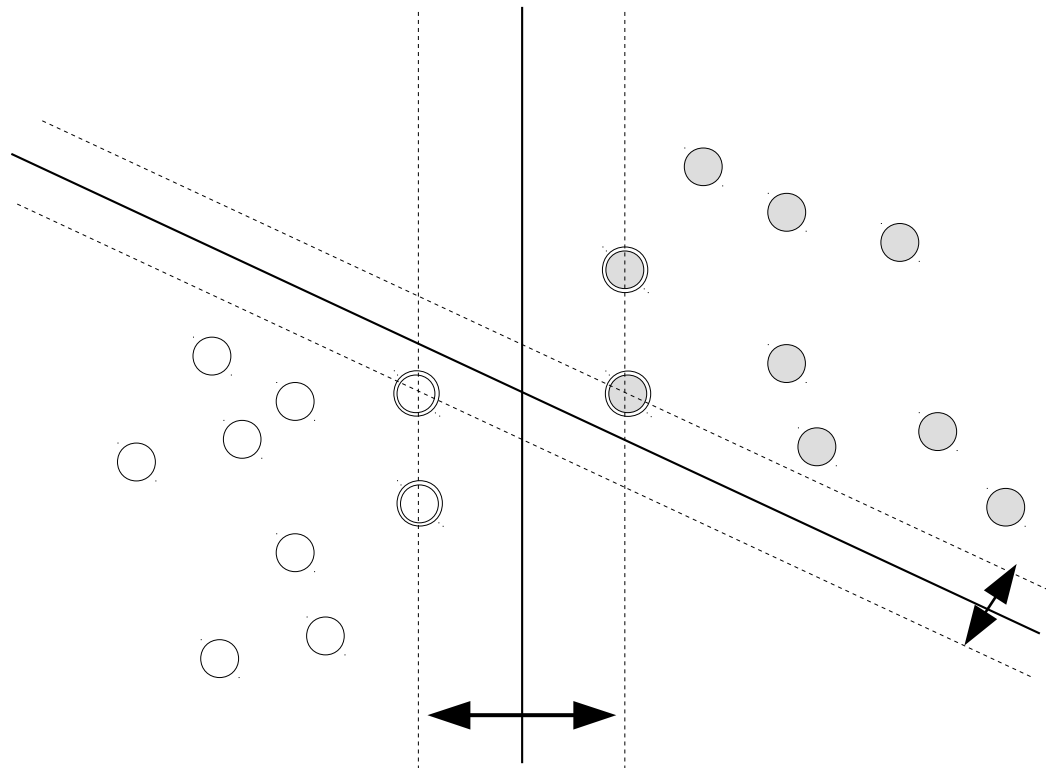


## 7. 識別 - サポートベクトルマシン -

- マージンを最大化する識別面を求める

識別面と、最も  
近いデータとの  
距離



○ ○ : サポートベクトル

## 7.1 問題の定式化

- 学習データ

$$\{(\boldsymbol{x}_i, y_i)\} \quad i = 1, \dots, N, \quad y_i = 1 \text{ or } -1$$

- 識別面の式

$$\boldsymbol{w} \cdot \boldsymbol{x}_i + w_0 = 0$$

- 識別面の制約（係数を定数倍しても平面は不変）

$$\min_{i=1, \dots, N} |\boldsymbol{w} \cdot \boldsymbol{x}_i + w_0| = 1$$

- 学習パターンと超平面との最小距離

$$\min_{i=1, \dots, N} \text{Dist}(\boldsymbol{x}_i) = \min_{i=1, \dots, N} \frac{|\boldsymbol{w} \cdot \boldsymbol{x}_i + w_0|}{\|\boldsymbol{w}\|} = \frac{1}{\|\boldsymbol{w}\|}$$

点と直線の距離の公式

$$r = \frac{|ax + by + c|}{\sqrt{a^2 + b^2}}$$

## 7.1 問題の定式化

- 目的関数：  $\min \frac{1}{2} ||\boldsymbol{w}||^2$
- 制約条件：  $y_i(\boldsymbol{w} \cdot \boldsymbol{x}_i + w_0) \geq 1 \quad i = 1, \dots, N$
- 解法：ラグランジュの未定乗数法
  - 問題  $\min f(x) \quad s.t. \quad g(x) = 0$
  - ラグランジュ関数  $L(x, \alpha) = f(x) + \alpha g(x)$ 
    - $x, \alpha$  で偏微分して 0 になる値が極値

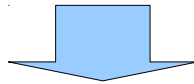
## 7.1 問題の定式化

- 計算

$$L(\boldsymbol{w}, w_0, \alpha) = \frac{1}{2} \|\boldsymbol{w}\|^2 - \sum_{i=1}^N \alpha_i (y_i (\boldsymbol{w} \cdot \boldsymbol{x} + w_0) - 1)$$

$$\frac{\partial L}{\partial w_0} = 0 \quad \Rightarrow \quad \sum_{i=1}^N \alpha_i y_i = 0$$

$$\frac{\partial L}{\partial \boldsymbol{w}} = 0 \quad \Rightarrow \quad \boldsymbol{w} = \sum_{i=1}^N \alpha_i y_i \boldsymbol{x}_i$$

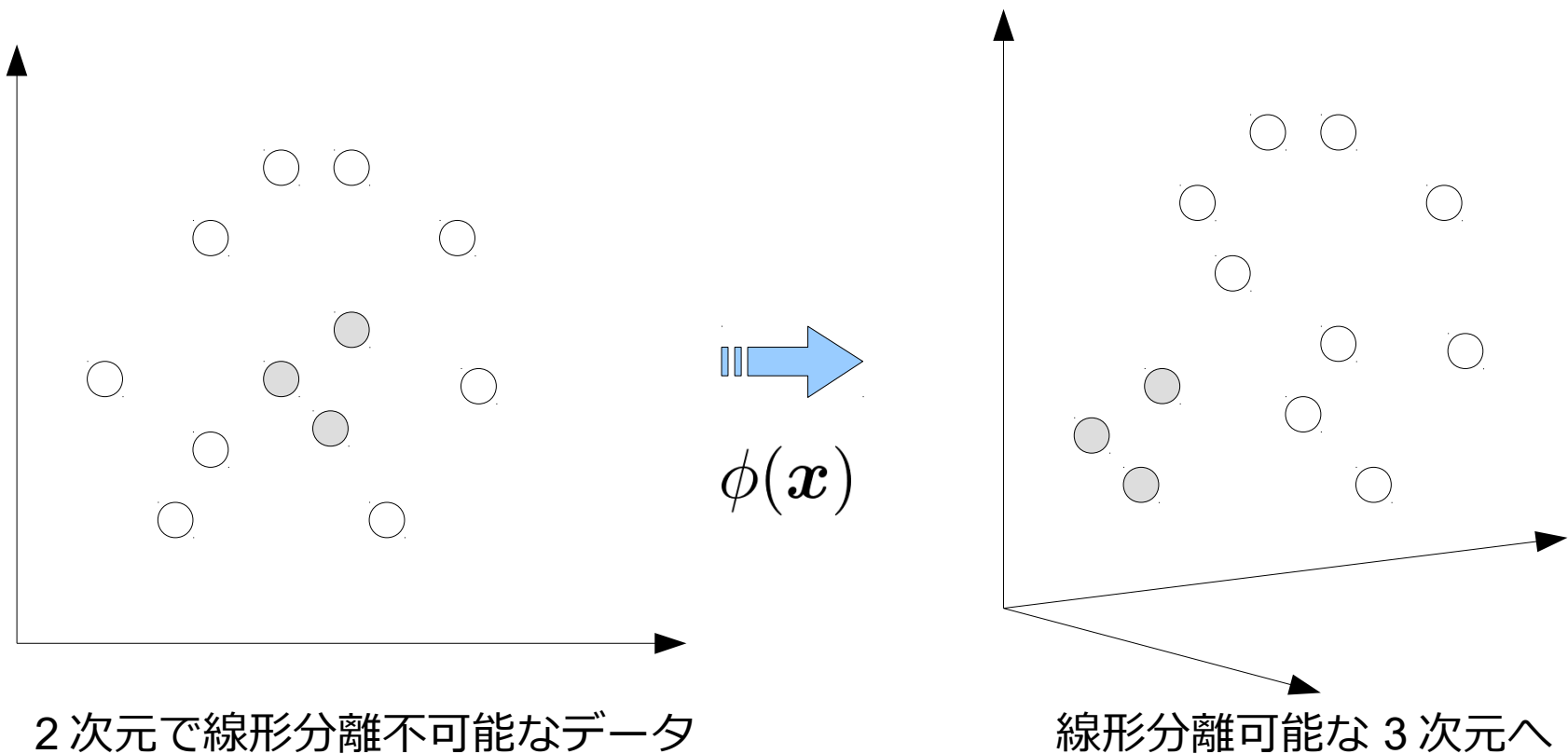


$$L(\alpha) = \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i \cdot \boldsymbol{x}_j - \sum_{i=1}^N \alpha_i$$

$\alpha$  についての  
2次計画問題

## 7.2 カーネル関数の利用

- 特徴ベクトルの次元を増やす



ただし、元の空間でのデータ間の  
距離関係は保持するように

## 7.2 カーネル関数の利用

- 非線形変換関数：  $\phi(\boldsymbol{x})$
- カーネル関数
  - 元の空間での距離が変換後の空間の内積に対応

$$K(\boldsymbol{x}, \boldsymbol{x}') = \phi(\boldsymbol{x}) \cdot \phi(\boldsymbol{x}')$$

- カーネル関数の例

- 多項式カーネル  $K(\boldsymbol{x}, \boldsymbol{x}') = (\boldsymbol{x} \cdot \boldsymbol{x}' + 1)^p$

- ガウシアンカーネル  $K(\boldsymbol{x}, \boldsymbol{x}') = \exp - \left( \frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{\sigma^2} \right)$

この形であれば、対応する非線形変換が存在することが数学的に保証されている

## 7.3 カーネル関数を用いた SVM

- 変換後の識別関数：  $g(\mathbf{x}) = \mathbf{w} \cdot \phi(\mathbf{x}) + w_0$
- SVM で求めた  $\mathbf{w}$  の値を代入

$$\begin{aligned} g(\mathbf{x}) &= \sum_{i=1}^N \alpha_i y_i \phi(\mathbf{x}) \cdot \phi(\mathbf{x}_i) + w_0 \\ &= \sum_{i=1}^N \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + w_0 \end{aligned}$$

非線形変換の  
式は不要！！！！

**カーネルトリック**

# Weka の SMO

The image shows two overlapping windows from the Weka software. The background window is 'Weka Explorer' and the foreground window is 'weka.gui.GenericObjectEditor'.

**Weka Explorer (Background):**

- Buttons: Preprocess, Classify, Cluster, Associate, Select attributes
- Classifier: Choose SMO -C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K \*weka.classifi
- Test options:
  - ☐ Use training set
  - ☐ Supplied test set (Set...)
  - ☒ Cross-validation Folds 10
  - ☐ Percentage split % 66
- More options...
- (Nom) class
- Start Stop
- Result list (right-click for options): 16:00:33 - functions.SMO
- Status: OK

**weka.gui.GenericObjectEditor (Foreground):**

weka.classifiers.functions.SMO

About: Implements John Platt's sequential minimal optimization algorithm for training a support vector classifier. [More] [Capabilities]

buildLogisticModels: False

c: 1.0

checksTurnedOff: False

debug: False

doNotCheckCapabilities: False

epsilon: 1.0E-12

filterType: Normalize training data

kernel: Choose PolyKernel -E 1.0 -C 250007

numFolds: -1

randomSeed: 1

toleranceParameter: 0.001

[Open...] [Save...] [OK] [Cancel]

**Classifier output:**

Time taken to build model: 0.0

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances  
Incorrectly Classified Instance  
Kappa statistic  
Mean absolute error  
Root mean squared error  
Relative absolute error  
Root relative squared error  
Coverage of cases (0.95 level)  
Mean rel. region size (0.95 level)  
Total Number of Instances

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate
	1.000	0.000
	0.980	0.050
	0.900	0.010
Weighted Avg.	0.960	0.020

=== Confusion Matrix ===

a	b	c	<-- classified as
50	0	0	a = Iris-setosa
0	49	1	b = Iris-versicolor
0	5	45	c = Iris-virginica

Log x 0