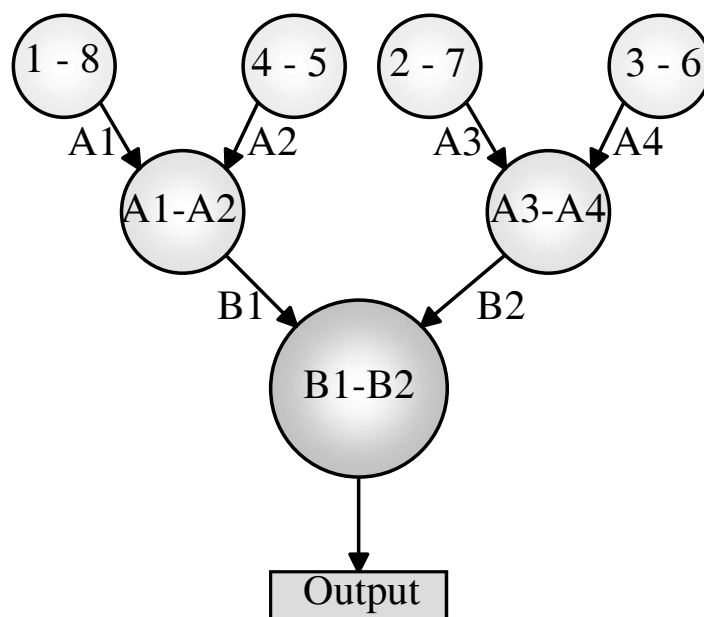
Positioning of Adaptive Directed Acyclic Graphs (ADAG)



ADAG

- Adaptive Directed Acyclic Graphs
- The architecture is like a paired knock-out competition.
- In each round, two classes are paired and play a knock-out match.
- The winners proceed to the next round.
- The champion will be the classification result.

ADAG Architecture (8 classes)



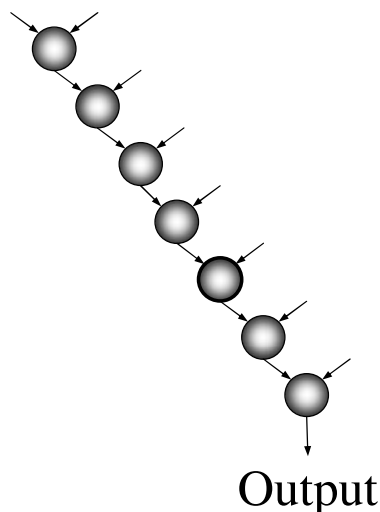
More Accuracy

- ADAG gives higher accuracy than DDAG,
 - when the number of classes increases, and
 - when each binary classifier is not very accurate.
- Reduction of levels of evaluation
= Reduction of cumulative error.

More Accuracy

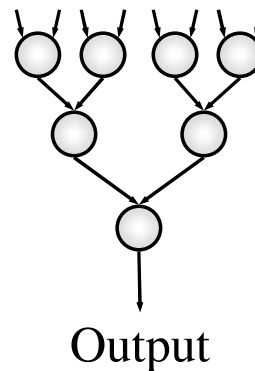
- Suppose each classifier has 1% error rate.

DDAG



$$\text{Error} = 1 - 0.99^7 = 6.79\%$$

ADAG



$$\text{Error} = 1 - 0.99^{\log_2(8)} = 2.97\%$$

Expected Accuracy of DDAG & ADAG

- Let p be the probability that the correct class will be eliminated from the implementation list, when it is tested against another class.
- Let the probability of one of any two classes, except for the correct class, being eliminated from the list be 0.5.
- Assume the probability distribution of the position of the correct class in the list is uniform.
- Let N be the number of classes.

Theorems

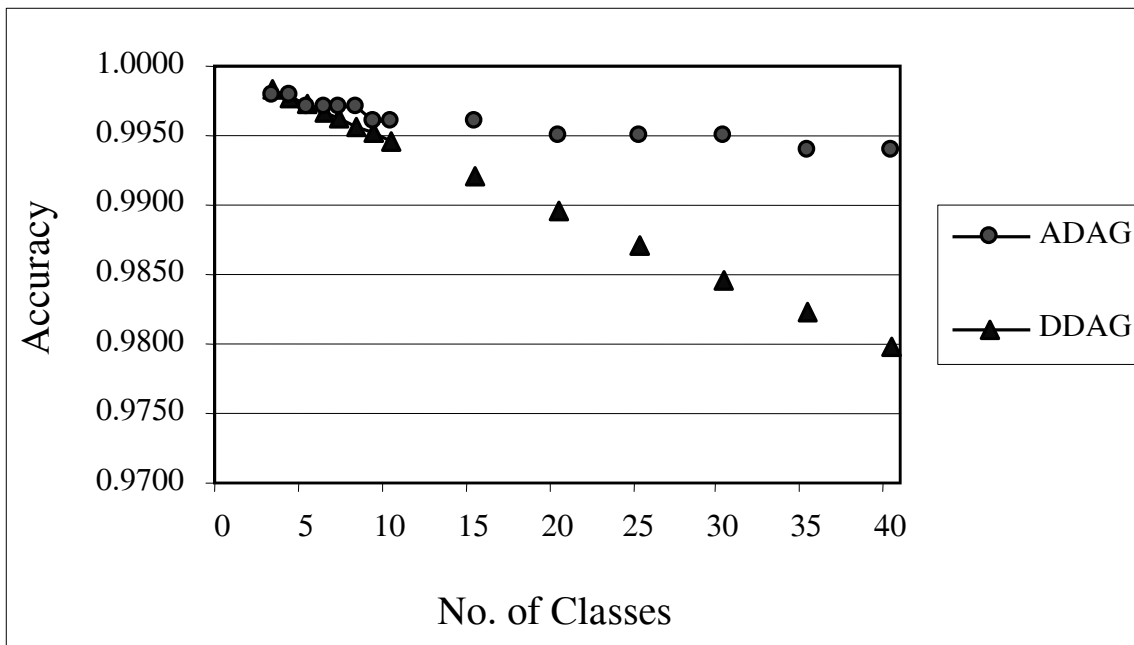
- The expected accuracy of DDAG is

$$\frac{1}{N} \left[\frac{1-p}{p} + (1-p)^{N-1} - \frac{(1-p)^N}{p} \right]$$

- The expected accuracy of ADAG is

$$\left(\frac{2N - 2^{\lceil \log_2 N \rceil}}{N} \right) (1-p)^{\lceil \log_2 N \rceil} + \left(\frac{2^{\lceil \log_2 N \rceil} - N}{N} \right) (1-p)^{\lceil \log_2 N \rceil - 1}$$

Expected Accuracy of DDAG & ADAG



- $p = 0.1\%$

Experiments

- 2 data sets
 - Thai vowel data set (12 classes)
 - UCI letter data set (26 classes)
- ADAG should have much advantage when the number of classes increases.
- Another factor is the accuracy of binary classifiers (the value of $(1-p)$ in previous theorems).

Experiments: Thai Vowel Data Set

- 12-Class
- Use Polynomial kernel and RBF kernel
 - Polynomial : $|(\mathbf{x} \cdot \mathbf{y} + 1)/72|^d$
 - RBF : $\exp(-|\mathbf{x} - \mathbf{y}|^2/72c)$
- For each value of d or c
 - no. of experiments = 20,000 with several different sequences of classes in the implementation list chosen randomly

Thai Vowel

- No. of training examples: 6,192
- No. of test examples: 3,096

Polynomial			
d	ADAG	DDAG	DIFF
6	86.12	86.09	0.03
7	86.12	86.08	0.04
8	85.98	85.91	0.07
9	85.39	85.33	0.06
10	85.09	85.05	0.04
11	84.58	84.55	0.03
12	84.30	84.27	0.03

Thai Vowel (Cont.)

RBF			
c	ADAG	DDAG	DIFF
0.1	74.32	74.31	0.01
0.2	84.50	84.48	0.02
0.3	86.55	86.52	0.03
0.4	86.77	86.75	0.02
0.5	86.64	86.63	0.01

Experiments: UCI Letter Data Set

- 26-Class
- Use Polynomial kernel and RBF kernel
- No. of experiments = 50,000 with several different sequences of classes in the implementation list chosen randomly

UCI Letter

Polynomial			
d	ADAG	DDAG	DIFF
1	83.83	83.31	0.52 ⁺⁺⁺⁺
2	95.59	95.17	0.42 ⁺⁺⁺⁺
3	95.96	95.51	0.45 ⁺⁺⁺⁺
4	95.88	95.46	0.42 ⁺⁺⁺⁺
5	95.83	95.34	0.49 ⁺⁺⁺⁺
6	95.35	94.87	0.48 ⁺⁺⁺⁺
7	94.78	94.25	0.53 ⁺⁺⁺⁺
8	93.83	93.14	0.69 ⁺⁺⁺⁺
9	93.22	92.45	0.77 ⁺⁺⁺⁺
10	92.42	91.62	0.80 ⁺⁺⁺⁺

- ⁺⁺⁺⁺ denotes 99.99% confidence level for difference

UCI Letter (Cont.)

RBF			
c	ADAG	DDAG	DIFF
0.1	90.81	90.64	0.17 ⁺⁺
0.2	94.13	93.97	0.16 ⁺⁺⁺
0.3	95.46	95.36	0.10 ⁺⁺
0.4	96.31	96.21	0.10 ⁺⁺
0.5	96.48	96.39	0.09 ⁺⁺
0.6	97.03	96.97	0.06 ⁺
0.7	97.28	97.22	0.06 ⁺
0.8	97.20	97.15	0.05 ⁺

- ⁺⁺⁺ denotes 99.00% confidence level for difference
- ⁺⁺ denotes 95.00% confidence level for difference
- ⁺ denotes 90.00% confidence level for difference

UCI Letter (Cont.)

RBF			
c	ADAG	DDAG	DIFF
0.9	97.27	97.22	0.05 ⁺
1.0	97.38	97.34	0.04
1.5	97.59	97.55	0.04
2.0	97.63	97.62	0.01
2.5	97.76	97.76	0.00
3.0	97.91	97.90	0.01
3.5	97.84	97.84	0.00
4.0	97.80	97.79	0.01

- ⁺ denotes 90.00% confidence level for difference