情報意味論(13)

(簡単に)事例ベースアプローチ

櫻井彰人
慶應義塾大学理工学部

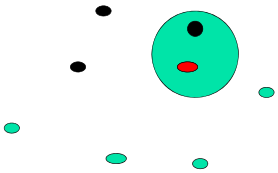
事例ベース学習

- キーアイデア
  - 訓練データ$<x_i, f(x_i)>$を全て憶えていよう(とりあえずは、何も、または、あまりしない)
  - 問い合わせがあったら、その時点で、しよう
- この類に属する方法
  - 最近傍法(Nearest neighbor)
  - *k*-Nearest neighbor
  - Locally weighted regression
  - Radial basis functions
- Lazy 対 eager

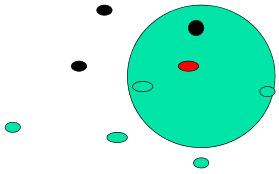
最近傍法

- 最近傍法(Nearest neighbor)
  - 問合せx_q に対し、最近接のx_nを見つけ、$f(x_q) \leftarrow f(x_n)$ とする
- *k*-Nearest neighbor
  - *k*個の最近接データの間で、多数決
  - *k*個の最近接データの間で、平均値

1-Nearest Neighbor



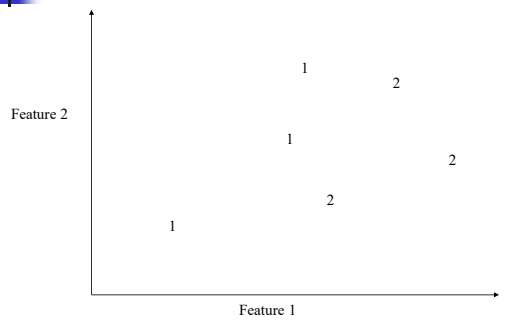
3-Nearest Neighbor



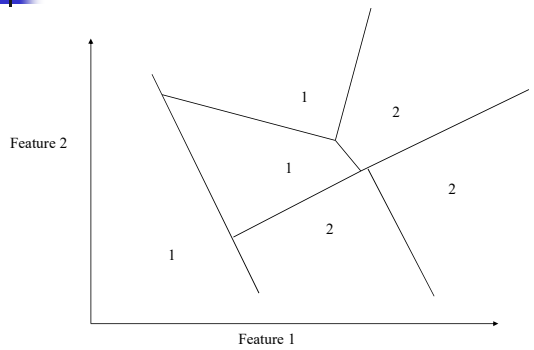
最近傍法の特徴

- いつ使うか
  - 属性が R^n の点とみなせる
  - 属性数はあまり多くない(数十個?)
  - 大量の訓練データ
- 長所
  - 学習が速い
  - 複雑な目標関数も表現可能
  - (訓練データがもつ)情報を失うことがない
- 短所
  - 問合せ時、遅い
  - 無関係な属性によって、簡単に、ごまかされる

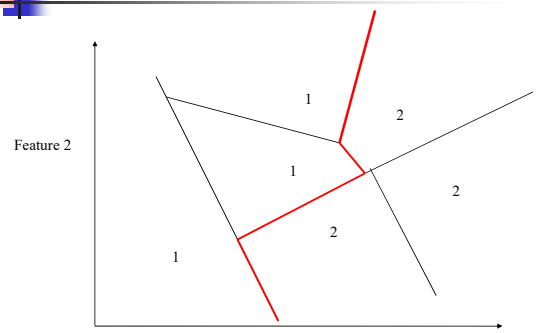
幾何的解釈



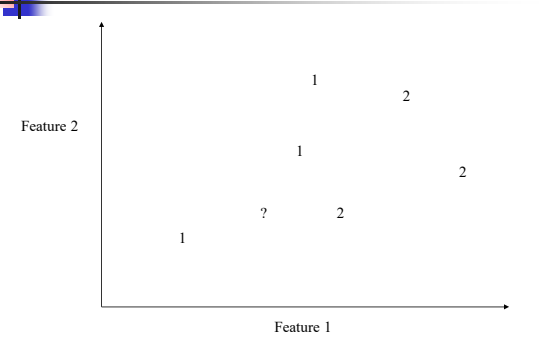
境界



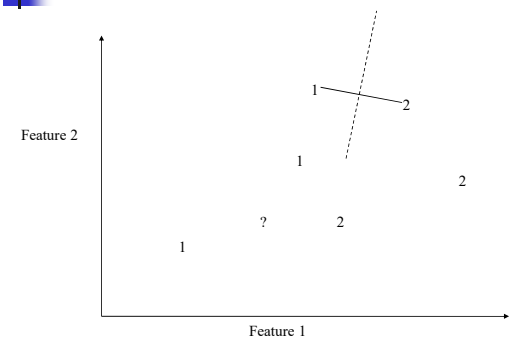
境界



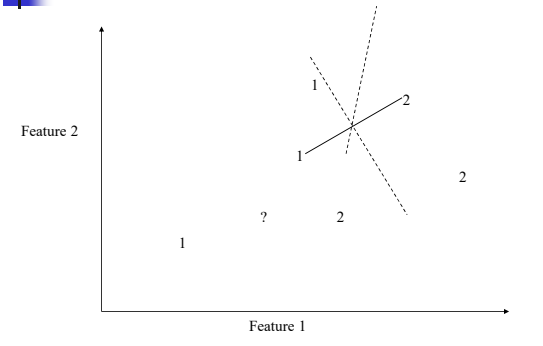
境界を描く



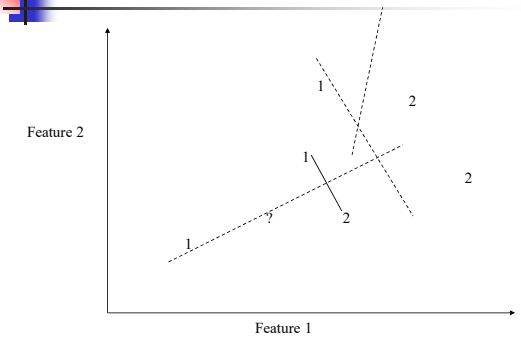
境界を描く



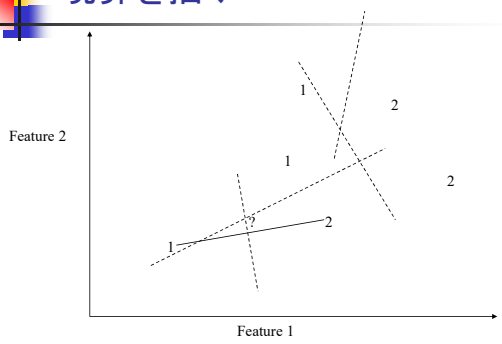
境界を描く



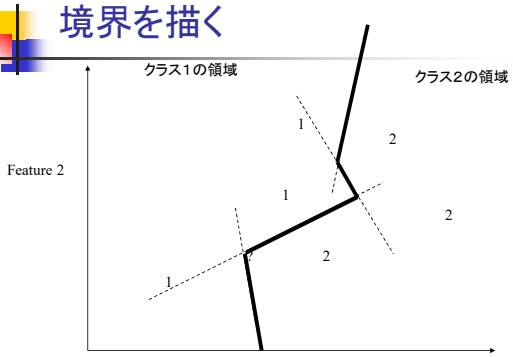
境界を描く



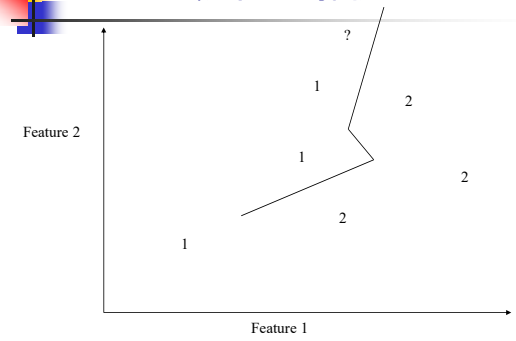
境界を描く



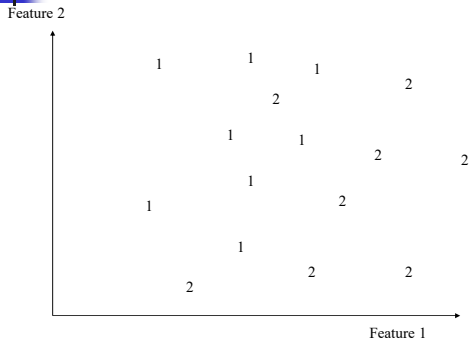
境界を描く



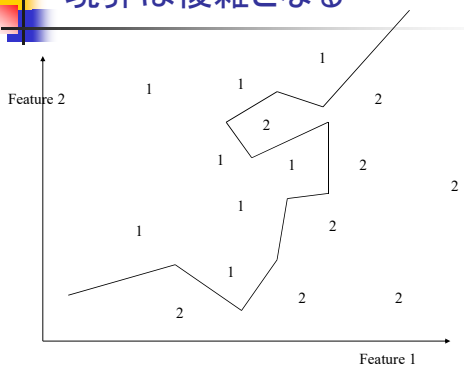
1-NNの幾何的解釈



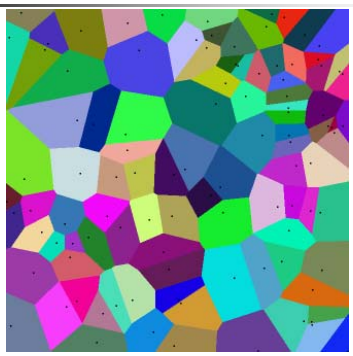
データ点が多いとき

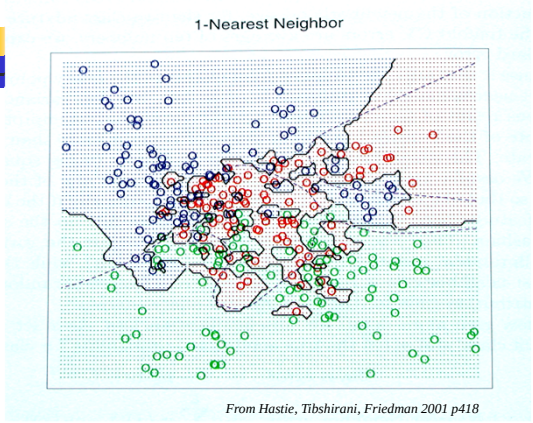


境界は複雑となる

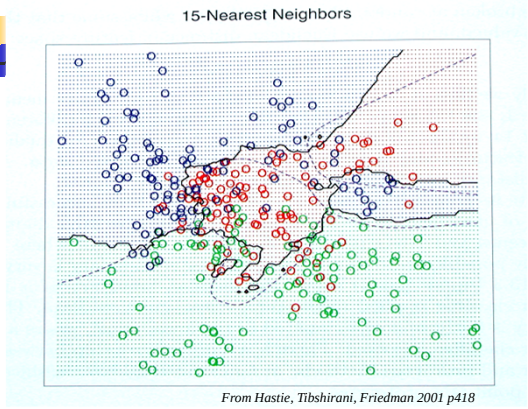


Voronoi図





From Hastie, Tibshirani, Friedman 2001 p418



From Hastie, Tibshirani, Friedman 2001 p418

From Hastie, Tibshirani, Friedman 2001 p419

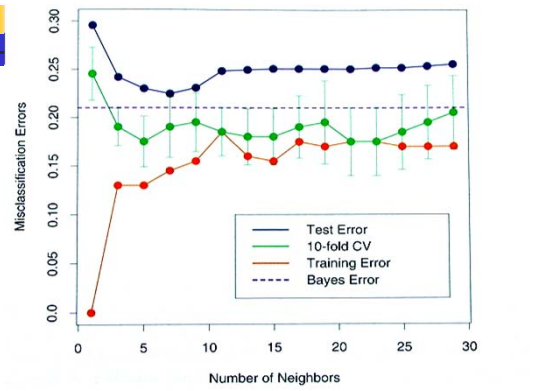


Table 6. Results summary of TC systems on Reuters versions 1–4.

System	Reuters version 1	Reuters version 2	Reuters version 3	Reuters version 4
WORD	—	.15 (Scut)	.31 (Pcut)	.29 (Pcut)
kNN	—	.69 (Scut)	.85 (Scut)	.82 (Scut)
LLSF	—	—	.85 (Scut)	.81 (Scut)
NNets.PARC (perceptron)	—	—	—	.82 (Pcut)
CLASSI (perceptron)	—	—	.80	—
RIPPER (DNF)	—	.72 (Scut)	.80 (Scut)	—
SWAP-1 (DNF)	—	—	.79	—
DTree IND	—	.67 (Pcut)	—	—
DTree C4.5	—	—	.79 (F_1)	—
CHARADE (DNF)	—	—	.78	—
EXPERTS (n-gram)	—	.75 (Scut)	.76 (Scut)	—
Rocchio	—	.66 (Scut)	.75 (Scut)	—
NaiveBayes	—	.65 (Pcut)	.71	—
CONSTRUE (Exp. Sys.)	.90	—	—	—

Yiming Yang, An Evaluation of Statistical Approaches to Text Categorization, Information Retrieval, vol.1, 69-90 (1999)

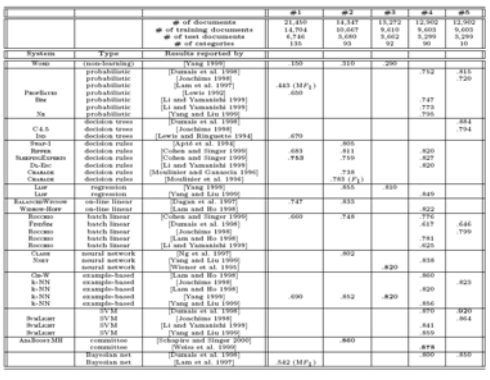


Table 6. Comparative results among different classifiers obtained on five different version of the Reuters collection. Unless otherwise noted, entries indicate the microaveraged breakeven point; within parentheses, "M" indicates macroaveraging and "F_1" indicates use of the F_1 measure. Boldface indicates the best performer on the collection.

Fabrizio Sebastiani, Machine learning in automated text categorization, ACM Computing Surveys, vol.34, no.1, 1-47 (2002)

Table VI. Comparative Results Among Different Classifiers Obtained on Five Different Versions of Reuters. (Unless otherwise noted, entries indicate the microaveraged breakeven point, with parentheses, "M" indicates macroaveraging and "F_1" indicates use of the F_1 measure; boldface indicates the best performer on the collection.)

			#1	#2	#3	#4	#5
		# of documents	21,450	14,347	13,272	12,902	12,902
		# of training documents	14,704	10,667	9,610	9,603	9,603
		# of test documents	6,746	3,680	3,662	3,299	3,299
		# of categories	135	93	92	90	10
System	Type	Results reported by					
Word	non-learning	Yang (1999)	310	299	291		
	probabilistic	[Dumais et al. 1998]				752	815
	probabilistic	[Joachims 1998]				720	
	probabilistic	[Lam et al. 1997]	443 (MF_1)				
PropBayes	probabilistic	[Lewis 1992a]	650				
Bim	probabilistic	[Li and Yamanishi 1999]				747	
Nb	probabilistic	[Li and Yamanishi 1999]				772	
	probabilistic	[Yang and Liu 1999]				795	
	decision tree	[Dumais et al. 1998]					884
C4.5	decision tree	[Joachims 1998]				794	
Ind	decision tree	[Lewis and Ringuette 1994]	670				
Swap-1	decision rules	[Apte et al. 1994]		805			
Ripper	decision rules	[Cohen and Singer 1999]	683	811		820	
SleepingExperts	decision rules	[Cohen and Singer 1999]	753	759		827	
Dl-Esc	decision rules	[Li and Yamanishi 1999]				820	
Charade	decision rules	Moulinier and Ganascia 1996		738			
Charade	decision rules	[Moulinier et al. 1996]		783 (F_1)			
Llsf	regression	Yang (1999)		855	810		
Llsf	regression	[Yang and Liu 1999]				849	
BALANCEWinnow	on-line linear	[Dagan et al. 1997]	747 (M)	833 (M			
Winnow-Herr	on-line linear	[Lam and Ho 1998]				822	
Rocchio	batch linear	[Cohen and Singer 1999]	660	748		776	
FindSim	batch linear	[Dumais et al. 1998]				617	646
Rocchio	batch linear	[Joachims 1998]				799	
Rocchio	batch linear	[Lam and Ho 1998]				781	
Rocchio	batch linear	[Li and Yamanishi 1999]				625	
Classi	neural network	[Ng et al. 1997]		802			
Nnet	neural network	Yang and Liu 1999]			.829	838	
	neural network	[Wiener et al. 1995]					
GIS-W	example-based	[Lam and Ho 1998]				860	
k-NN	example-based	[Joachims 1998]				823	
k-NN	example-based	[Lam and Ho 1998]				820	
k-NN	example-based	Yang 1999]	690	852	.820		
k-NN	example-based	[Yang and Liu 1999]				856	
	SVM	[Dumais et al. 1998]				870	.920
SvmLearn	SVM	[Joachims 1998]				864	
SvmLearn	SVM	[Li Yamanishi 1999]				841	
SvmLearn	SVM	[Yang and Liu 1999]				858	
AdaBoost.MH	committee	[Schapire and Singer 2000]			.860		
	committee	[Weiss et al. 1999]				.878	
	Bayesian net	[Dumais et al. 1998]				800	850
	Bayesian net	[Lam et al. 1997]	542 (MF_1)				

Fabrizio Sebastiani, Machine learning in automated text categorization, ACM Computing Surveys, vol.34, no.1, 1-47 (2002)

極限における振舞い

- $p(x)$: 事例xがラベル1(正)をもつ事後確率
- Nearest neighbor:
  - 事例数→∞のとき、Gibbsアルゴリズムに漸近
  - Gibbs: 確率$p(x)$で1を予測
- k-Nearest neighbor
  - 事例数→∞かつkが大きくなると、Bayes最適
  - Bayes最適: 全ての仮説を考える
    $p(x)$>0.5 なら1、それ以外0

注: Gibbs の期待誤差はBayesの倍以下

復習

Bayes 最適な分類器

$$\arg\max_{c_j \in \{+,-\}} \sum_{h_i \in H} P(c_j \mid h_i) P(h_i \mid D)$$

注: Bayes 最適な分類器は H に含まれるとは限らない

注: 論文にはうまくいくと報告されているのだが、試してみるとMAPやMLと変わらない場合がある。どのような場合にそうなるか、興味のあるところである

注: 実行可能か？ 見るからに時間がかかりそう

Gibbs 分類器 – 速度向上

1. 仮説を P(h|D) に従ってランダムに選ぶ
2. 新事例をこれに従い分類する

慶賀: もし仮説を事前分布 P(h) に従ってランダムに選ぶと,

$E[error_{Gibbs}] \le 2E[error_{BayesOptimal}]$

(詳細は "Mitchell Machine Learning Chap. 6.8")

仮説の個数が多くて、ベイズ最適な分類器が計算できないときに有用

距離荷重つき k-NN

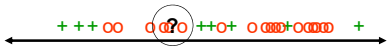
- 近い事例の判断を重視したい

$$\hat{f}(x_q) \leftarrow \frac{\sum_{i=1}^{k} w_i f(x_i)}{\sum_{i=1}^{k} w_i}, \quad w_i \equiv \frac{1}{d(x_q, x_i)^2}$$

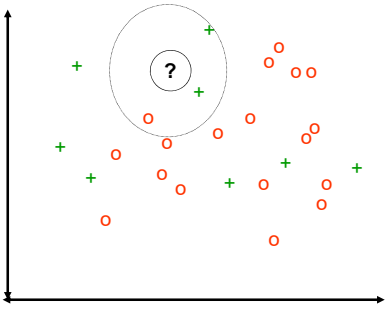
但し、$d(x_q,x_i)$は、x_qとx_iの間の距離

- これにより、k個のみならず全データを使うことに意味がでてくる⇒Shepardの方法

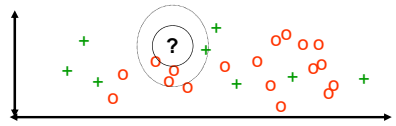
K-NN と不要な特徴



K-NN と不要な特徴



K-NN と不要な特徴



距離の問題

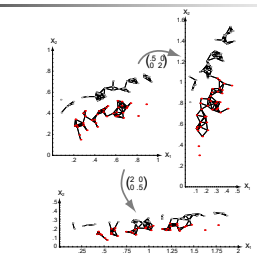
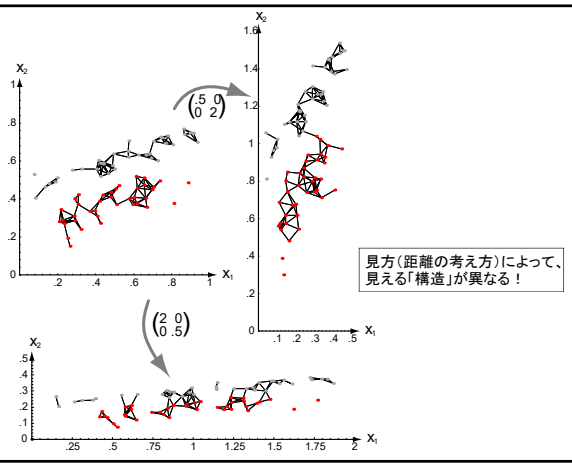


FIGURE10.8. Scaling axes affects the clusters in a minimum distance cluster method. The original data and minimum-distance clusters are shown in the upper left; points in one cluster are shown in red, while the others are shown in gray. When the vertical axis is expanded by a factor of 2.0 and the horizontal axis shrunk by a factor of 0.5, the clustering is altered (as shown at the right). Alternatively, if the vertical axis is shrunk by a factor of 0.5 and the horizontal axis is expanded by a factor of 2.0, smaller more numerous clusters result (shown at the bottom). In both these scaled cases, the assignment of points to clusters differ from that in the original space. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification* . Copyright 2001 by John Wiley & Sons, Inc.



次元の呪い

- 20個の属性で記述されるが、その内、たった2属性のみが意味ある場合を考える
- 次元の呪い：
  - k-NNなら、他の18属性の値でどんな結論も出うる
- 解決方法
  - j番目の属性にz_jの荷重を。z_jは予測誤差最小となるように選択
  - cross-validationを用いて自動的にz_jを決定

Locally weighted regression

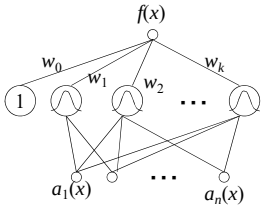
- k-NN は各問合せx_qでfの局所近似を構成していた
- x_qの周囲で$f(x)$の近似関数を明示的に構成したらどうだろうか？
  - k-NNに線型回帰したら？
  - 2次回帰では？
  - 区分回帰したら？
- 最小化すべき誤差にもいくつかの候補が

$$E_1(x_q) \equiv \frac{1}{2} \sum_{x \in x_q \text{の} k\text{-NN}} \left(f(x) - \hat{f}(x_q)\right)^2$$

$$E_2(x_q) \equiv \frac{1}{2} \sum_{x \in D} \left(f(x) - \hat{f}(x_q)\right)^2 K\left(d(x_q, x)\right)$$

Radial Basis Function Network

- 局所近似の線型結合による大域近似
- 神経回路網の一種
- distance-weighted regression に類似
  - lazy ではなく eager であるが



$$f(x) = w_0 + \sum_{u=1}^{k} w_u K_u\left(d(x_u, x)\right)$$

$K_u\left(d(x_u, x)\right)$ の一例

$$K_u\left(d(x_u, x)\right) \equiv e^{-\frac{1}{2\sigma^2} d(x_u, x)^2}$$

RBFの学習

- $K_u(d(x_u,x))$ の x_u の定め方
  - 事例空間に一様にばら撒く
  - 事例を使用（事例の分布が反映）
- 荷重の学習（K_uは正規分布とする）
  - 各K_uの分散（と平均）を定める
    - 例えば、EMを使用
  - K_uを固定したまま、線型出力部分を学習
    - 線型回帰で高速に

Lazy 対 eager

- Lazy: 事例からの一般化をしないでいる。問合せがあったときに考える
  - k-Nearest Neighbor
- Eager: 問合せ前に予め一般化しておく
  - 「学習」アルゴリズム、ID3, 回帰, RBF„
- 違いはあるか？
  - Eager学習は全域的な近似を作成
  - Lazy学習は局所近似を大量に作成
  - 同じ仮説空間を使うなら、lazy の方が複雑な関数を作成
    - over-fitting の可能性
    - 柔軟（複雑なところと単純なところの組合せ）

まとめ

- 事例ベースアプローチ
  - 大域的な構造を仮定しない
    - どんな場合にも使える
  - 雑音に弱い（大域構造を用いた平滑化ができない）
  - 次元の呪い