サポートベクターマシン

赤穂昭太郎
産業技術総合研究所

2006.7.6～7 統数研公開講座
「カーネル法の最前線—SVM, 非線形データ解析, 構造化データ—」

目次

1. サポートベクターマシン(SVM)とは
2. SVMの仲間たち
3. 汎化能力の評価
4. 学習アルゴリズム
5. カーネル設計の基本

1.サポートベクターマシンとは？

- サポートベクターマシン＝
  「線形モデル＋高次元写像＋正則化」の一つ
  （ほかのカーネル法と基本的に共通）

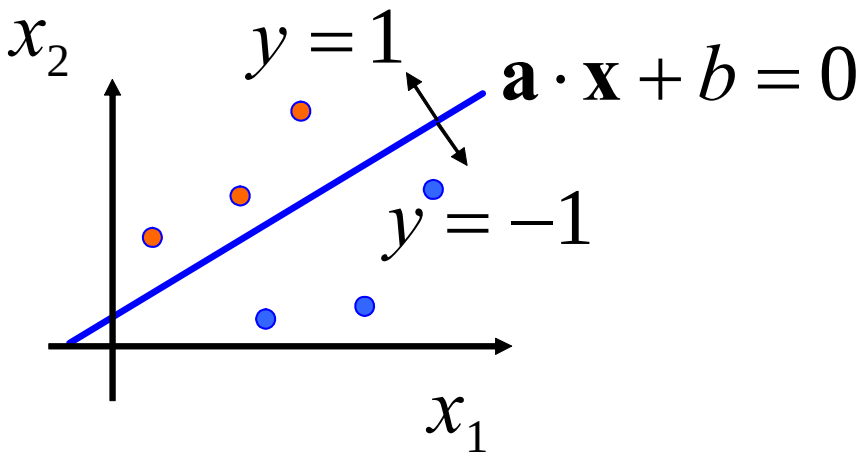
- 計算量を少なくするいくつかの工夫
  - 凸2次計画法
  - スパース表現
  - カーネルトリック

- ローカルミニマムの問題がない

識別問題

- 入力 x を二つのクラスに分類する関数 $y = f(\mathbf{x})$ を学習する（1クラスあるいは3クラス以上の場合は後で考える）
  例：文字認識、遺伝子解析、データマイニング、spam filter
- 学習サンプルから学習 $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m) \in R^n \times \{\pm 1\}$
- 線形識別器：

$$y = \mathrm{sgn}[\mathbf{a} \cdot \mathbf{x} + b]$$

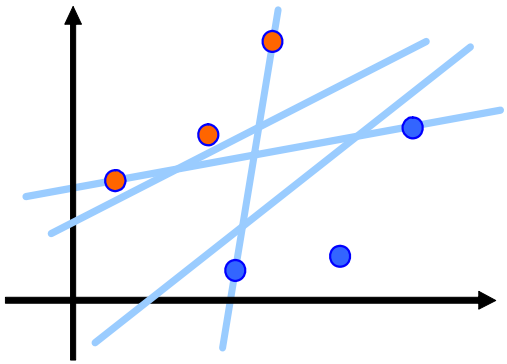


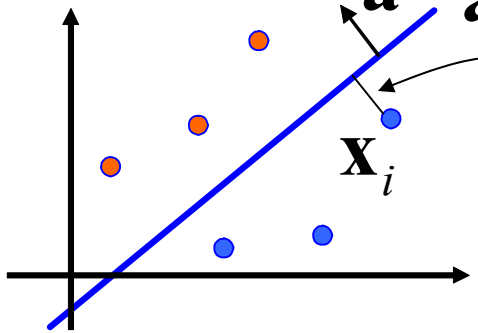
$$\mathbf{a} \cdot \mathbf{x} = a_1 x_1 + a_2 x_2 + \cdots + a_n x_n$$

$$\mathrm{sgn}[u] = \begin{cases} 1 & \cdots u \geq 0 \\ -1 & \cdots u < 0 \end{cases}$$

マージン最大化(1/2)

- 正しい識別を行う超平面はたくさんある
- 真ん中ぐらいを通るのがいいだろう





超平面への(向きあり)距離

$$d_i = -\frac{\mathbf{a}\cdot\mathbf{x}_i + b}{\|\mathbf{a}\|}$$

$$\mathbf{a}\cdot\underbrace{\left(\mathbf{x}_i + d_i\frac{\mathbf{a}}{\|\mathbf{a}\|}\right)}_{\text{垂線の足}} + b = 0$$

マージン最大化

$$\max_{\mathbf{a},b}\min_{i}\frac{|\mathbf{a}\cdot\mathbf{x}_i + b|}{\|\mathbf{a}\|}$$

このままでは複雑なので簡単化する

マージン最大化(2/2)

- とりあえず線形分離可能性（サンプルを100%分類できる超平面の存在）を仮定
- 超平面 $\mathbf{a}\cdot\mathbf{x}+b=0$ は定数倍しても不変なのですべてのサンプルに対し，

$$\begin{cases} \mathbf{a}\cdot\mathbf{x}_i+b\geq 1 & \cdots y_i=1 \\ \mathbf{a}\cdot\mathbf{x}_i+b\leq -1 & \cdots y_i=-1 \end{cases}$$ と仮定してよい

一行で書けば $y_i(\mathbf{a}\cdot\mathbf{x}_i+b)\geq 1$

- このとき、マージン $=\min_i \dfrac{|\mathbf{a}\cdot\mathbf{x}_i+b|}{\|\mathbf{a}\|}=\dfrac{1}{\|\mathbf{a}\|}\rightarrow \max$

$$\Updownarrow$$

$$\|\mathbf{a}\|^2\rightarrow \min$$

凸2次計画問題（ハードマージン・主問題）

解くべき問題（ハードマージン）

$$\min_{\mathbf{a},b} \|\mathbf{a}\|^2$$

subject to $y_i(\mathbf{a}\cdot\mathbf{x}_i + b) \geq 1 \quad (i = 1,\cdots,m)$

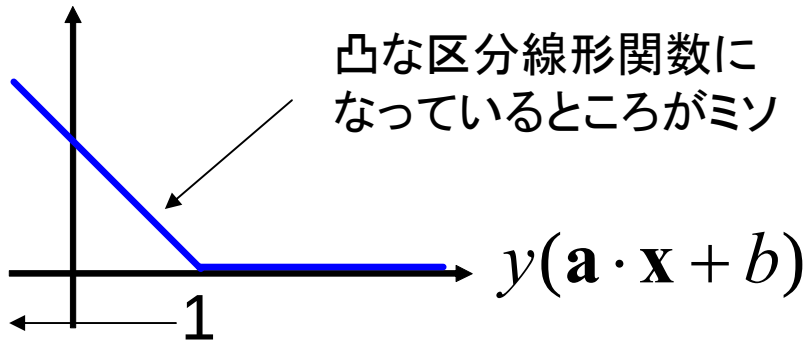
- 線形制約下での凸な2次関数の最適化
- ローカルミニマムの問題がない
  （局所最適＝大域的最適）
- 計算パッケージなどを用いれば比較的容易に解ける
- このままでも解くことはできるが，双対問題に変換すると，制約の部分がより簡単な式で表される

ソフトマージン

- 先に進む前に，線形分離可能ではない場合の対処法を導入
- 制約を満たさないサンプルは絶対に許さない（ハードマージン）ではなく，許すがそのかわり罰金を科す（ソフトマージン）

$$r(\mathbf{x}, y) = \left(1 - y(\mathbf{a}\cdot\mathbf{x} + b)\right)_+$$

$$(x)_+ = \begin{cases} x & (x \geq 0) \\ 0 & (x < 0) \end{cases}$$



- 線形分離な場合でもノイズがある場合などは無理に線形分離な超平面を求めないほうがよい．

凸2次計画問題(ソフトマージン・主問題)

ソフトマージンを用いた最適化問題

$$\frac{1}{2}\|\mathbf{a}\|^2 + C\sum_{i=1}^{m}\left(1 - y_i(\mathbf{a}\cdot\mathbf{x}_i + b)\right)_+ \rightarrow \min_{\mathbf{a},b}$$

C : マージンを破るコストの重み
ハードマージンではC=∞

スラック変数を導入して$(x)_+$を消す

$$\frac{1}{2}\|\mathbf{a}\|^2 + C\sum_{i=1}^{m}\xi_i \rightarrow \min_{\mathbf{a},b,\xi_1,\ldots,\xi_n}$$

subject to $\xi_i \geq 1 - y_i(\mathbf{a}\cdot\mathbf{x}_i + b)$

$\xi_i \geq 0 \qquad (i = 1,\cdots,m)$

双対問題の導出

- 制約付き最適化→Lagrange の未定係数法

$$L = \frac{1}{2}\|\mathbf{a}\|^2 + C\sum_i \xi_i - \sum_i \alpha_i(\xi_i - 1 + y_i(\mathbf{a}\cdot\mathbf{x}_i + b)) - \sum_i \beta_i \xi_i$$

$\alpha_i, \beta_i \geq 0$ ：Lagrange の未定係数

- $\mathbf{a}$ で微分＝0 $\Longrightarrow \mathbf{a} = \sum_i \alpha_i y_i \mathbf{x}_i$
- b で微分＝0 $\Longrightarrow \sum_i y_i \alpha_i = 0$
- ξ_i で微分＝0 $\Longrightarrow \alpha_i + \beta_i = C \xrightarrow[\beta_i \geq 0]{} \alpha_i \leq C$
- L に代入すれば双対問題(Wolfe dual)が得られる

$$\max_{\alpha_1,\ldots,\alpha_n} \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$$

$$\text{subject to } 0 \leq \alpha_i \leq C, \sum_i y_i \alpha_i = 0$$

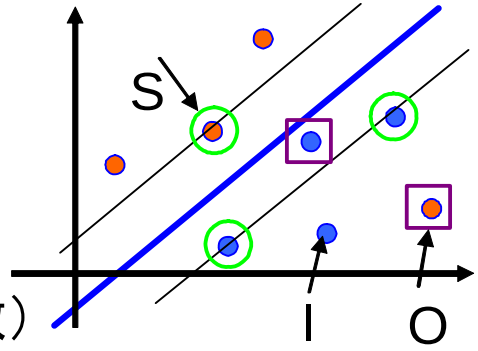
双対問題の性質

- $\mathbf{a} = \sum_i \alpha_i y_i \mathbf{x}_i$: サンプル点の線形和 (cf. Representer Theorem)
- Karush-Kuhn-Tucker 条件
  不等式を等号で満たす場合のみLagrangeの未定係数≠0

$$\alpha_i = 0 \iff \xi_i - 1 + y_i(\mathbf{a}\cdot\mathbf{x}_i + b) > 0$$
$$\beta_i = 0 \iff \xi_i > 0$$

  - S: マージン上にある $\xi_i = 0, -1 + y_i(\mathbf{a}\cdot\mathbf{x}_i + b) = 0 \Rightarrow 0 < \alpha_i < C$
  - I: マージン内にある $\xi_i = 0, -1 + y_i(\mathbf{a}\cdot\mathbf{x}_i + b) > 0 \Rightarrow \alpha_i = 0$
  - O: マージンを超える $\xi_i > 0, -1 + y_i(\mathbf{a}\cdot\mathbf{x}_i + b) < 0 \Rightarrow \alpha_i = C$

$$\mathbf{a} = \sum_{i\in S} \alpha_i y_i \mathbf{x}_i + C\sum_{i\in O} y_i \mathbf{x}_i$$



- $\alpha_i \neq 0$ を満たすベクトルを**サポートベクター**という
  サンプル点のうちサポートベクターだけの関数
  に帰着された(SVMの語源？: サポート＝支持関数)

しきい値bについて

- マージン上の任意のベクトル $j \in S$ について

$$-1 + y_i(\mathbf{a}\cdot\mathbf{x}_i + b) = 0$$

$$\Longrightarrow \quad b = y_i - \mathbf{a}\cdot\mathbf{x}_i = y_i - \sum_{j\in S}\alpha_j\mathbf{x}_j\cdot\mathbf{x}_i - C\sum_{j\in O}\mathbf{x}_j\cdot\mathbf{x}_i$$

- もともとしきい値を b=0 とすることも考えられる（原点を通る超平面）.
- すると最適化問題の中の $\sum_i y_i\alpha_i = 0$ の条件は不要になる.
- 後で述べるカーネルを用いた表現では，多くの場合暗に基底 1 が含まれている

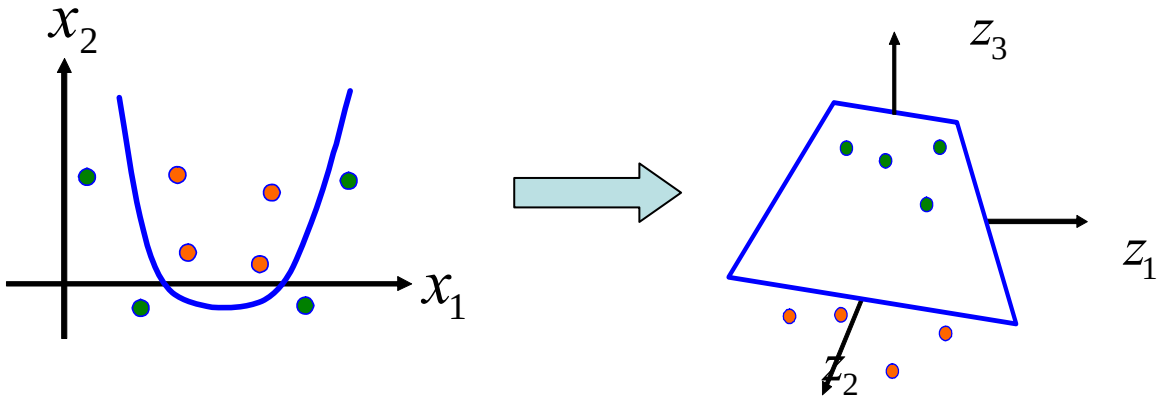
$$\tilde{\mathbf{a}} = (\mathbf{a}^T, b)^T \qquad \tilde{\mathbf{x}} = (\mathbf{x}^T, 1)^T$$

$$\mathbf{a}\cdot\mathbf{x} + b \quad \Longrightarrow \quad \tilde{\mathbf{a}}\cdot\tilde{\mathbf{x}}$$

- $\tilde{\mathbf{x}}$ の空間でマージンを考えていないので少し結果は異なる

高次元空間への写像

- 線形で能力不足のとき，
  - 非線形化（ニューラルネットなど）：
    学習に時間がかかる，ローカルミニマムの問題など
  - 高次元に写像して線形（SVMなどはこちら）



$$(x_1, x_2) \mapsto (z_1, z_2, z_3) = (x_1, {x_1}^2, x_2)$$

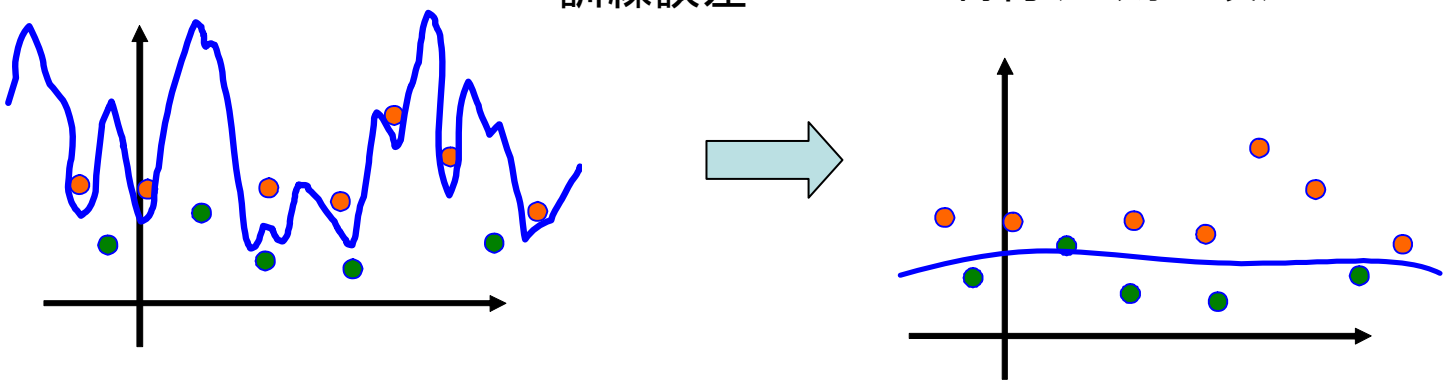
$$x_2 = a_1 x_1 + a_2 {x_1}^2 + b = a_1 z_1 + a_2 z_2 + b$$

次元の呪いと正則化

- サンプル数と同じ次元まで上げればどんなものでも線形識別可能となる
- オーバーフィットして，サンプルを増やしても真の解に近づかない（不良設定問題）
- なめらかさなどの制約をおいて対処する（**正則化**）

$$\min_f \sum_i R(y_i, f(\mathbf{x}_i)) + \lambda \Omega(f)$$

訓練誤差　　制約（正則化項）



SVMと正則化

- SVMは正則化とみなせる

$$\min_f \sum_i r(y_i, f(x_i)) + \lambda\Omega(f)$$

$$\min_{\mathbf{a},b} \sum_{i=1}^{m} \left(1 - y_i(\mathbf{a}\cdot\mathbf{x}_i + b)\right)_+ + \frac{1}{2C}\|\mathbf{a}\|^2$$

- 参考：正則化≒MAP推定

$$p(x, y \mid f) = p(y \mid x, f)q(x) = q(x)\exp(-r(y, f(x)))/Z(f(x))$$

$$p(f) \propto \exp(-\lambda\Omega(f))$$

MAP $\min_f \sum_i \{r(y_i, f(x_i)) - \log Z(f(x_i))\} + \lambda\Omega(f)$

正則化
・規格化因子のない分布モデル（cf.boosting など）
・対数尤度のロバスト化 (cf.M-estimatorなど)

カーネルによる高次元化

- SVMによって得られる識別関数は $y = \mathrm{sgn}\left[\sum_i \alpha_i \mathbf{x}_i \cdot \mathbf{x} + b\right]$ になる。
  内積の形になっていることに注意。
  （最適化関数等もすべて $\mathbf{x}$ に関しては内積の形）
- 高次元化: $\mathbf{x}$ を $\mathbf{z}(\mathbf{x})$ に置き換える。
- 高次元化の問題点:
  高次元への写像の計算が大変.
  素朴にやると次元分の計算量がかかる
- 再生核ヒルベルト空間への写像なら上げる次元よりはるかに低い計算量で計算可能

$$\mathbf{x} \mapsto \mathbf{z}(\mathbf{x}) = k(\mathbf{x}, \bullet)$$

$$\mathbf{x}_i \cdot \mathbf{x}_j \mapsto \mathbf{z}(\mathbf{x}_i) \cdot \mathbf{z}(\mathbf{x}_j) = k(\mathbf{x}_i, \bullet) \cdot k(\mathbf{x}_j, \bullet) = k(\mathbf{x}_i, \mathbf{x}_j)$$

カーネル化SVM（SVMまとめ）

- 最適化問題

$$\max_{\alpha_1,\ldots,\alpha_n} \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j)$$
$$\text{subject to} \quad 0 \le \alpha_i \le C, \sum_i y_i \alpha_i = 0$$

- サポートベクトル

$$S: 0 < \alpha_i < C \qquad O: \alpha_i = C \qquad I: \alpha_i = 0$$

- 識別関数

$$y = \mathrm{sgn}\left[\sum_{i \in S \cup O} \alpha_i k(\mathbf{x}_i, \mathbf{x}) + b\right]$$

- しきい値

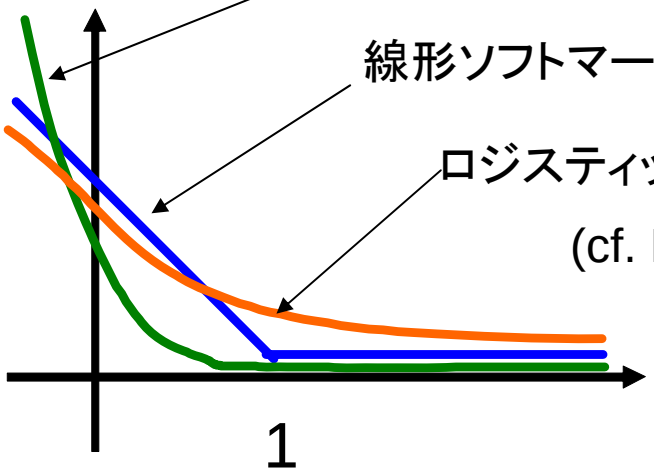
$$b = y_i - \sum_j \alpha_j k(\mathbf{x}_j, \mathbf{x}_i) \qquad (i \in S)$$

2.SVMの仲間たち

- SVMの特徴とそれを支える要素：
  - 線形モデル→カーネル化可能
  - 正則化→汎化性
  - 凸2次計画問題→グローバルミニマム，スパース性
    ロバスト性
- 逆にこれらの要素を満たすような定式化をすれば，
  - マージン最大化（正則化項）やソフトマージン（損失関数）を別の形に一般化出来る
  - 識別問題でない問題（回帰など）に適用可能

損失関数

- 局所解の一意性のために凸関数をとるのがよい
- さらに，区分線形・2次なら凸2次計画法となり，サポートベクターだけの関数になりスパースネスなどの性質をもつ



2次ソフトマージン $R(\xi) = (1-\xi)_+^2$

線形ソフトマージン $R(\xi) = (1-\xi)_+$

凸2次計画
（サポートベクター）

ロジスティック $R(\xi) = \log(1+\exp(-\xi))$

(cf. PLRM 松井先生の講義参照)

$$\xi = y(\mathbf{a}\cdot\mathbf{x}+b)$$

ν SVM

- 通常のソフトマージンSVMではパラメータCの取り方が直感的でない
- マージンを超えるサンプルの割合の上限値νを指定（**νトリック**）

$$\frac{1}{2}\|\mathbf{a}\|^2 - \nu\rho + \frac{1}{m}\sum_{i=1}^{m}\left(\rho - y_i(\mathbf{a}\cdot\mathbf{x}_i + b)\right)_+ \rightarrow \min_{\mathbf{a},b,\rho}$$

- 追加パラメータρ． マージンの値はρ/||a||
  これが大きくなるような正則化項
- 定理：もとのSVMでC=1/ρに取ればν SVMと同じ結果が得られる
- 双対問題

$$\max_{\alpha_1,\ldots,\alpha_n} -\frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j k(\mathbf{x}_i,\mathbf{x}_j)$$

$$\text{subject to} \quad 0 \le \alpha_i \le \frac{1}{m}, \sum_i y_i\alpha_i = 0, \sum_i \alpha_i \ge \nu$$

LP (Linear Programming) マシン

- SVMではマージンの大きさが2乗ノルムであったが，それを変更することが考えられる

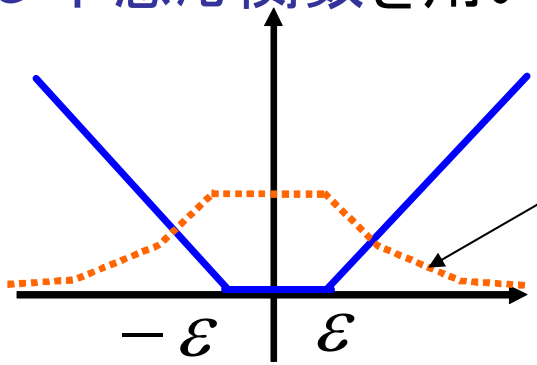
$$f(x) = \sum_i \omega_i k(\mathbf{x}_i, \mathbf{x})$$

  - 正則化項（SVMでは2次形式） $\sum_i |\omega_i|$
  - 損失関数（SVMと同じ） $R(\mathbf{x}, y; f) = \left(1 - yf(\mathbf{x})\right)_+$

- 線形制約・線形目的関数⇒線形計画：2次計画より軽い計算
- 一般に2次の正則化よりもスパースな解が得られる
- マージンといった幾何的対応物はない
- 正則化項がノルムの関数ではないので，Representer theorem は使えない．RKHS全体での最適化ではない．

SVR(Support Vector Regression) (1/2)

- いろいろな損失関数が考えられるが識別の場合と形が似ているのが以下の ε 不感応関数を用いた定式化



$$(|y-(\mathbf{a}\cdot\mathbf{x}+b)|-\varepsilon)_+$$

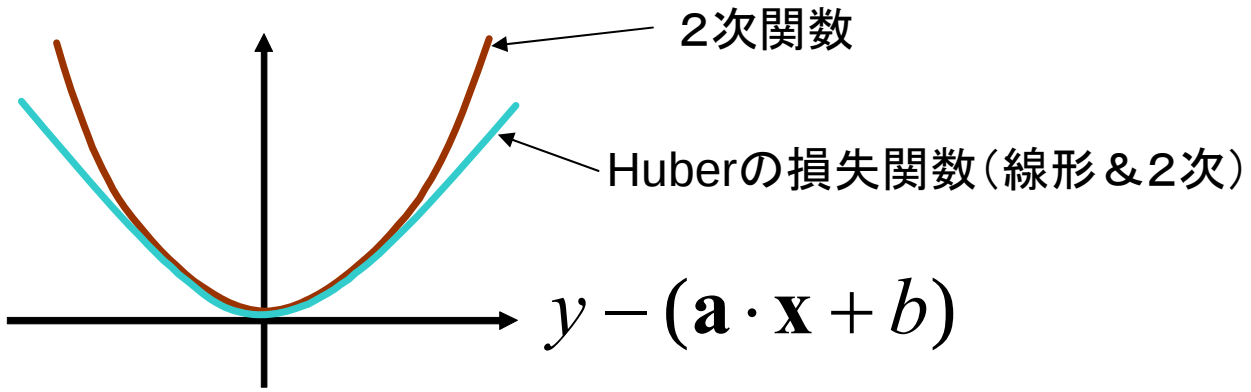
$$y-(\mathbf{a}\cdot\mathbf{x}+b)$$

$$\frac{1}{2}\|\mathbf{a}\|^2+C\sum_{i=1}^{m}(|y_i-(\mathbf{a}\cdot\mathbf{x}_i+b)|-\varepsilon)_+\rightarrow\min_{\mathbf{a},b}$$

- これは識別の場合とほぼ同じ形で双対問題の導出も単純 $y-(\mathbf{a}\cdot\mathbf{x}+b)$ が正のときと負のときがあるのでそれぞれスラック変数が必要

SVR(Support Vector Regression)(2/2)

- ε 不感応性関数は**ロバスト推定**の損失関数の一つ
- それ以外の損失関数でも2次計画問題で解けるものがある



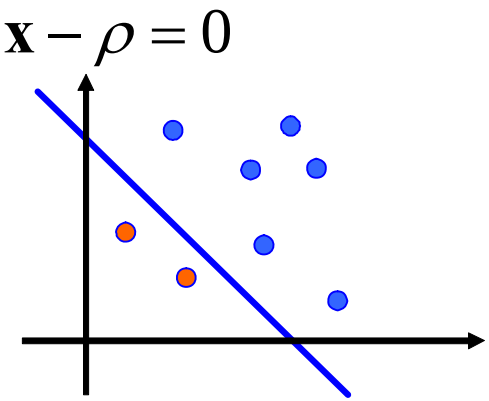
– 2次関数→リッジ回帰・正規過程
（逆行列のみ．2次計画は不要．ただしロバストではない）

– Huberのロバスト損失→凸2次計画

Single class SVM

- 異常値検出などの応用では，入力ベクトルのドメインを推定し，外れ値の判定が必要となる（νトリックでモデル化）

$$\frac{1}{2}\|\mathbf{a}\|^2 - \rho + \frac{1}{\nu m}\sum_{i=1}^{m}(\rho - \mathbf{a}\cdot\mathbf{x}_i)_+ \rightarrow \min_{\mathbf{a},\rho}$$



- できるだけ原点から遠く，かつ多くのデータを平面の外側に含むようにする規準
- 双対問題＋カーネル化

$$\max_{\alpha_1,\ldots,\alpha_n} -\frac{1}{2}\sum_{i,j}\alpha_i\alpha_j k(\mathbf{x}_i,\mathbf{x}_j) \quad 0\le\alpha_i\le\frac{1}{\nu m}, \sum_i \alpha_i = 1$$

$$\rho = \sum_j \alpha_j k(\mathbf{x}_j,\mathbf{x}_i) \qquad (\forall i : \alpha_i \neq 0)$$

周辺話題1：多クラス分類(1/2)

- 2クラス問題に落とす

	識別手法	識別器の数	1識別器あたりのサンプル数	特徴
1 vs other	マージンの大きいクラスに分類	クラス数 K	全サンプル	単純 サンプル数に偏り
1 vs 1	グラフィカルモデルなどを使って識別結果を統合	最大2クラスの組み合わせ K(K-1)/2	2クラス分のサンプル	1識別器の学習は軽い．組み合わせ方が難しい．多重比較法などとの関連性
ECOC(error correcting output coding)	クラスを二つに分け，0,1の符号語を割り当て誤り訂正	符号語の数 (通常 >K)	全サンプル	有効な符号化法の設計が重要

周辺話題1 : 多クラス分類(2/2)

- ２次計画問題として定式化
  最も素直だが必ずしも決定版ではない

$$\frac{1}{2}\sum_{r=1}^{M}\|\mathbf{a}_r\|^2+\frac{C}{m}\sum_{i=1}^{m}\sum_{r\neq y_i}\left(2-((\mathbf{a}_{y_i}-\mathbf{a}_r)\cdot\mathbf{x}_i+(b_{y_i}-b_r))\right)_+ \rightarrow \min_{\mathbf{a}_1,\ldots,\mathbf{a}_M,b_1,\ldots,b_M}$$

$$y=\arg\max_r \mathbf{a}_r\cdot\mathbf{x}+b_r$$

- (SVM の代わりに)カーネル判別分析を使う
  ロバストネスやスパースネスなどのメリットが必要ならば
  SVMの方がよい

周辺話題2：マージン定義

- カーネル化してしまうと入力空間で見るともはや「真ん中」を通る識別面ではない
- Amari & Wu 1999
  場所によって解像度を変える．（カーネル設計とも関連）
- Akaho 2002
  入力空間でのマージンを大きくする
- Invariance に関する事前知識を入れる
  変換に不変な方向への計量を小さくする
  Translation invariance kernel

3.汎化能力の評価

- 汎化能力＝サンプルだけではなく，背後にある未知のデータに対してもフィットできる能力
- SVMは高次元に写像しているにも関わらず，経験的には高い汎化能力をもつ（次元の呪いの克服）
- 正則化は一つの説明だが，定量的には不足
- PAC学習の枠組みを使って，SVMの汎化能力が次元によらないことを示す

期待損失と経験損失

- サンプルは常に未知の確率分布 $P(\mathbf{x}, y)$ に従って生成されるとする
- パラメータ α をもつ識別器を用いて $\mathbf{x}$ の識別を行ったとき，正解 y に対する損失（誤差）$r(\mathbf{x}, y; \alpha)$ が与えられているとする
（以下の説明では2値の誤り誤差を考える）
- 期待損失（未知のサンプルに対する誤差）最小：理想

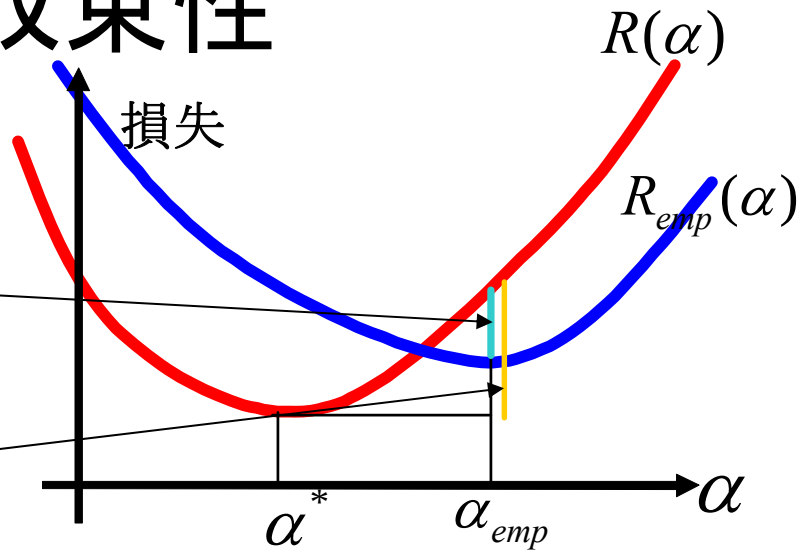
$$R(\alpha) = \int r(\mathbf{x}, y; \alpha) dP(\mathbf{x}, y) \rightarrow \min \rightarrow R(\alpha^*)$$

- 経験損失（サンプル誤差）最小：現実

$$R_{emp}(\alpha) = \frac{1}{m} \sum_i r(\mathbf{x}_i, y_i; \alpha) \rightarrow \min \rightarrow R_{emp}(\alpha_{emp})$$

損失の一様収束性

- 汎化能力を表すいくつかの指標
  - α_{emp}における経験損失と期待損失の差
    $R(\alpha_{emp}) - R_{emp}(\alpha_{emp})$
    見積もりに比べて実際どれだけ損するか
  - α^*と α_{emp}とでの期待損失の差
    $R(\alpha_{emp}) - R(\alpha^*)$
    最適なものに比べてどれだけ損するか



- Hoeffding不等式 $P[|R(\alpha) - R_{emp}(\alpha)| \geq \varepsilon] = 2\exp(-2m\varepsilon^2)$ （α固定）
- 一様収束

$$R(\alpha_{emp}) - R(\alpha^*) < R(\alpha_{emp}) - R(\alpha^*) + R_{emp}(\alpha^*) - R_{emp}(\alpha_{emp})$$
$$= \{R(\alpha_{emp}) - R_{emp}(\alpha_{emp})\} - \{R(\alpha^*) - R_{emp}(\alpha^*)\}$$
$$\leq |R(\alpha_{emp}) - R_{emp}(\alpha_{emp})| + |R(\alpha^*) - R_{emp}(\alpha^*)|$$
$$\leq 2\sup_\alpha |R(\alpha) - R_{emp}(\alpha)|$$

$$\Longrightarrow P[\sup_\alpha |R(\alpha) - R_{emp}(\alpha)| \geq \varepsilon]$$

簡単な場合

- α が有限集合Bのときは比較的単純

$$P(A_1 \cup A_2) = P(A_1) + P(A_2) - P(A_1 \cap A_2) \le P(A_1) + P(A_2)$$

$$P(\bigcup_i A_i) \le \sum_i P(A_i)$$

$$P(\sup_{\alpha \in B}(R(\alpha) - R_{emp}(\alpha)) > \varepsilon) \le \sum_{\alpha \in B} P(R(\alpha) - R_{emp}(\alpha)) > \varepsilon)$$

$$\le 2|B|\exp(-2m\varepsilon^2)$$

- 無限集合のときも基本的には有限の場合に帰着させる

対称化

$$P(\sup_{\alpha}(R(\alpha) - R_{emp}(\alpha)) > \varepsilon) \le 2P(\sup_{\alpha}(R_{emp}(\alpha) - R'_{emp}(\alpha)) > \varepsilon/2)$$

一般に無限の可能性

違う学習セットでの損失の差

有限（2値損失なら高々2^{2m}）

VC次元(1/3)

- 2値損失の場合

$$P(\sup_{\alpha}(R_{emp}(\alpha)-R'_{emp}(\alpha))>\varepsilon/2)\le NP(R_{emp}(\alpha)-R'_{emp}(\alpha)>\varepsilon/2)$$

Nは場合の数

$$P(R_{emp}(\alpha)-R'_{emp}(\alpha)>\varepsilon)=P(\frac{1}{m}\sum_i\xi_i-\frac{1}{m}\sum_i\xi'_i\ge\varepsilon)\le 2\exp(-m\varepsilon^2/2)$$

α 固定

$\xi_i=y_i-f(\mathbf{x}_i;\alpha)$ i.i.d. の2値確率変数

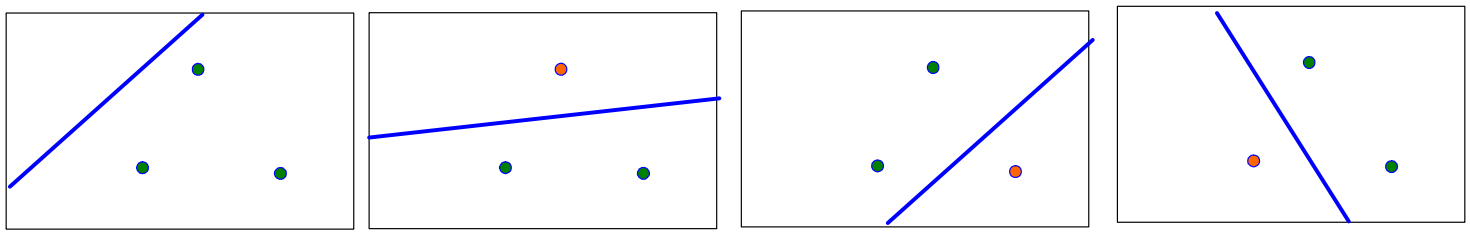
- $(\xi_1,\ldots,\xi_m)$ の場合の数の評価（最大 2^m ）
  関数のクラスの複雑さに応じてmが大きくなると2^m通りを下回る．2^m 通りを実現できる最大の $m=h$ をその関数クラスの VC次元(Vapnik-Chervonenkis dimension)と呼ぶ.
- 定理：$(\xi_1,\ldots,\xi_m)$ の場合の数 N はVC次元を h とすると,

$$N\le\begin{cases}2^m & (m\le h)\\ (em/h)^h & (m>h)\end{cases}$$

VC次元(2/3)

- VC次元の幾何的な意味
  サンプルをうまく配置した上で，すべてのラベル付けができる(これをshutterされるという)最大のサンプル数

  例：2次元の線形識別器のVC次元＝3



- 従って，$m>h$ なら

$$P(\sup(R(\alpha)-R_{emp}(\alpha))>\varepsilon)\le 4(em/h)^h \exp(-m\varepsilon^2/8)$$

- 右辺を δ と置いて書き換えると，$m>h$のとき，確率1-δ 以上で

$$R(\alpha)\le R_{emp}(\alpha)+\sqrt{\frac{8}{m}\left(h\log\frac{2em}{h}+\log\frac{4}{\delta}\right)}$$

VC次元(3/3)

- かなり荒っぽい上界で，これを改善する理論も数多く存在する.
- 線形識別器のVC次元は $h=n+1$　(n:入力次元)
- 次元を上げれば上げるほど汎化能力が悪くなる
（次元の呪い）
- 一方，SVMではマージンを大きくするという規準を入れたので，何も制約のない場合よりは汎化能力はよいだろう
- マージン識別器のVC次元が入力次元によらない上界をもつことを示す→次元の呪いを受けない

マージン識別器のVC次元

定理：原点を通るマージンγ以上の識別器のクラスのVC次元の上限は $h \le D^2 / \gamma^2$

ただしDはデータを含む原点を通る最小の球の半径

証明) 1) k個のサンプルが shutter されている⇒ $\forall y_i, k\gamma \le \left\| \sum_i y_i \mathbf{x}_i \right\|$

$$\left.\begin{array}{l} \because\ y_i \mathbf{a} \cdot \mathbf{x}_i \ge 1 \Rightarrow \sum_i \mathbf{a} \cdot \left(\sum_i y_i \mathbf{x}_i \right) \ge k \\ 1 / \|\mathbf{a}\| \ge \gamma \end{array}\right\}$$

+ Cauchy-Schwarz の不等式より

2) $\left\| \sum_i y_i \mathbf{x}_i \right\|^2 \le kD^2$を満たす y_iが存在する

$\because\ y_i \in \{\pm 1\}$ を i.i.d な一様分布とすると

$$E\left\| \sum_i y_i \mathbf{x}_i \right\|^2 = \sum_i E\left\| y_i \mathbf{x}_i \right\|^2 = \sum_i \left\| \mathbf{x}_i \right\|^2 \le kD^2$$

平均が不等式を満たせば符号式を満たすyは存在する

3) 上記 1),2) より, $k \le D^2 \gamma^2$

注：経験的VC次元，Luckiness

- 今まで述べたVC次元の理論では，関数クラスは学習前に事前に決まっている必要がある.
- ところがSVMではマージンの値は学習後に決まる
- 従って，前節までの議論は厳密には成立しない

- 「たまたまデータの分布がLucky で大きなマージンが得られた」という事象の確率を評価する必要がある.
(Herbrich2001などを参照)

Leave-one-out 誤差

- Leave-one-out 誤差（一つ抜いたサンプルで学習したものの平均）

$$L(D_m) = \sum_{i=1}^{m} r(\mathbf{x}_i, y_i; \alpha_{emp}(D_m - (\mathbf{x}_i, y_i))) \qquad D_m = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m}$$

定理：leave-one-out 誤差は以下の意味で不偏

$$E\left[\frac{L(D_{m+1})}{m+1}\right] = E[R(\alpha_{emp})] \Leftarrow$$

m個のデータから学習したパラメータの未知データに対する誤差の期待値

証明：Lの中の各項の期待値が右辺の値に等しい

- **サポートベクターの数** k_m

定理：r が2値損失のとき，

$$E[R(\alpha_{emp})] \leq \frac{E[k_{m+1}]}{m+1}$$

証明：サポートベクトル以外のサンプルを除いても識別関数は変化せず，leave-one-out 誤差は0．そこでサポートベクトルについては最悪全部損失が1だとすると最大 leave-one-out 誤差は k_m

4.学習アルゴリズム

- 制約つきNewton法（内点法），最急降下法
- Chunking 法
  （いくつかの変数ごとに最適化(他は固定)
  やはり凸2次計画問題となる）
  特に2変数の場合⇒
  SMO (Sequential Minimal Optimization)アルゴリズム
  等式制約から結局1変数の2次関数の最小化に帰着され，
  陽に解が求まる
- オンライン学習
  SVMでもオンライン学習の方法はいくつか提案されているが，
  スパースネスを保ちつつ軽い計算量で行うのが困難

5.カーネル設計の基本

- カーネル組み合わせの基本操作
- 構造化データ
- 計数カーネル
- 畳み込みカーネル
- 確率モデルに基づくカーネル
  - Fisher カーネル, 周辺化カーネル
  - 拡散カーネル
- カーネルが含むハイパーパラメータの選択

カーネル組み合わせの基本操作(1/2)

- 正定値性を保つカーネルの基本操作

1. 和(含む正定数加算) $k(\mathbf{x}_1, \mathbf{x}_2) = k_1(\mathbf{x}_1, \mathbf{x}_2) + k_2(\mathbf{x}_1, \mathbf{x}_2)$
2. 積(含む正定数倍) $k(\mathbf{x}_1, \mathbf{x}_2) = k_1(\mathbf{x}_1, \mathbf{x}_2) k_2(\mathbf{x}_1, \mathbf{x}_2)$
3. 関数の積 $k(\mathbf{x}_1, \mathbf{x}_2) = c(\mathbf{x}_1) c(\mathbf{x}_2)$

- 基本的には上の3つに尽きる

注:シグモイドカーネル $k(\mathbf{x}, \mathbf{y}) = \dfrac{1}{1 + \exp(-\beta \mathbf{x} \cdot \mathbf{y})}$

はニューラルネットワークやロジスティック回帰との関連性で用いられることがあるが,
厳密には正定値カーネルではない

カーネル組み合わせの基本操作(2/2)

- 応用操作
  1. 特徴量同士の内積 $k(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{z}(\mathbf{x}_1) \cdot \mathbf{z}(\mathbf{x}_2)$

     (関数の積の和)
  2. 共形変換 $k(\mathbf{x}_1, \mathbf{x}_2) = c(\mathbf{x}_1) c(\mathbf{x}_2) k_1(\mathbf{x}_1, \mathbf{x}_2)$

     (conformal transformation)
  3. 指数関数 $k(\mathbf{x}_1, \mathbf{x}_2) = \exp(k_1(\mathbf{x}_1, \mathbf{x}_2))$

     (Taylor 展開すれば積和, ガウスカーネルはこれと共形変換の組合せ)
  4. べき乗 $k(\mathbf{x}_1, \mathbf{x}_2) = k_1(\mathbf{x}_1, \mathbf{x}_2)^p$

     (pは自然数, 多項式カーネルはこの一種)

さらにこれらの再帰的な繰返しによりいくらでも複雑にできる

構造化データ

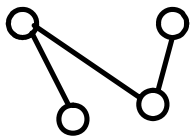
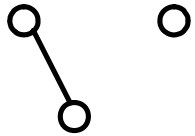
- 文字列，グラフなど
- 基本はたくさんの特徴量を計算して
  $k(\mathbf{x},\mathbf{y}) = \mathbf{z}(\mathbf{x}) \cdot \mathbf{z}(\mathbf{y})$ の形のカーネルを計算
- ポイントは特徴ベクトルが大きいのでいかに効率よく計算できるか
- 離散構造の入力を連続値の特徴ベクトルに変換する際の問題点：
  - 適切な数量化（類似度としての意味付け）
  - 次元の違いの吸収など

- 詳細は松井先生の講義を参照

計数カーネル

- 計数(count)カーネル
  グラフなどで，部分構造をいくつ含んでいるかを特徴ベクトルとし，その内積によって定義されるカーネル

- 単純な例：[頂点の数，辺の数]


[4,3]

[3,1] ⇨ $k(x,y)=[4,3]\cdot[3,1]=15$

- 数ではなく「ある，なし」の２値ベクトルにすることも多い

畳み込みカーネル(Haussler1999)

- 複雑な構造をしていて全体に対するカーネルを定義するのが難しいときは，部分に対するカーネルを組み合わせる分割統治が有効
- 畳み込みカーネル (convolution kernel)

$\mathbf{x}$ がD個の部分構造 $\mathbf{x}_1,\ldots,\mathbf{x}_D$から$\mathbf{x} = c(\mathbf{x}_1,\ldots,\mathbf{x}_D)$ によって作られているとみなせるとする． $\mathbf{y}$ のi番目の部品 $\mathbf{y}_i$ と $\mathbf{x}_i$ の間に $k_i(\mathbf{x}_i,\mathbf{y}_i)$ が定義されているとき，

$$k(\mathbf{x},\mathbf{y}) = \sum_{\{\mathbf{x}_i\}\in c^{-1}(\mathbf{x})} \sum_{\{\mathbf{y}_i\}\in c^{-1}(\mathbf{y})} \prod_{i=1}^{D} k_i(\mathbf{x}_i,\mathbf{y}_i)$$

- 基本的に特徴量の和と積だけなので正定値性は自明

Fisher カーネル(Jaakkola et al.1999)

- 入力　$\mathbf{x}$ の確率モデル $p(\mathbf{x};\boldsymbol{\theta})$が与えられるとき，そのパラメータ $\hat{\boldsymbol{\theta}}$に対し，特徴ベクトル $\mathbf{f}(\mathbf{x};\hat{\boldsymbol{\theta}}) = \partial \log p(\mathbf{x};\hat{\boldsymbol{\theta}}) / \partial \boldsymbol{\theta}$ (Fisherスコア)の内積で定まるカーネル

$$k(\mathbf{x},\mathbf{y}) = \mathbf{f}(\mathbf{x};\hat{\boldsymbol{\theta}})^T I^{-1} \mathbf{f}(\mathbf{y};\hat{\boldsymbol{\theta}})$$

ただし I はFisher 情報行列の推定値$I = E_{\hat{\boldsymbol{\theta}}}[\mathbf{f}(\mathbf{x};\hat{\boldsymbol{\theta}})\mathbf{f}(\mathbf{x};\hat{\boldsymbol{\theta}})^T]$

- 正定値性は自明
- 指数分布族なら特徴ベクトルは十分統計量の空間上の点とみなせ，Fisher 情報行列の逆はそこでの自然な計量
- 入力の確率モデルにとってよい特徴量が識別に役立つとは限らないが，識別が入力の分布に多少とも依存していれば有効（詳細は K.Tsuda, et al.2003 参照）
- 実用上は，離散構造を定まった次元の実数ベクトルに変換できる点が重要

周辺化カーネル(Tsuda et al.2002)

- 潜在変数モデル $p(\mathbf{x},\mathbf{z};\theta)$ があり，観測変数 $\mathbf{x}$ とする．
  $(\mathbf{x},\mathbf{z})$ に対するカーネル $k_C((\mathbf{x}_1,\mathbf{z}_1),(\mathbf{x}_2,\mathbf{z}_2))$
  が定義されているとき，潜在変数について期待値を取った

$$k(\mathbf{x}_1,\mathbf{x}_2) = E_\theta[k_C((\mathbf{x}_1,\mathbf{z}_1),(\mathbf{x}_2,\mathbf{z}_2)) \mid \mathbf{x}_1,\mathbf{x}_2]$$

$$= \int k_C((\mathbf{x}_1,\mathbf{z}_1),(\mathbf{x}_2,\mathbf{z}_2)) p(\mathbf{z}_1 \mid \mathbf{x}_1;\theta) p(\mathbf{z}_2 \mid \mathbf{x}_2;\theta) d\mathbf{z}_1 d\mathbf{z}_2$$

  を周辺化カーネル(marginalized kernel)という
- これは共形変換の和を取っているので正定値性は自明

拡散カーネル(Kondor et al.2002)

- 有限ドメインでのカーネル⇒グラム行列で決まる.
- グラム行列が正定値対称ならOK
- 拡散カーネル(diffusion kernel):
  下のような無向グラフの上のネットワーク流を考える

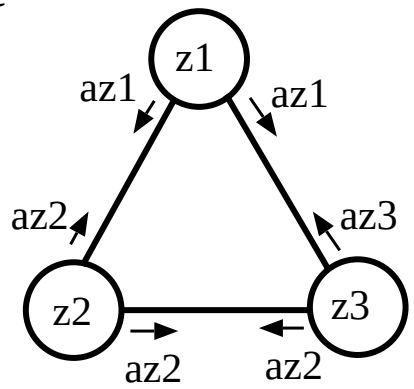
$$\frac{d}{dt}\mathbf{Z} = \alpha A\mathbf{Z} \qquad A = \begin{cases} 1 & \ldots \quad (i,j):\text{connected} \\ -d_i & \ldots \quad i=j \\ 0 & \ldots \quad \text{otherwise} \end{cases}$$

d_i は結合リンクの数

- 標準正規分布でノードを初期化した
  ときの時刻 ½ における **Z** の共分散行列

$$K = \exp(\alpha A)$$

（行列指数関数: A が対称なら正定値対称となる）



カーネルが含むハイパーパラメータ選択

- 基本的にモデル選択の問題
  - クロスバリデーションが有効な場合が多い
- ガウスカーネル
  - σ=0 の極限でグラム行列対角（1-NN）
  - $\sigma = \infty$の極限でグラム行列が退化 $\alpha_i = C$（多い方のクラスに分類）
  - Cとの関係
    - C=0の極限で$\sigma = \infty$と同じ
    - C=∞の極限でハードマージン
- カーネル選択の基本指針
  - 個々の問題に対して，類似度としての意味があるようにする
  - 対角成分が大きくなりすぎないようにする（1-NNになる）
  - グラム行列が退化しすぎないようにする

参考文献

- SVM入門
  - 赤穂昭太郎, 津田宏治: “サポートベクターマシン 基本的仕組みと最近の発展”, In 別冊・数理科学 脳情報数理科学の発展, サイエンス社, 2002.
  - N.Cristianini, J.Shawe-Taylor:*An Introduction to Support Vector Machines*, Cambridge University Press, 2000.（邦訳：サポートベクターマシン入門，共立出版）
  - 前田英作: “痛快! サポートベクトルマシン”, 情報処理, 42(7), 676--683, 2001.
  - K.-R.Mueller, S.Mika, G.~Raetsch, K.Tsuda,B.Schoelkopf: “An introduction to kernel-based learning algorithms”, *IEEE Trans. on Neural Networks*, 12(2),181--201, 2001.
  - 津田宏治: “サポートベクターマシンとは何か”,電子情報通信学会誌, 83(6), 460--466, 2000.
  - 津田宏治: “カーネル法の理論と実際”, In 甘利他: パターン認識と学習の統計学 (統計科学のフロンティア 6),岩波書店, 2003.

- 汎化能力の評価などが詳しい解説
  - R.Herbrich: *Learning Kernel Classifiers Theory and Algorithms*, MIT Press, 2001.
  - B.Schoelkopf, A.J.Smola: *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*, MIT Press, 2001.
  - V.N.Vapnik: *The Nature of Statistical Learning Theory*, Springer-Verlag, 1995.
  - V.N.Vapnik: *Statistical Learning Theory*, John Wiley & Sons, 1999.

- 構造化データに対するカーネル設計に関する解説
  - J.Shawe-Taylor, N.Cristianini:*Kernel Methods for Pattern Analysis*, Cambridge University Press, 2004.

- SVMを含む機械学習全般
  - T.Hastie, R.Tibshirani, J.Friedman: *The Elements of Statistical Learning*, Springer-Verlag, 2001.
  - 渡辺澄夫: データ学習アルゴリズム (データサイエンスシリーズ 6), 共立出版, 2001.
  - 渡辺澄夫（編）:学習システムの理論と実現，森北出版,2005.

- 個々に参照した文献
  - S.Amari, S.Wu:Improving support vector machine classifiers by modifying kernel functions, Neural Networks, 12(6), 1999.
  - S. Akaho: SVM that maximizes the margin in the input space, Systems and Computers in Japan, Vol. 35, No. 14, pp. 78-86, 2004.
  - K.Tsuda, S.Akaho,M.Kawanabe,K.R.Mueller: Asyptotic theory of Fisher kernels, *Neural Computation,* vol.16, no.1115-137,2003