

パターン認識特論

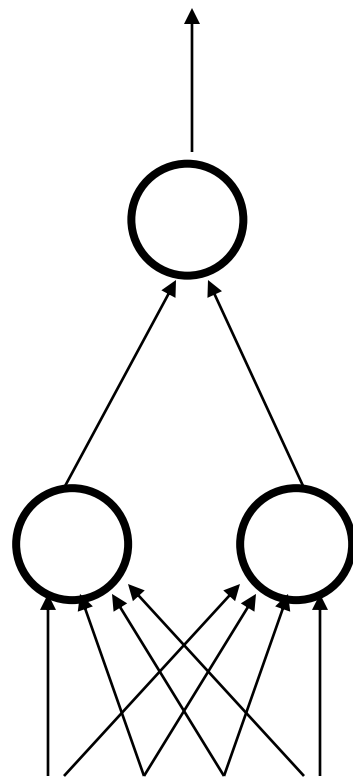
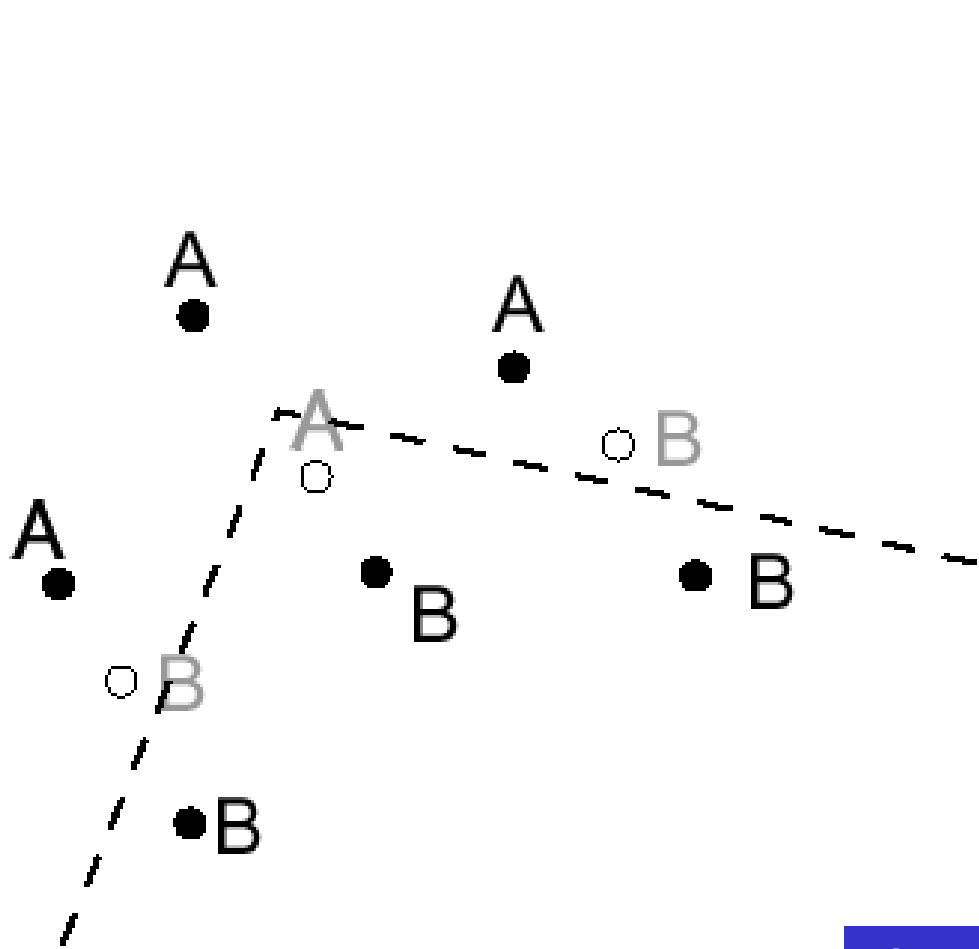
VC dimension

Support Vector Machines

Kernel Method

汎化と分化の制御

中間(隠れ層)素子数の設定



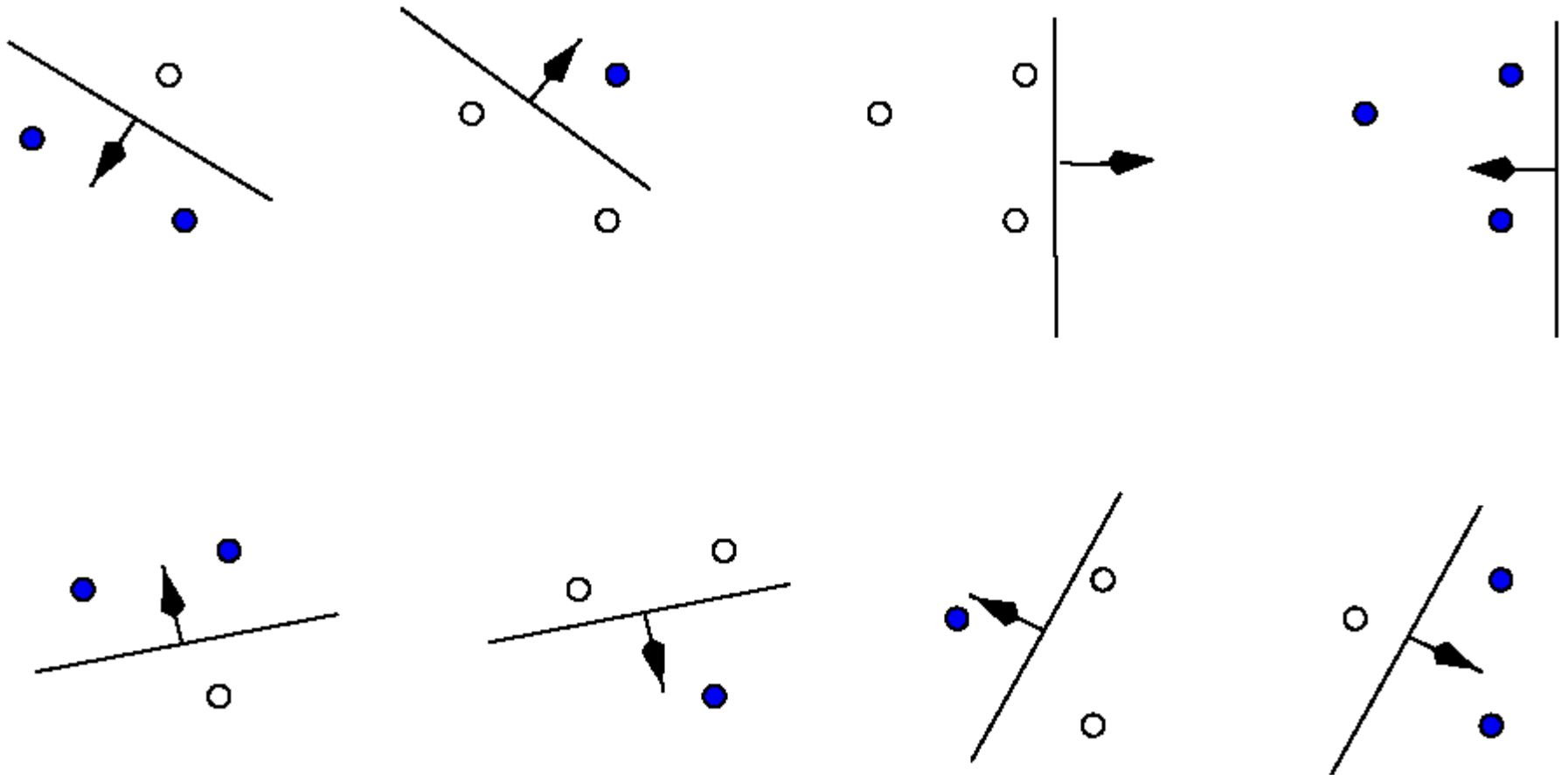
中間層の素子数を増やせば分化が進み、減らせば汎化が進む

VC dimension

- ある空間で、ある関数形で表現可能な識別面を設定し、特徴ベクトルを2クラスに分類する問題を考える。
- h 個の特徴ベクトルに対して任意のラベリングをしても、識別面のパラメータをうまく選べば、これらの特徴点分離可能になるとする。このような特徴点の最大個数 h を、この空間中でのこの識別関数のVC (Vapnik-Chervonenkis) dimensionと呼ぶ。
- 注意すべきことは、上述のような h 個の特徴ベクトルが存在するということであり、任意の特徴ベクトルの配置に対して上記が成立する必要は無いということである。

VC dimension の例

2次元平面上での線形識別面 ($h=3$)



VC dimension の性質

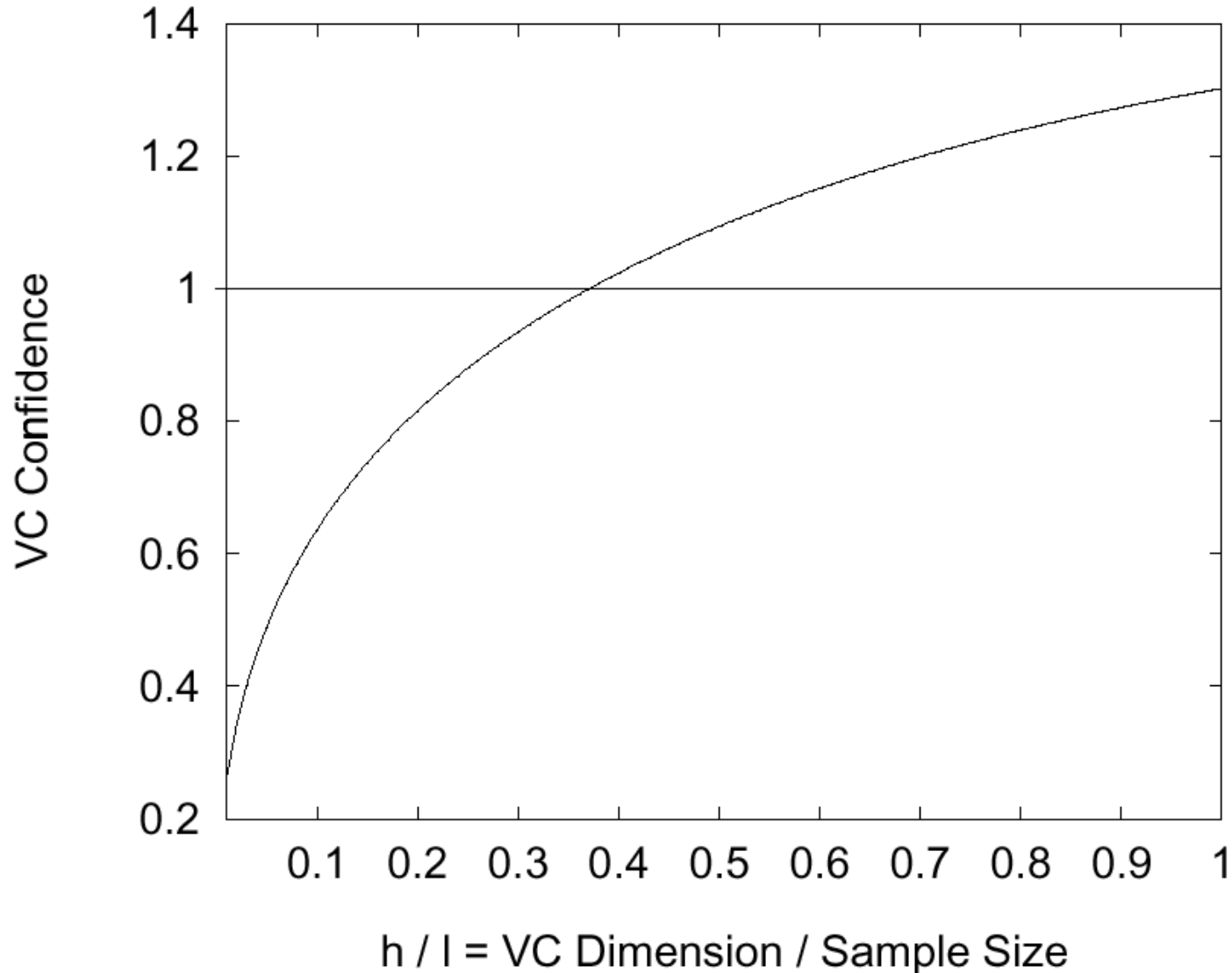
- 期待リスク $R(\alpha) = \int \frac{1}{2} |y - f(\mathbf{x}, \alpha)| dP(\mathbf{x}, y)$
- 実リスク $R_{emp}(\alpha) = \frac{1}{2l} \sum_{i=1}^l |y_i - f(\mathbf{x}_i, \alpha)|$
- 期待リスクの上限

$$R(\alpha) \leq R_{emp}(\alpha) + \underbrace{\sqrt{\left(\frac{h(\log(2l/h) + 1) - \log(\eta/4)}{l} \right)}}_{\text{VC Confidence}}$$

y : 教師信号、 $f(\cdot)$: 識別関数、 h : VC次元、 l : サンプル数、 $1-\eta$: 上限が正しい確率

VC Confidence の性質

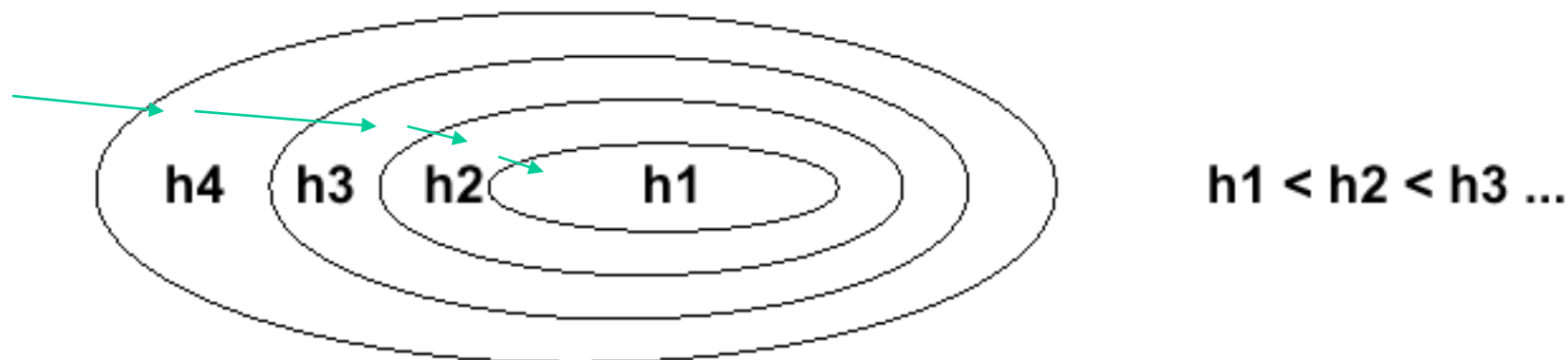
$\eta=0.05$: 上限が正しい確率=0.95の場合



SRM: Structural Risk Management

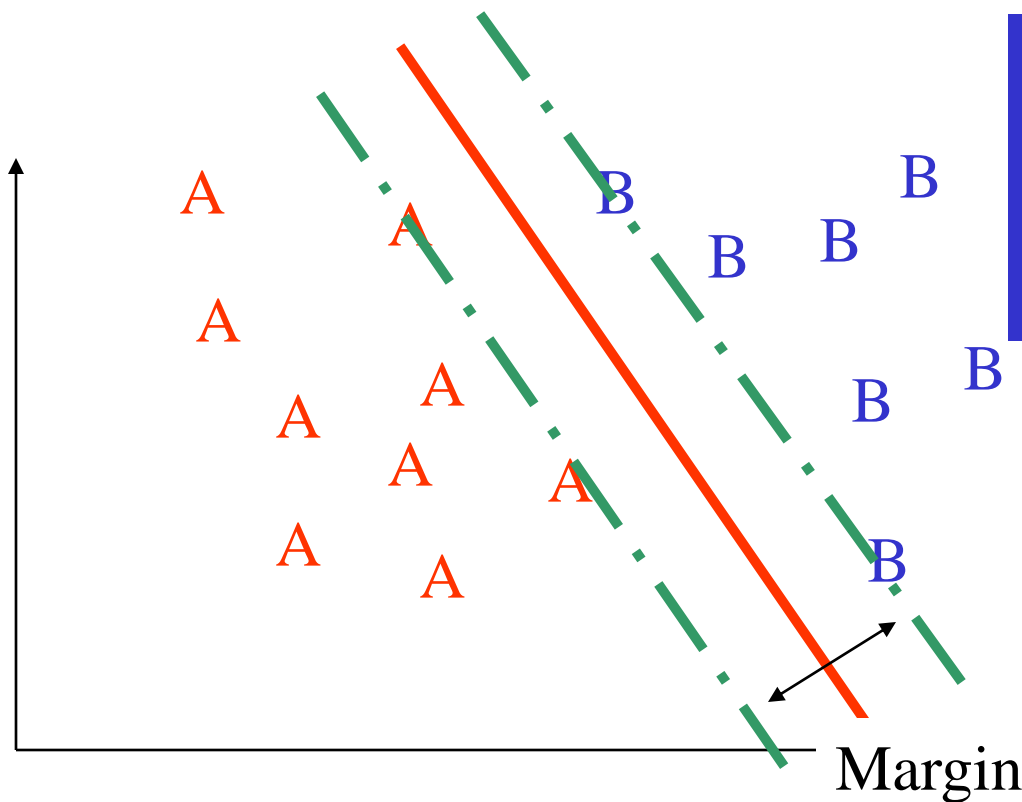
$R_{emp}(\alpha)$ が小さく、且つ、VC confidence が小さいとき、リスクの上限が低く押さえられる。

- ある識別関数の集合を、VC dimension の値に応じて部分集合に分けておく。そして、各集合の中でそれぞれ $R_{emp}(\alpha)$ の最小化を行う。そして、 $R_{emp}(\alpha) + VC \text{ confidence}$ が一番小さいものを選べば、その識別関数の集合の中で最適な識別機械が構成できる。



Linear Support Vector Machine: Separable Case

- どんな識別面が「良い」識別面か？



線形識別面から、最近傍のサンプル点までの距離が最大になるように識別面を設定すればよい。

Linear Support Vector Machine: Separable Case (問題)

識別面の方程式

$$x \cdot w + b = 0$$

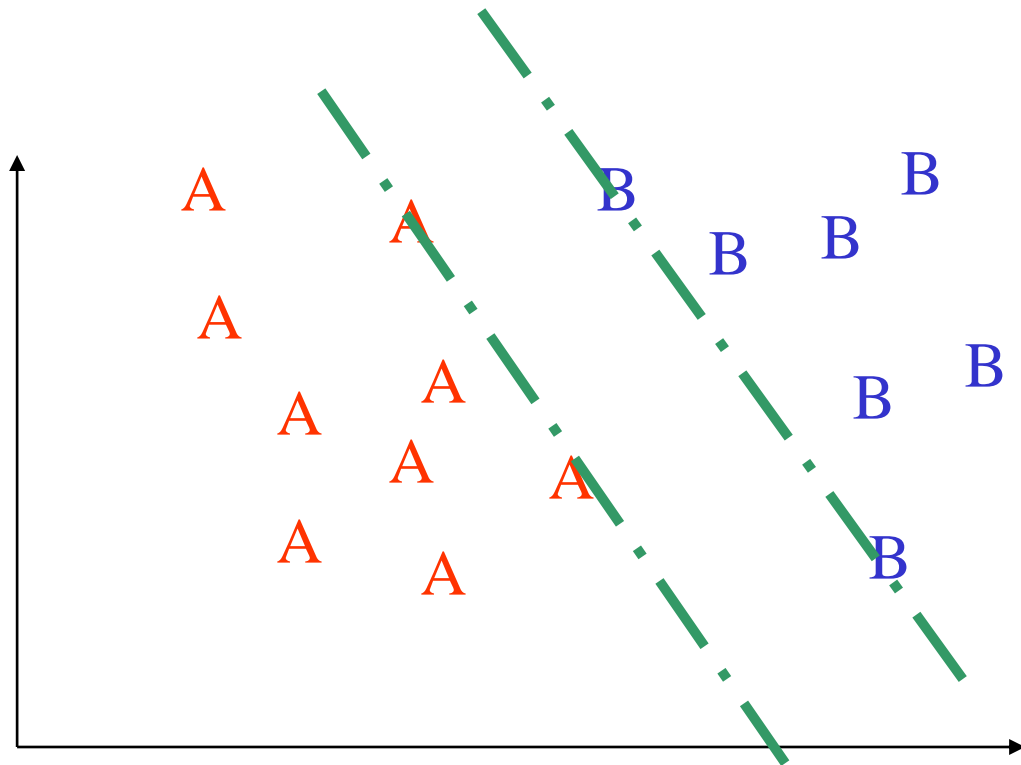
とし、教師信号 t を ± 1 とする。また、Bの点が $x \cdot w + b \geq 1$ 、
Aの点が $x \cdot w + b \leq -1$ を満足するとする。

条件1

このとき、各サンプル (x_i, t_i) は、

$$t_i(x_i \cdot w + b) - 1 \geq 0$$

を満足する。

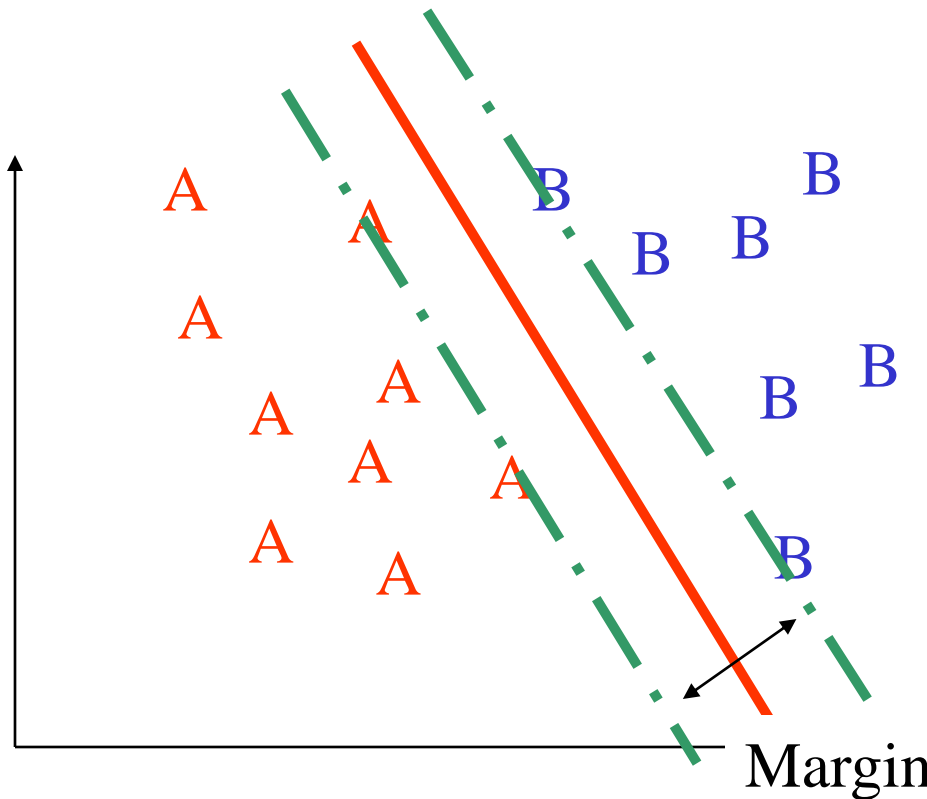


Linear Support Vector Machine: Separable Case (何をすべきか)

このとき、識別面に最も近いB、Aの点はそれぞれ $x \cdot w + b = 1$, $x \cdot w + b = -1$ 上に存在する。原点からこれらの超平面までの距離は、それぞれ

$|1-b| / \|w\|$, $|-1-b| / \|w\|$ なので2平面間の距離(Margin)は、 $2/\|w\|$ となる。すなわち、 $\|w\|$ が小さくなるほど、Marginが大きくなる。

したがって、前述の条件1の下で $\|w\|$ を最小化すれば良い。



Linear Support Vector Machine: Separable Case (定式化と解法)

この問題は、Lagrange の未定係数法により、以下の関数を最小化する問題に帰着する(後述のKuhn-Tucker条件を使うことが前提)。

$$L_p = 1/2 \|\mathbf{w}\|^2 - \sum_i \alpha_i (t_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1)$$

右辺第1項はMarginを最大化する項、第2項はすべてのデータが正しく識別されるための項である。このとき、 $\partial L_p / \partial \mathbf{w}$, $\partial L_p / \partial b$ ともに0になるという条件から式の簡単化が行え、結果的に上の式は

$$\sum_i \alpha_i - 1/2 \sum_{i,j} \alpha_i \alpha_j t_i t_j \mathbf{x}_i \cdot \mathbf{x}_j$$

となる。これを $\partial L_p / \partial \mathbf{w} = 0$, $\partial L_p / \partial b = 0$, $t_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 \geq 0$,

$\alpha_i \geq 0$, $\alpha_i (t_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1) = 0$ という条件 (Kuhn-Tucker条件)

のもとで最小化すれば良い。(これらの等式不等式制約は、ある正則化条件の下で全て線形化することができ、結果的に2次計画問題になる。この問題は、内点法などを用いて解くことができる。)

Linear Support Vector Machine: Separable Case (何が得られるか)

明らかに $\alpha_i \neq 0$ となる x_i が識別面に最も近い点となり、この点では、 $t_i(x_i \cdot w + b) - 1 = 0$ となる。このような点を **サポートベクター** と呼ぶ。

最適化後には、

$$\sum_i \alpha_i (t_i(x_i \cdot w + b) - 1) = 0$$

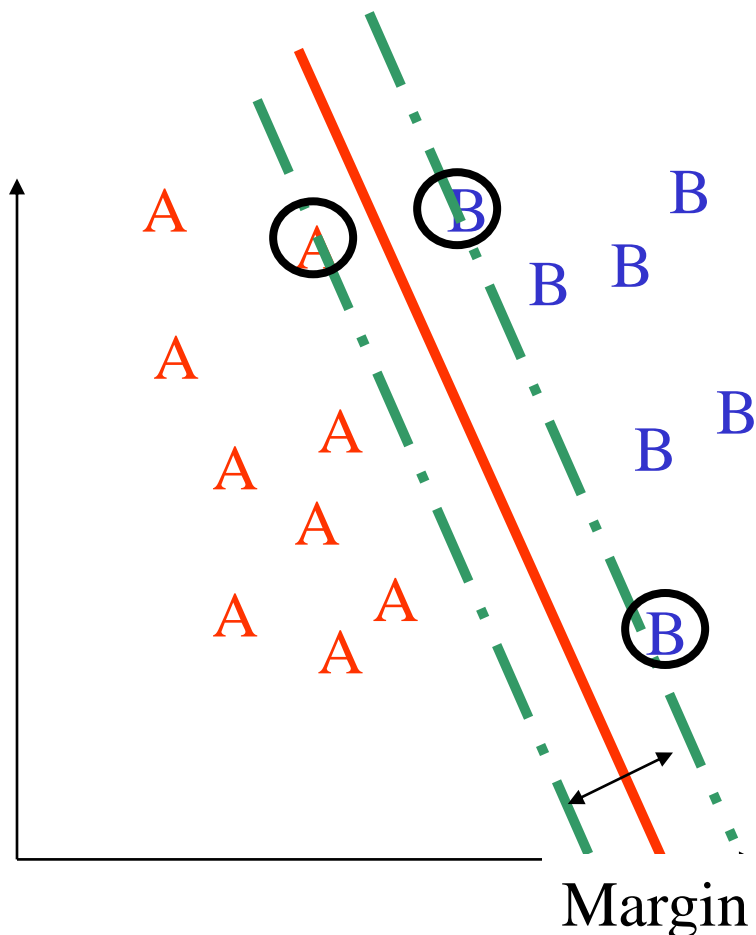
となり、Support Vector以外では $\alpha_i = 0$ 。また、最適な b は、A, B の Support Vector のうち任意の2つ x_A, x_B および最適化された w から、

$$b = -1/2 w \cdot (x_A + x_B)$$

と計算される。Support Vectorの標識集合を S_v とすると、最適識別面は次式で表される。

$$\sum_{i \in S_v} t_i \alpha_i (x_i \cdot (x - 1/2(x_A + x_B))) = 0$$

また、 $w = \sum_{i \in S_v} t_i \alpha_i x_i$ となる。



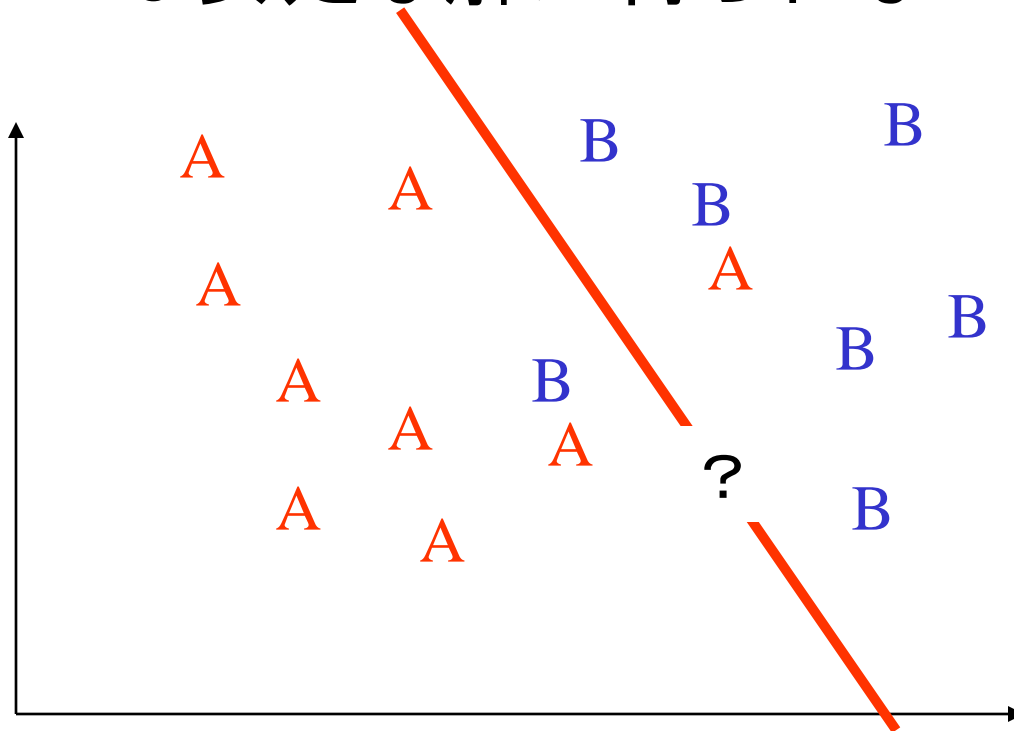
实例

<http://svm.cs.rhul.ac.uk/pagesnew/GPat.shtml>

Linear Support Vector Machine:

Non-Separable Case (どうすれば良いか)

- 線形識別可能でないとき、前述の方法では安定な解が得られない。



学習サンプルを用いたときのエラーを許容する定式化を行う。Soft Marginの導入

Linear Support Vector Machine:

Non-Separable Case (何を最適化するか)

識別面の方程式 $x \cdot w + b = 0$ 、教師信号 t を ± 1 とする。また、Bの点が $x_i \cdot w + b \geq 1 - \xi_i$ 、Aの点が $x_i \cdot w + b \leq -1 + \xi_i$ を満足するとする。 ξ_i は、スラック変数と呼ばれる0以上の変数。

このとき、各サンプル (x_i, t_i) は、

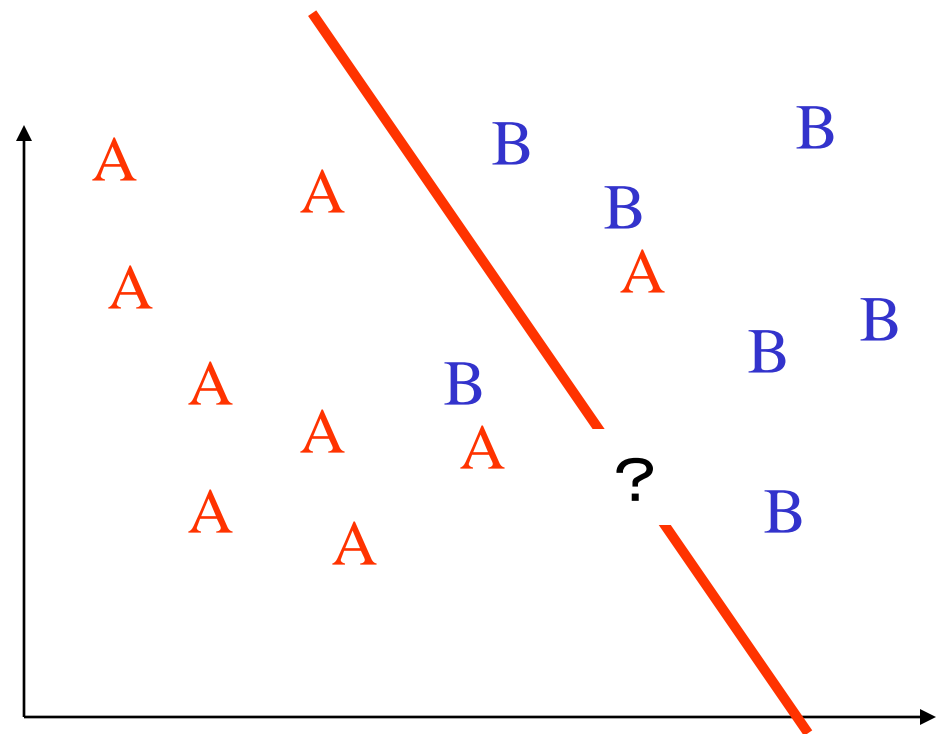
$$t_i(x_i \cdot w + b) - 1 + \xi_i \geq 0$$

を満足する。また、 $\sum_i \xi_i$ は、

トレーニング時のエラーの総和を表している。したがって、最小化の対象を

$$\frac{1}{2} \|w\|^2 + C \sum_i \xi_i$$

線形分離不可能な場合にも対処でき、トレーニング時のエラーも最小化できる。



Linear Support Vector Machine: Non-Separable Case (定式化と解法)

この問題は、

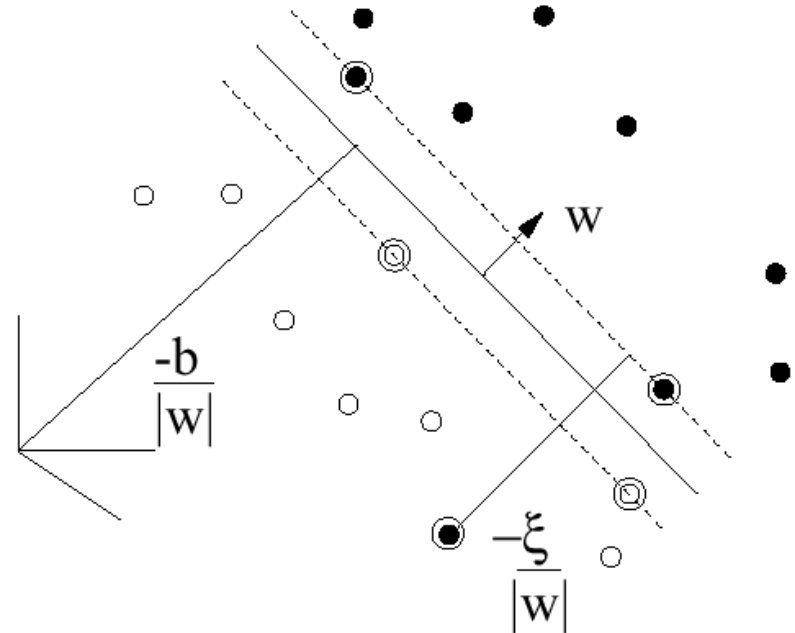
$$L_p = 1/2 \|\mathbf{w}\|^2 + C \sum_i \xi_i - \sum_i \alpha_i (t_i (\mathbf{x}_i \cdot \mathbf{w} + b) - 1 + \xi_i) - \sum_i \mu_i \xi_i$$

となり、式変形により先の問題と同様に、

$$\sum_i \alpha_i - 1/2 \sum_{i,j} \alpha_i \alpha_j t_i t_j \mathbf{x}_i \cdot \mathbf{x}_j$$

を最大化する問題となる。この場合, Kuhn-Tucker条件は異なるが、同様に内点法などで解くことができる。

この場合のサポートベクターは $\alpha_i \neq 0$ かつ $\xi_i = 0$ となる \mathbf{x}_i であり、 $t_i (\mathbf{x}_i \cdot \mathbf{w} + b) - 1 = 0$ を満足する。
 $\xi_i \neq 0$ となる点では、誤識別が起きている。



实例

<http://svm.cs.rhul.ac.uk/pagesnew/GPat.shtml>

Non-Linear Support Vector

Machine: Kernel Method (アイデア)

識別面が線形であっても入力 x を非線形変換 $\varphi(x)$ で次元を拡張すれば、元の空間で非線形識別面を構成したのと同じことになる。

例:

入力 x を (x, x^2) に変換して、その空間で線形識別面を構成することはもとの空間で2次式で表される識別面を構成したと等価である。

疑問点: 計算が複雑になるのではないか? 次元が上がると、膨大なメモリが必要ではないか?

答え: 次元拡張しても、内積の形でしか $\varphi(x)$ を使わないのであれば、ほとんど同様の方法で、SVMが構成できる。このときに使う $\varphi(x)$ の内積のことをkernelと呼ぶ。

$$K(x, y) = \varphi(x) \cdot \varphi(y)$$

Non-Linear Support Vector Machine: Kernel Method を用いたSVM

$$\sum_i \alpha_i - 1/2 \sum_{i,j} \alpha_i \alpha_j t_i t_j \mathbf{x}_i \cdot \mathbf{x}_j$$

を最大化する問題が、

$$\sum_i \alpha_i - 1/2 \sum_{i,j} \alpha_i \alpha_j t_i t_j K(\mathbf{x}_i, \mathbf{x}_j)$$

を最大化する問題になるだけで、本質的な問題の構造に影響を与えない。このときの最適識別面は、

$$\sum_{i \in S_V} t_i \alpha_i (K(\mathbf{x}_i, \mathbf{x}) - 1/2 (K(\mathbf{x}_i, \mathbf{x}_A) + K(\mathbf{x}_i, \mathbf{x}_B))) = 0$$

となる。

Non-Linear Support Vector Machine: Kernel の拡張

Kernel は、明示的な内積形式でなくてもよい。

$H_{ij} = t_i t_j K(x_i, x_j)$ とすれば、

$$\sum_i \alpha_i - 1/2 \sum_{i,j} \alpha_i \alpha_j t_i t_j K(x_i, x_j) = \sum_i \alpha_i - 1/2 \sum_{i,j} \alpha_i H_{ij} \alpha_j$$

となる。この式の最大化が行える条件は、行列 $[H_{ij}]$ の固有値が全て正であることである。

このために $K(x, y)$ は Mercer の条件と呼ばれる次式を満足するように選ばれることが多い。

$$K(x, y) = \sum_{p=0}^{\infty} C_p (x \cdot y)^p \text{ かつ、} C_p \text{ が一様収束。}$$

この条件を満足するものとしては、次のようなものがある。

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + 1)^p \quad \text{および、} (x \cdot y)^p \quad \text{多項式カーネル}$$

$$K(\mathbf{x}, \mathbf{y}) = e^{-\|\mathbf{x} - \mathbf{y}\|^2 / 2\sigma^2} \quad \text{Radial Basis Function (RBF)}$$

$$K(\mathbf{x}, \mathbf{y}) = \tanh(\kappa \mathbf{x} \cdot \mathbf{y} - \delta) \quad \kappa \text{ と } \delta \text{ の選び方によっては、条件を満足しない}$$

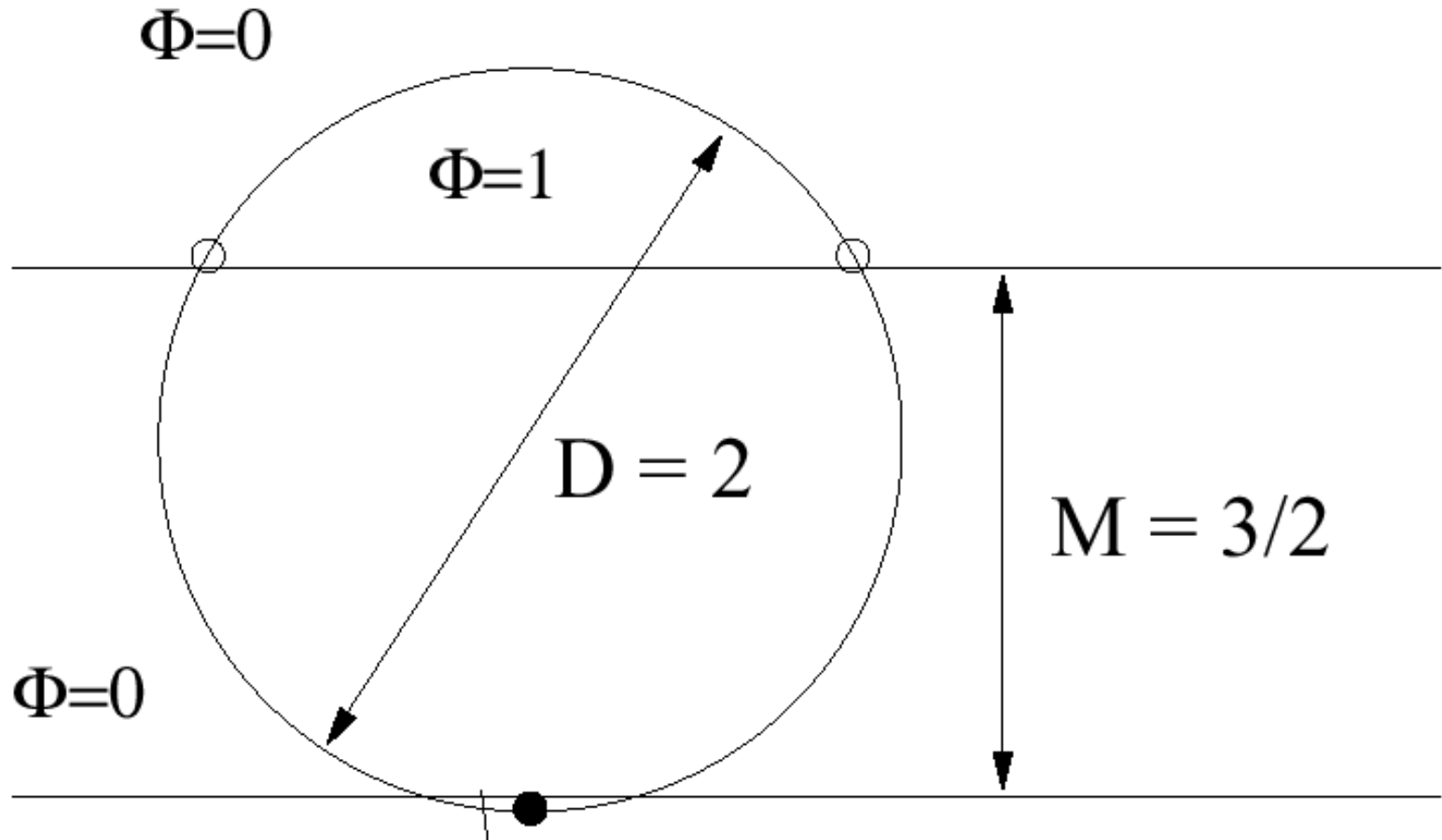
实例

<http://svm.cs.rhul.ac.uk/pagesnew/GPat.shtml>

SVMの性質(準備)

Gap Tolerant Classifiers

- 球内の点を shutter する問題。Gapの中と円の外の点は無視する。以下は2次元平面上の直径2の場合、 $M \leq 3/2, h=3$, $3/2 < M < 2, h=2$, $2 \leq M, h=2$



SVMの性質(準備)

Gap Tolerant Classifiers

- n 次元空間内の超球を用いた場合の結果。

$$h \leq \min\{\lceil D^2/M^2 \rceil, n\} + 1$$

Gap Tolerant Classifier (GTC) を用いた Structural Risk Minimization.: トレーニングを行い、直径を変えて複数のGTCを得る。そして、そのVC dimension とトレーニングエラーからリスクの上限を計算し、最適なGTCを決定する。

得られたGTCは、超球の外部を見ないという点だけがSVMと異なる。

- SVMの場合、VC dimension は、データに依存して変化するため、一概には決められないが、上述の上限以下であることは示されている。(実際の実験例では、理論上界の約1/100以下のリスク値が得られている。)

SVMの性質(1)

- マージンとリスクの関係: GTCの議論から, マージンが大きいほどVC次元が下がると言える。特殊な例として、以下の定理が示されている。「サンプル数 l でトレーニングされた原点を通過する最適識別面のエラーの期待値は、次式で与えられる。」

$$E[P(error)] \leq \frac{E[D^2/M^2]}{l}$$

また、期待リスク R は、 $R_{\text{emp}} + \lambda/2 \|w\|^2$ とも表される。

- サポートベクターの個数とリスクの関係 (線形分離可能な場合)

$$E[P(error)] \leq \frac{E[\text{Number of support vectors}]}{\text{Number of training samples}}$$

サポートベクターを除去して学習した識別面に、サポートベクターをテストデータとして与えたとき、誤りが生じ得るため。

SVMのTutorial

- C. J. C. Burges. [A Tutorial on Support Vector Machines for Pattern Recognition](#). *Knowledge Discovery and Data Mining*, 2(2), 1998. A detailed tutorial on Support Vector machines for the classification task, from background material (e.g. VC dimension, structural risk minimization) through notes on training algorithms. Many examples are given.
- A. J. Smola and B. Schölkopf. [A Tutorial on Support Vector Regression](#). NeuroCOLT Technical Report NC-TR-98-030, Royal Holloway College, University of London, UK, 1998. This tutorial gives an overview of the basic ideas underlying Support Vector (SV) machines for regression and function estimation. Furthermore, it includes a summary of currently used algorithms for training SV machines, covering both the quadratic (or convex) programming part and advanced methods for dealing with large datasets. Finally, some modifications and extensions that have been applied to the standard SV algorithm are mentioned, and it discusses the aspect of regularization and capacity control from a SV point of view.
- B. Schölkopf. [Support vector learning](#), 1999. Tutorial given at DAGM'99.

SVMのプログラム

- [Nearest Point Algorithm](#). by [Sathiya Keerthi](#) (in FORTRAN).
- [SVM Java Applet](#). by [Chris Burges et al.](#)
- [BSVM](#). A decomposition method for bound-constrained SVM formulations.
- [QP SVM Classification and Regression](#). Fortran Implementation.
- [Chunking Code](#). by C. Saunders, M. O. Stitson, J. Weston, L. Bottou, B. Schölkopf, and A. Smola at Royal Holloway, AT&T, and GMD FIRST ([Documentation](#)).
- [Interior Point Optimizer for SVM Pattern Recognition](#). by [Alex Smola](#).
- [LEARNSC](#). SVM, NN and FL MATLAB based user-friendly routines.
- [LIBSVM](#). [An SVM library with a graphic interface](#).
- [looms](#). a leave-one-out model selection software based on BSVM.
- [mySVM](#). SVM implementation for pattern recognition and regression.
- [mySVM and SVMlight for Windows](#). SVM implementation for Windows, uses Microsoft Visual C++ 6.0.
- [Sequential Minimal Optimization](#). by [Xianping Ge](#).
- [SMOBR](#). [SMOBR is an implementation of the original Sequential Minimal Optimisation proposed by Platt written in C++](#).
- [SvmFu](#). by [Ryan Rifkin](#).
- [SVMLight](#). by [Thorsten Joachims](#).
- [SVM/LOO](#). Matlab code for SVM incremental learning and decremental unlearning (LOO validation).
- [SVM/optima](#). SVM QP routines in Fortran for classification/regression.
- [SVM Torch](#). Support Vector Machine for Large-Scale Regression and Classification Problems.
- [Matlab SVM Toolbox](#). by [Steve Gunn](#).
- [Matlab SVM Toolbox](#). Matlab implementation in the style of SVMlight, can train 1-norm and 2-norm SVMs.
- [OSU SVM Classifier Matlab Toolbox](#). A matlab toolbox with a C++ mex core to fast implement the SVM classifiers.
- [SVM Toolbox](#). Object Oriented MATLAB Support Vector Machine Toolbox, including C++ MEX implementation of the sequential minimal optimisation algorithm.

来週の講義は休講なので、
下記のレポートを課します。

- レポート課題：

C-SVMに対して、 ν -SVMという手法が存在するが、そのメリットとC-SVMに対する関係について説明しなさい。

- 提出日：12月17日講義開始前