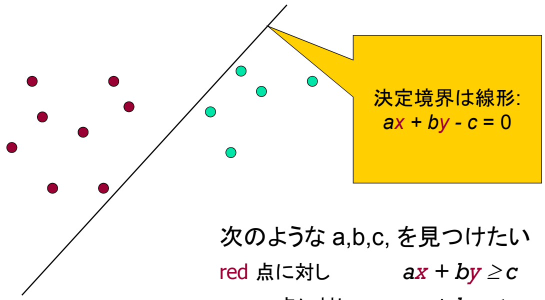
情報意味論(10)
サポートベクターマシン

理工学部管理工学科
櫻井彰人

基礎的復習: 線形判別関数

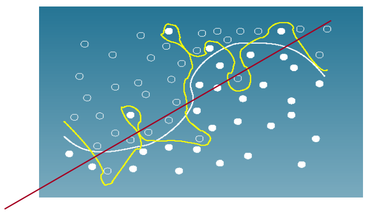


次のような a,b,c, を見つけたい

red 点に対し　$ax + by \ge c$

green 点に対し　$ax + by \le c$.

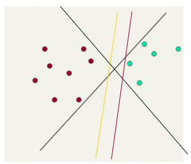
これも復習: 複雑な境界は？



Christopher Manning のスライドから

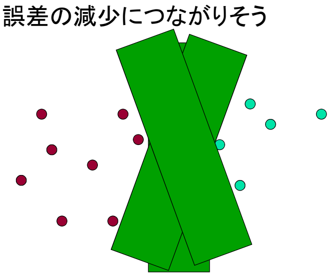
どの超平面を選ぶべきか?

- *a,b,c* にはいくつもの可能性あり
- 見つけたどれもが最良なわけではない
  [何か「よさ」の基準を設ける必要はある]
  - パーセプトロン学習アルゴリズムではどうであったか？
- サポートベクターマシンは「最良」のものをみつける.
  - 超平面とそれに近い「困難点」との距離を最大化する
  - 直感的解釈: 決定境界に近いところに(別のクラスの)点がなければ、決定の不確実さは少なかろう



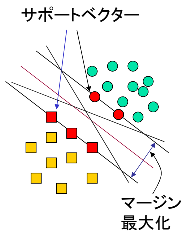
直感的解釈をもう一つ

- 分離境界を幅のある帯に置き換えてみよう。この幅が狭いときには、選択範囲がせばまり、汎化誤差の減少につながりそう



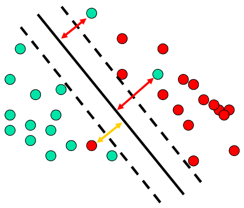
サポートベクターマシン (SVM)

- SVM は、分離超平面周囲のマージンを最大化する.
  - ラージマージン分類器ともいう
- 決定関数はサポートベクターと呼ばれる訓練データによって完全に定まる.
- 2次計画問題である
- 広範囲の問題に対してうまくいく方法であると考えられている



ラージマージン分類器

線形分離可能でないならば
- 誤りを許す
  - コストを払って、本来あるべき場所に動かす
- ただ, 超平面はどちらのクラスからも遠ざける



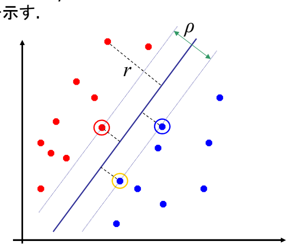
最大マージン: 定式化

- w: 決定超平面への垂線ベクトル
- x_i: i 番目のデータ点
- y_i: 属するクラス (+1 or -1)　注: 1/0 ではない
- 分類器:　$\mathrm{sign}(w^T x_i + b)$
- そのとき x_i の関数マージン:　$y_i (w^T x_i + b)$
  - 勿論 w を大きくすればマージンは増大する、そこで、、、

（訓練データ全体の関数マージンは、上記の値の最大値）

幾何的マージン

- データ点から分離超平面までの距離　$r = \frac{\mathbf{w}^T\mathbf{x}+b}{\|\mathbf{w}\|}$
- 分離超平面に最も近い点がサポートベクター.
- 分離超平面のマージンρ は相異なるクラスのサポートベクターがどの程度分離しているかを示す.



線形 SVM を数学的に

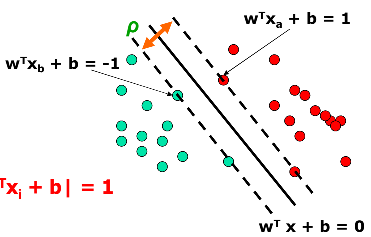
- 全ての点が超平面から関数値で 1 離れていると仮定しよう. そうであれば次の2つの制約が訓練データ集合 $\{(\mathbf{x_i}, y_i)\}$ から得られる

$$\mathbf{w^T x_i} + b \ge 1 \quad \text{if } y_i = 1$$
$$\mathbf{w^T x_i} + b \le -1 \quad \text{if } y_i = -1$$

- サポートベクターに対しては、上記不等式は等式となる; そうなると, 各データの超平面からの距離は $r = \frac{\mathbf{w}^T\mathbf{x}+b}{\|\mathbf{w}\|}$ であるから、マージンは次の値となる: $\rho = \frac{2}{\|\mathbf{w}\|}$

線形サポートベクターマシン

- 超平面
  $w^T x + b = 0$
- 制約:
  $\min_{i=1,\dots,n} |w^T x_i + b| = 1$
- 書換えると:
  $w^T(x_a - x_b) = 2$
  $\rho = \|x_a - x_b\|_2 = 2/\|w\|_2$



線形サポートベクターマシン

- 次の2次計画問題が得られる:

> 次のような w と b を見出せ:
> $\rho = \frac{2}{\|\mathbf{w}\|}$ は最大であり, 全ての $\{(\mathbf{x_i}, y_i)\}$ につき
> $\mathbf{w^T x_i} + b \ge 1$ if $y_i=1$;　$\mathbf{w^T x_i} + b \le -1$ if $y_i = -1$

- よりよい定式化 (min ||**w**|| = max 1/ ||**w**||):

> 次のような **w** と b を見出せ:
> $\Phi(\mathbf{w}) = ½\, \mathbf{w^T w}$ は最小であり, すべての $\{(\mathbf{x_i}, y_i)\}$ につき
> $y_i (\mathbf{w^T x_i} + b) \ge 1$

最適化問題の解法

次のような $\mathbf{w}$ と b とを見出せ
最小化: $\Phi(\mathbf{w}) = ½\, \mathbf{w}^T\mathbf{w}$;
全ての $\{(\mathbf{x_i}, y_i)\}$ につき: $y_i(\mathbf{w}^T\mathbf{x_i} + b) \geq 1$

- 線形制約のもとでの2次関数の最適化
- 2次計画問題は、よく知られた数理計画問題の一つ. 多くの解法が知られている
- 解法にあたっては、ラグランジュ乗数 α_i を主問題の各制約に割付けた双対問題を構成する:

次のような $\alpha_1 \ldots \alpha_N$ を見出せ
最大化: $Q(\boldsymbol{\alpha}) = \Sigma\alpha_i - ½\Sigma\Sigma\alpha_i\alpha_j y_i y_j \mathbf{x_i}^T\mathbf{x_j}$;
(1) $\Sigma\alpha_i y_i = 0$
(2) $\alpha_i \geq 0$, 任意の α_i

主問題のラグランジアンは

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\mathbf{w}\cdot\mathbf{w} - \sum_i \alpha_i\left(y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1\right)$$

従って、

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial \mathbf{w}} = \mathbf{w} - \sum_i \alpha_i y_i \mathbf{x}_i \qquad \frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial b} = \sum_i \alpha_i y_i$$

となるゆえ、停留点は、

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i \qquad 0 = \sum_i \alpha_i y_i$$

これらを主問題に戻せば

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\mathbf{w}\cdot\mathbf{w} - \sum_i \alpha_i\left(y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1\right)$$

$$= \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j \mathbf{x}_i\cdot\mathbf{x}_j - \sum_{i,j}\alpha_i\alpha_j y_i y_j \mathbf{x}_i\cdot\mathbf{x}_j + \sum_i \alpha_i$$

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

最適化問題の解法

- 解の形は:

$\mathbf{w} = \Sigma\alpha_i y_i \mathbf{x_i}$, かつ $\alpha_k \neq 0$ なるすべての $\mathbf{x_k}$ につき $b = y_k - \mathbf{w}^T\mathbf{x_k}$

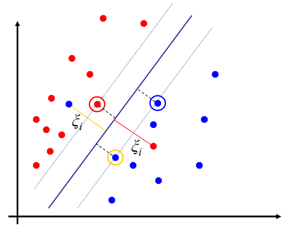
- 各非零の α_i は、対応する $\mathbf{x_i}$ がサポートベクターであることを示す.
- 識別関数は次のようになる:

$f(\mathbf{x}) = \Sigma\alpha_i y_i \mathbf{x_i}^T\mathbf{x} + b$

- 当該式は新規点とサポートベクトル $\mathbf{x_i}$ の**内積**であることに注意.
- また、当該最適化問題を解くには、訓練データのすべての組合せに関する内積 $\mathbf{x_i}^T\mathbf{x_j}$ の計算が含まれていることを注意しておく.

ソフトマージン分類器

- もし訓練データが線形分離可能でなければ, スラック変数 ξ_i を用いて分類が難しい点やノイズがのった点の誤分類を許すようにする.



ソフトマージン分類器

- 以前の定式化:

次のような $\mathbf{w}$ と b とを見出せ
最小化: $\Phi(\mathbf{w}) = ½\, \mathbf{w}^T\mathbf{w}$;
すべての $\{(\mathbf{x_i}, y_i)\}$ について: $y_i(\mathbf{w}^T\mathbf{x_i} + b) \geq 1$

- スラック変数を含む、新しい定式化:

次のような $\mathbf{w}$ と b とを見出せ
最小化: $\Phi(\mathbf{w}) = ½\, \mathbf{w}^T\mathbf{w} + C\Sigma\xi_i$;
すべての $\{(\mathbf{x_i}, y_i)\}$ について: $y_i(\mathbf{w}^T\mathbf{x_i} + b) \geq 1 - \xi_i$, かつ
すべての i について: $\xi_i \geq 0$

- パラメータ C は過学習を制御する方法と見ることができる.

ソフトマージン分類器 – 解

- ソフトマージン分類器の双対問題:

次のような $\alpha_1 \ldots \alpha_N$ を見出せ:
最大化: $Q(\boldsymbol{\alpha}) = \Sigma\alpha_i - ½\Sigma\Sigma\alpha_i\alpha_j y_i y_j \mathbf{x_i}^T\mathbf{x_j}$; ただし
(1) $\Sigma\alpha_i y_i = 0$
(2) すべての α_i につき $0 \leq \alpha_i \leq C$

- スラック変数 ξ_i もラグランジュ乗数も、双対問題には表れていない！
- 再び, 非零の α_i に対応する $\mathbf{x_i}$ はサポートベクターである.
- 当該双対問題への解は:

$\mathbf{w} = \Sigma\alpha_i y_i \mathbf{x_i}$
$b = y_k(1 - \xi_k) - \mathbf{w}^T\mathbf{x_k}$ where $k = \underset{k}{\operatorname{argmax}}\ \alpha_k$

明示的には $\mathbf{w}$ がなくても分類できる!

$f(\mathbf{x}) = \Sigma\alpha_i y_i \mathbf{x_i}^T\mathbf{x} + b$

主問題のラグランジアンは

$$L(\mathbf{w},b,\boldsymbol{\alpha}) = \frac{1}{2}\mathbf{w}\cdot\mathbf{w} + C\sum_i \xi_i - \sum_i \alpha_i\left(y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - (1-\xi_i)\right) - \sum_i \nu_i\xi_i$$

従って、

$$\frac{\partial L(\mathbf{w},b,\boldsymbol{\alpha})}{\partial \mathbf{w}} = \mathbf{w} - \sum_i \alpha_i y_i \mathbf{x}_i \qquad \frac{\partial L(\mathbf{w},b,\boldsymbol{\alpha})}{\partial b} = \sum_i \alpha_i y_i \qquad \frac{\partial L(\mathbf{w},b,\boldsymbol{\alpha})}{\partial \xi_i} = C - \alpha_i - \nu_i$$

となるゆえ、停留点は、

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i \qquad 0 = \sum_i \alpha_i y_i \qquad 0 = C - \alpha_i - \nu_i$$

これらを主問題に戻せば

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

ただし、条件は、

$$0 = \sum_i \alpha_i y_i \qquad 0 \le \alpha_i \le C \quad (\text{for all } i)$$

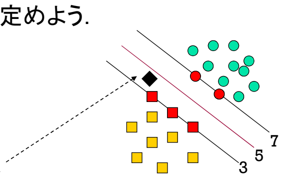
SVMを用いた分類

- 所与の新点 ($x1$,$x2$) に対し, その超平面への垂直射影を計る(scoreとしよう):
  - 2次元の場合: score = $w_1x_1+w_2x_2+b$.
  - すなわち: score = $wx + b = \Sigma\alpha_i y_i \mathbf{x}_i^T\mathbf{x} + b$
  - 信頼限度 t を定めよう.

score > t : yes

score < -t : no

それ以外: 判定放棄



線形 SVM: まとめ

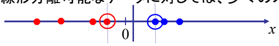
- 分類器は、分離超平面 *separating hyperplane.*
- 最も重要な訓練データ点がサポートベクターとなる; それが当該超平面を決める.
- 2次計画問題を解けば、どの点 $\mathbf{x}_i$ がサポートベクターで非零のラグランジュ乗数 α_i に対応するかが分かる.
- 当該問題の双対問題においても解法においても、訓練データ点は、内積の中にしか現れない:

次のような $\alpha_1 \ldots \alpha_N$ を見出せ:
最大化: $\mathbf{Q}(\boldsymbol{\alpha}) = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j$, 但し
(1) $\Sigma\alpha_i y_i = 0$
(2) すべての α_i につき: $0 \le \alpha_i \le C$

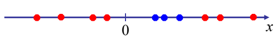
$$f(\mathbf{x}) = \Sigma\alpha_i y_i \mathbf{x}_i^T\mathbf{x} + b$$

非線形 SVM

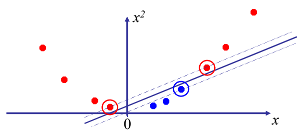
- 線形分離可能なデータに対しては、少々のノイズがあっても、うまくいく:



- しかし、データ集合が線形分離可能でなかったらどうしよう?

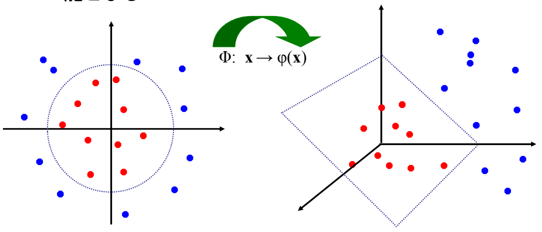


- 例えば… データをより高次元の空間に写像したらどうだろうか:



非線形 SVM: 特徴空間

- 一般的なアイデア: もともとの特徴空間は、いつでも、ある高次元特徴空間に写像すれば、線形分離可能となる:



高次元空間への写像: 問題点

- 計算時間:
  - データが1001個あれば、(非線形関数で)1000次元に写像すれば、必ず、線形分離できる。
  - しかし、非線形関数の計算は時間がかかるのに、データ1個につき1000回の計算(共通部分を多くして計算するにせよ)が必要では、大変な計算量となる

  ⇒ カーネル関数の利用(カーネルトリック)による、計算量の大幅な削減
- 汎化能力:
  - データが1001個あれば、(非線形関数で)1000次元に写像すれば、必ず、線形分離できる。
  - それは、すなわち、どんな教師データであっても、それを実現できるということ。すなわち、過学習!!

  ⇒ この問題の解決こそ、ラージマージン分類器の本領

高次元空間への非線形写像

- データ x から高次元空間 F への（非線形）写像 $\phi(x) = x'$ を考える. $\phi : R^N \to F$

主問題のラグランジアンは

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\mathbf{w}\cdot\mathbf{w} - \sum_i \alpha_i \left(y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1\right)$$

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\mathbf{w}\cdot\mathbf{w} - \sum_i \alpha_i \left(y_i(\mathbf{w}\cdot\phi(\mathbf{x}_i) + b) - 1\right)$$

双対問題: 拘束条件付き最大化

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

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$$

$$\sum_i \alpha_i y_i = 0 \text{ and } \forall i\ \alpha_i \ge 0$$

カーネルトリック “Kernel Trick”

- 注目すべきは、$\Phi(x)$ は、$\Phi(x)\cdot\Phi(y)$ というように、内積でしか表れない。

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$$

- そこで、もし、$K(x,y)=\Phi(x)\cdot\Phi(y)$ となる、簡単な関数Kがあれば、計算が非常に楽になる。
  - 特に、$K(x,y)$ が $x{\cdot}y$ の関数であるとさらに楽になる

Mercerの定理

- 関数 K が内積の形で書ける:

$$K(x,y) = \sum_{i=1}^{\infty} \lambda_i \phi_i(x)\phi_i(y)$$

必要十分条件は、K が対称かつ半正定値であることである. i.e.,

$$K(x,y) = K(y,x)$$

$$\iint K(x,y) f(x) f(y)\, dx dy \ge 0 \quad \text{for any } f$$

なお、$\phi_i(x)$は $K(x,y)$の固有関数となる. i.e.,

$$\int K(x,y)\phi_i(x)\, dx = \lambda_i \phi_i(y)$$

よく使われるカーネル関数

- 線形カーネル $K(x,y) = x^T y$
- 多項式カーネル $K(x,y) = (x^T y + 1)^p \text{ or } (x^T y)^p$
- RBFカーネル $K(x,y) = \exp\left(-\|x-y\|^2 / 2\sigma^2\right)$
- MLP $K(x,y) = \tanh\left(\beta_0 x^T y + \beta_1\right)$
- 例: 2-次元ベクトル $\mathbf{x}=[x_1\ \ x_2]$ に対し $K(\mathbf{x_i},\mathbf{x_j})=(1 + \mathbf{x_i}^T\mathbf{x_j})^2$,とおく
このとき、次の式が成立する $K(\mathbf{x_i},\mathbf{x_j})= \varphi(\mathbf{x_i})^T\varphi(\mathbf{x_j})$:

$K(\mathbf{x_i},\mathbf{x_j})=(1 + \mathbf{x_i}^T\mathbf{x_j})^2$,

$= 1+ x_{i1}^2x_{j1}^2 + 2\, x_{i1}x_{j1}\, x_{i2}x_{j2} + x_{i2}^2x_{j2}^2 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2}$

$= [1\ \ x_{i1}^2\ \ \surd 2\ x_{i1}x_{i2}\ \ x_{i2}^2\ \ \surd 2x_{i1}\ \ \surd 2x_{i2}]^T\ [1\ \ x_{j1}^2\ \ \surd 2\ x_{j1}x_{j2}\ \ x_{j2}^2\ \ \surd 2x_{j1}\ \ \surd 2x_{j2}]$

$= \varphi(\mathbf{x_i})^T\varphi(\mathbf{x_j})$

ただし $\varphi(\mathbf{x}) = [1\ \ x_1^2\ \ \surd 2\ x_1x_2\ \ x_2^2\ \ \surd 2x_1\ \ \surd 2x_2]$

SVM: 汎化能力の推定

- 汎化能力最大(新規データに対して最も正確)の分類器がほしい.
- 良い汎化性能を得るための糸口は?
  - 訓練データを大きくする
  - 訓練データに対する誤りを小さくする
  - 容量/分散 (モデル記述パラメータ数, モデルの表現能力) をおおきくする
- SVM では、これらの量に基づいて、新規データに対する誤差限界を明示的に示すことができる.

容量/分散: VC 次元

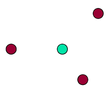
- 理論的なリスク限界:

$$R(\alpha) \le R_{emp}(\alpha) + \sqrt{\frac{h(\log(2l/h)+1) - \log(\eta/4)}{l}}$$

- Risk = 平均誤り率
- α – 当該モデル (パラメータで決まる)
- R_{emp} – 経験リスク, l – 観測数, h – VC 次元, 当該式は確率 (1-η)で正しい
  - VC (Vapnik-Chervonenkis) 次元/容量: shatterできる点の最大数
  - ある点集合がshatterできるとは、その任意のラベル付けを当該分類器が行えること.
- 重要な理論的性質. しかし, 実際にはあまり使われない

例

- d 次元空間に n 個の点があり、それらは、red か green とラベルが付けられていると仮定する. n を (d の関数として) どれだけ大きくとれば, red 点と green 点が線形分離でなくなる例が作れるか?
- 例, d =2 に対しては $n \geq 4$.



スケッチ: マージン最大化の理論的な正当化

- Vapnik は次のことを証明した:
  最適な線形判別器クラスの VC 次元 h は、次の上界をもつ

$$h \le \min\left\{\left\lceil \frac{D^2}{\rho^2} \right\rceil, m_0\right\} + 1$$

  ただし ρ はマージン, D は訓練事例をすべて囲い込む最小の超球の直径, そして m_0 は(事例の表現空間の)次元である.

- 直感的に, これは空間の次元 m_0 にかかわりなく, マージン ρ を最大化することにより、VC 次元を最小化することができる.
- こうして, 分類器の複雑度は、次元数に関わりなく小さく保つことができる.

SVM の性能

- SVM は、最良の性能を持つと考える人は多い.
- 多くの場合、統計的な有意性は明確ではない.
- SVM と同程度の性能をもつ手法は他にもある.
- 例: regularized logistic regression (Zhang & Oles)
  - Tong Zhang, Frank J. Oles: Text Categorization Based on Regularized Linear Classification Methods. Information Retrieval 4(1): 5-31 (2001)
- 比較研究の例: Yang & Liu
  - Yiming Yang, Xin Liu: A re-examination of text categorization methods, 22nd Annual International SIGIR (1999).

評価例: 古典的な Reuters データ

- 非常によく使われたデータセット
- 21578 documents
- 9603 training, 3299 test articles (ModApte split)
- 118 categories
  - 一つの article は複数の category に属しうる
  - 118 個の2値分類
- 1 document 当たりの category 数
  - 1.24
- 10 categories のみ大きい(全 118 categories)

大きめの categories (#train, #test)

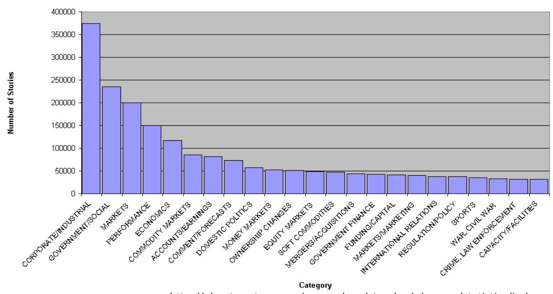
- Earn (2877, 1087)
- Acquisitions (1650, 179)
- Money-fx (538, 179)
- Grain (433, 149)
- Crude (389, 189)
- Trade (369,119)
- Interest (347, 131)
- Ship (197, 89)
- Wheat (212, 71)
- Corn (182, 56)

Reuters Text Categorization data set (**Reuters-21578**) document 例

```
<REUTERS TOPICS="YES" LEWISSPLIT="TRAIN" CGISPLIT="TRAINING-SET" OLDID="12981" NEWID="798">
<DATE> 2-MAR-1987 16:51:43.42</DATE>
<TOPICS><D>livestock</D><D>hog</D></TOPICS>
<TITLE>AMERICAN PORK CONGRESS KICKS OFF TOMORROW</TITLE>
<DATELINE>    CHICAGO, March 2 - </DATELINE><BODY>The American Pork Congress kicks off
tomorrow, March 3, in Indianapolis with 160 of the nations pork producers from 44 member states determining industry positions on a number of issues, according to the National Pork Producers Council, NPPC.
    Delegates to the three day Congress will be considering 26 resolutions concerning various issues, including the future direction of farm policy and the tax law as it applies to the agriculture sector. The delegates will also debate whether to endorse concepts of a national PRV (pseudorabies virus) control and eradication program, the NPPC said.
    A large trade show, in conjunction with the congress, will feature the latest in technology in all areas of the industry, the NPPC added. Reuter
&#3;</BODY></TEXT></REUTERS>
```

New Reuters: RCV1: 810,000 文書

- Reuters RCV1 の頻出トピック



http://about.reuters.com/researchandstandards/corpus/statistics/index.asp

（クラス当たりの）評価尺度

- Recall: クラス *i* のdocument中、正しく *i* に分類されたものの割合:
$$\frac{c_{ii}}{\sum_j c_{ij}}$$
- Precision:クラス *i* に分類されたdocument中、本当にクラス *i* に属するものの割合:
$$\frac{c_{ii}}{\sum_j c_{ji}}$$
- “Correct rate”: (1- error rate) 正しく分類されたdocumentの割合:
$$\frac{\sum_i c_{ii}}{\sum_j \sum_i c_{ij}}$$

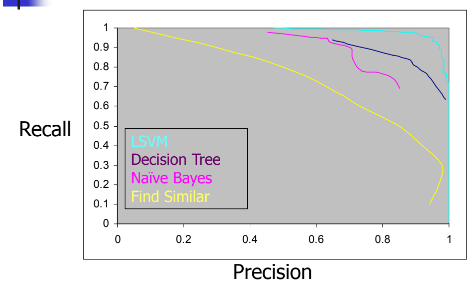
Dumais et al. 1998: Reuters – Break-Even Performance

	Rocchio	NBayes	Trees	LinearSVM
earn	92.9%	95.9%	97.8%	98.2%
acq	64.7%	87.8%	89.7%	92.8%
money-fx	46.7%	56.6%	66.2%	74.0%
grain	67.5%	78.8%	85.0%	92.4%
crude	70.1%	79.5%	85.0%	88.3%
trade	65.1%	63.9%	72.5%	73.5%
interest	63.4%	64.9%	67.1%	76.3%
ship	49.2%	85.4%	74.2%	78.0%
wheat	68.9%	69.7%	92.5%	89.7%
corn	48.2%	65.3%	91.8%	91.1%
Avg Top 10	64.6%	81.5%	88.4%	91.4%
Avg All Cat	61.7%	75.2%	na	86.4%

Break Even: (Recall + Precision) / 2

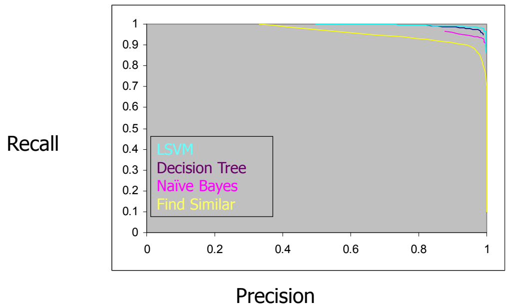
S. T. Dumais, J. Platt, D. Heckerman, and M. Sahami. Inductive learning algorithms and representations for text categorization. In CIKM-98: Proceedings of the Seventh International Conference on Information and Knowledge Management, 1998.

Precision vs. Recall - Category “Grain”

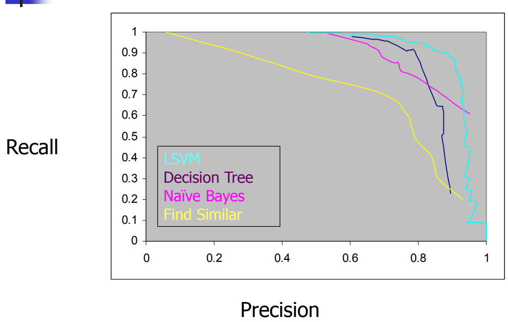


Recall: = TP/(TP+TN); % 当該カテゴリ中そのカテゴリに属すると判定したもの
Precision: = TP/(TP+FP); % そのカテゴリに属するとした中で本当にそのカテゴリに属するもの

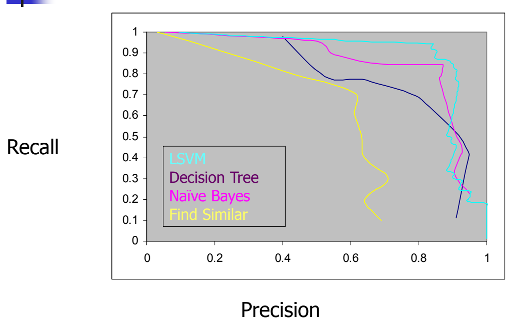
Precision vs. Recall - Category “Earn”



Precision vs. Recall - Category “Crude”



Precision vs. Recall - Category “Ship”



カーネルによる違い (Joachims)

	Bayes	Rocchio	C4.5	k-NN	SVM (poly) degree $d =$ 1	2	3	4	5	SVM (rbf) width $\gamma =$ 0.6	0.8	1.0	1.2
earn	95.9	96.1	96.1	97.3	98.2	98.4	**98.5**	98.4	98.3	**98.5**	98.5	98.4	98.3
acq	91.5	92.1	85.3	92.0	92.6	94.6	**95.2**	95.2	95.3	95.0	95.3	95.3	**95.4**
money-fx	62.9	67.6	69.4	78.2	66.9	72.5	75.4	74.9	**76.2**	74.0	75.4	**76.3**	75.9
grain	72.5	79.5	89.1	82.2	91.3	93.1	**92.4**	91.3	89.9	**93.1**	91.9	91.9	90.6
crude	81.0	81.5	75.5	85.7	86.0	87.3	88.6	**88.9**	87.8	**88.9**	89.0	88.9	88.2
trade	50.0	77.4	59.2	77.4	69.2	75.5	76.6	77.3	**77.1**	76.9	78.0	**77.8**	76.8
interest	58.0	72.5	49.1	74.0	69.8	63.3	67.9	73.1	**76.2**	74.4	75.0	**76.2**	76.1
ship	78.7	83.1	80.9	79.2	82.0	85.4	86.0	**86.5**	86.0	**85.4**	86.5	87.6	87.1
wheat	60.6	79.4	85.5	76.6	83.1	84.5	85.2	**85.9**	83.8	**85.2**	85.9	85.9	85.9
corn	47.3	62.2	87.7	77.9	86.0	86.5	85.3	**85.7**	83.9	**85.1**	85.7	85.7	84.5
microavg.	**72.0**	**79.9**	**79.4**	**82.3**	84.2	85.1	85.9	86.2	85.9	86.4	86.5	86.3	86.2
					combined: **86.0**					combined: **86.4**			

Fig. 2. Precision/recall-breakeven point on the ten most frequent Reuters categories and microaveraged performance over all Reuters categories, k-NN, Rocchio, and C4.5 achieve highest performance at 1000 features (with k = 30 for k-NN and β = 1.0 for Rocchio). Naive Bayes performs best using all featurcs.

T. Joachims, *Text Categorization with Support Vector Machines: Learning with Many Relevant Features*. Proceedings of the European Conference on Machine Learning (ECML), Springer, 1998

Yang&Liu: SVM vs 他の手法

Table 1: Performance summary of classifiers

method	miR	miP	miF1	maF1	error
SVM	.8120	.9137	.8599	.5251	.00365
KNN	.8339	.8807	.8567	.5242	.00385
LSF	.8507	.8489	.8498	.5008	.00414
NNet	.7842	.8785	.8287	.3765	.00447
NB	.7688	.8245	.7956	.3886	.00544

miR = micro-avg recall; miP = micro-avg prec.;
miF1 = micro-avg F1; maF1 = macro-avg F1.

まとめ

- サポートベクターマシン (SVM) は
  - サポートベクターに基づいて超平面を決める
    - Support vector = 判定境界付近のクリティカルな点
  - 線形 SVM は線形分類器.
  - カーネル: 高次元へ写像するが、その内積は低次元の内積で簡単に計算できる
  - リスクの上界 (リスク = テストデータでの期待誤り)
  - (邪魔な属性が多いときの)分類器としてベスト?
    - 数1000も属性があるときは、安定的に強い
  - ポピュラー: SVMlight がきっかけ?
    - 速くて無料 (研究目的には)
    - 他にもいくつか: TinySVM, libsvm, ….

参考

- A Tutorial on Support Vector Machines for Pattern Recognition (1998) Christopher J. C. Burges
- S. T. Dumais, Using SVMs for text categorization, IEEE Intelligent Systems, 13(4):21-23, Jul/Aug 1998
- S. T. Dumais, J. Platt, D. Heckerman and M. Sahami. 1998. Inductive learning algorithms and representations for text categorization. *Proceedings of CIKM '98*, pp. 148-155.
- A re-examination of text categorization methods (1999) Yiming Yang, Xin Liu 22nd Annual International SIGIR
- Tong Zhang, Frank J. Oles: Text Categorization Based on Regularized Linear Classification Methods. Information Retrieval 4(1): 5-31 (2001)
- Trevor Hastie, Robert Tibshirani and Jerome Friedman, "Elements of Statistical Learning: Data Mining, Inference and Prediction" Springer-Verlag, New York.
- 'Classic' Reuters data set: http://www.daviddlewis.com /resources /testcollections/reuters21578/
- T. Joachims, *Learning to Classify Text using Support Vector Machines*. Kluwer, 2002.

カーネル法

カーネルにはいろいろ

- 「カーネル」という用語は、一般には（対称や正定値とは限らない）2変数関数に使われることも多い。
  例）ノンパラメトリックな確率密度推定

  $$p(x) = \frac{1}{N}\sum_{i=1}^{N} g(x - x_i)$$

  などに用いる密度関数 $g(x)$ も「カーネル」と呼ばれる（関係はあるのですが）。
  例）積分変換の核関数

- 最近では、正定値カーネルのことを単に「カーネル」「カーネル法」と呼ぶことが多いので、注意が必要。

カーネル法の動機

- 線形関数の学習には良い性質が多い
  - 最適解は一つ
  - 高速な学習アルゴリズムが存在する
  - 統計的な解析がうまくできる
- しかし、大きな問題がある
  - 学習能力が不十分

歴史を紐解くと

- 例の Minsky & Pappert "Perceptrons" でその弱点を明らかにした
- ニューラルネットワークは、この弱点を、線形関数に対し非線形の活性化関数を施すことによって、克服した
  - 学習能力の低さを解決し、学習アルゴリズムが広く利用できることを示した
  - しかし学習速度の遅さ、極小点の多さという問題に遭遇した

そこで、カーネル法

- カーネル法は、線形関数に拘るのだが、それを高次元空間で行おうというもの:

$$\phi : x \in X \longmapsto \phi(x) \in \mathcal{F}$$

- 期待しているのは、特徴空間が、入力空間よりはるかに高い次元を持つということ.

例

- 次の写像を考える

$$(x_1, x_2) \mapsto (x_1^2, x_1 x_2, x_2 x_1, x_2^2)$$

- この特徴空間で線形制約を考えると:

$$a x_1^2 + b x_2^2 = c$$

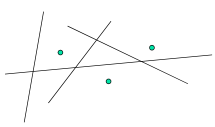
- (この場合)楕円になる – i.e. すなわち、元の入力空間でみると、非線形な形となる.

特徴空間の表現能力

- 表現能力(従って、学習能力)はその次元に比例する – 例:

定理: m 次元空間の一般位置に m+1 個の点が与えられると、そのどのような2値分類も線形閾値関数で表現できる

- 2次元:



関数の形

- というわけで、カーネル法では、特徴空間では線形関数を用いる:

$$x \longmapsto \langle \mathbf{w}, \phi(x) \rangle$$

- 回帰課題のときには、これが回帰関数とする。
- 分類課題の場合には、この関数値に閾値関数を施す必要がある(単に、0 or 1 にするためです)

$$x \longmapsto \mathrm{sign}\left(\langle \mathbf{w}, \phi(x) \rangle + b \right)$$

高次元空間での問題

- 学習能力は、実は、高すぎることになる。そして過学習が発生する:
  どんな分類も表現できるということは、汎化能力がないということである
- 長いベクトルを扱うわけであるから、計算量も馬鹿にならない

対処法

- 過学習への対応:マージン最大化
- 計算量への対応: カーネルトリック

ところで、SVMでは

- 最適化問題は

$$\min_{\mathbf{w},b,\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\xi_i$$

$$\text{subject to}: \quad y_i\left(\langle \mathbf{w}, \phi(x_i)\rangle + b\right) \geq 1 - \xi_i, \quad \xi_i \geq 0$$

計算量を考えてみると

- 2次カーネルでは

$$(x_1, x_2) \mapsto (x_1^2, x_1x_2, x_2x_1, x_2^2)$$

例えば 20x30 = 600 ピクセルの画像に対しては、180000次元となってしまう

- この特徴空間で計算するのは、あまりに馬鹿らしいということになる

双対表現を考える

- 荷重ベクトルの空間を制限したらどうだろうか。例えば、次のように、訓練データの線形結合で表現できるもののみを考える(パーセプトロンアルゴリズムを思い出す):

$$\mathrm{w} = \sum_{i=1}^{m} \alpha_i \phi(x_i)$$

- このとき、未知例に対する値は、

$$\langle \mathrm{w}, \phi(x)\rangle = \left\langle \sum_{i=1}^{m} \alpha_i \phi(x_i), \phi(x)\right\rangle = \sum_{i=1}^{m} \alpha_i \langle \phi(x_i), \phi(x)\rangle$$

双対変数を学習する

- この α_i は双対変数と呼ばれる
- 訓練データで張られる空間に直行する要素は意味がないので、結局、一般の表現定理に行き着く
- この双対変数の学習を行えばよいことになる

SVM の双対問題

- SVMの双対表現は、最適化問題の双対問題としても得られる:

$$\max_{\alpha} \quad \sum_{i=1}^{m} \alpha_i - \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j \left\langle \phi(x_i), \phi(x_j) \right\rangle$$

$$\text{subject to}: \quad \sum_{i=1}^{m} y_i \alpha_i = 0$$

$$0 \le \alpha_i \le C \quad \text{for all } i$$

- なお、定数項は、境界上の事例から求まる

ここで、カーネルを用いると

- 内積しか使っていないことに中位
- 仮に、次のようにして計算が簡単にできたら:

$$\kappa(x, z) = \langle \phi(x), \phi(z) \rangle$$

- そうすると、訓練時にも未知データに対しても、(極めて高次元の)特徴空間で計算を明示的に行う必要がなくなる

その例

- 例によって次の例を考えると

$$\phi : (x_1, x_2) \mapsto (x_1^2, x_1 x_2, x_2 x_1, x_2^2)$$

- こんな具合:

$$\begin{aligned}
\langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle &= \left\langle \mathbf{x}\mathbf{x}', \mathbf{z}\mathbf{z}' \right\rangle_F \\
&= \mathrm{tr}(\mathbf{x}\mathbf{x}'\mathbf{z}\mathbf{z}') \\
&= \langle \mathbf{x}, \mathbf{z} \rangle^2
\end{aligned}$$

計算の効率性

- つまり、例えば、先ほどの 180000次元のベクトルの場合、実は 600 次元のベクトルで内積をとり、それを二乗するだけでよいことになる
- 一気に敷衍すれば、無限次元の特徴空間も扱うことができることになる. 例えばガウスカーネル:

$$\kappa(\mathbf{x}, \mathbf{z}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{z}\|^2}{2\sigma^2} \right)$$