

知的情報処理 3. 最近傍法

櫻井彰人
慶應義塾大学理工学部

本日の目的

- 「k-最近傍法」or「最近傍法」とは
- 機械学習の最も基本なアイデア
 - 事例をたくさん記憶する
 - 必要になったら、事例そのものか、近い事例を思い出す。
- もう少し具体的には
 - 事例(入力 x_i と出力 y_i の対)を記憶する
 - 問合せ(入力 x に対する出力 y は何?)に対し、
 - 答える(x に近い事例を参照して、最適な y を作る)
 - そして
- その基本と問題(実に一般的な問題です)

R の基礎勉強もします

目次

- 丸暗記(?)
- 作ってみよう(ぴったり一致するもの)
 - 簡単なプログラム例
- 少し発展させよう(ぴったりでなくとも)
 - 近さとは
 - 次元と単位と距離
 - どの事例を参考に
 - 複数個と多数決
 - プログラム例

丸暗記

- 考えてみよう。
- 最近傍法に必要なものは?
- 答え:
- 教えられたことを全部、間違いなく覚えること、すなわち、丸暗記
- 問合せがあったら、記憶の中から、最も似たものを探し出す。



丸暗記 … オウムのよう



http://teflworldwiki.com/index.php/Rote_Learning

<http://sozaishu.seesaa.net/article/115935835.html>

丸暗記学習するプログラム

- 丸暗記すること自体は、コンピュータにとって難しいことではない(そのようなプログラムを書くことは、難しいことではない)
- しかし、
 - 思い出すときに、速くできるようにする、
 - データが大量でも対応できるようにするのは、容易ではない。
- 今回は、これらの問題は考えない(つまり、基本技術だけ考える)

こんなアプリ



目次

- 丸暗記(?)
- 作ってみよう(ぴったり一致するもの)
 - 簡単なプログラム例
- 少し発展させよう(ぴったりでなくとも)
 - 近さとは
 - 次元と単位と距離
 - どの事例を参考に
 - 複数個と多数決
 - プログラム例

作ってみよう

整理しよう			
穴埋め1	穴埋め2	穴埋め3	穴埋め4
1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16
17	18	19	20
21	22	23	24
25	26	27	28
29	30	31	32
33	34	35	36
37	38	39	40
41	42	43	44
45	46	47	48
49	50	51	52
53	54	55	56
57	58	59	60
61	62	63	64
65	66	67	68
69	70	71	72
73	74	75	76
77	78	79	80
81	82	83	84
85	86	87	88
89	90	91	92
93	94	95	96
97	98	99	100
101	102	103	104
105	106	107	108
109	110	111	112
113	114	115	116
117	118	119	120
121	122	123	124
125	126	127	128
129	130	131	132
133	134	135	136
137	138	139	140
141	142	143	144
145	146	147	148
149	150	151	152
153	154	155	156
157	158	159	160
161	162	163	164
165	166	167	168
169	170	171	172
173	174	175	176
177	178	179	180
181	182	183	184
185	186	187	188
189	190	191	192
193	194	195	196
197	198	199	200
201	202	203	204
205	206	207	208
209	210	211	212
213	214	215	216
217	218	219	220
221	222	223	224
225	226	227	228
229	230	231	232
233	234	235	236
237	238	239	240
241	242	243	244
245	246	247	248
249	250	251	252
253	254	255	256
257	258	259	260
261	262	263	264
265	266	267	268
269	270	271	272
273	274	275	276
277	278	279	280
281	282	283	284
285	286	287	288
289	290	291	292
293	294	295	296
297	298	299	300
301	302	303	304
305	306	307	308
309	310	311	312
313	314	315	316
317	318	319	320
321	322	323	324
325	326	327	328
329	330	331	332
333	334	335	336
337	338	339	340
341	342	343	344
345	346	347	348
349	350	351	352
353	354	355	356
357	358	359	360
361	362	363	364
365	366	367	368
369	370	371	372
373	374	375	376
377	378	379	380
381	382	383	384
385	386	387	388
389	390	391	392
393	394	395	396
397	398	399	400
401	402	403	404
405	406	407	408
409	410	411	412
413	414	415	416
417	418	419	420
421	422	423	424
425	426	427	428
429	430	431	432
433	434	435	436
437	438	439	440
441	442	443	444
445	446	447	448
449	450	451	452
453	454	455	456
457	458	459	460
461	462	463	464
465	466	467	468
469	470	471	472
473	474	475	476
477	478	479	480
481	482	483	484
485	486	487	488
489	490	491	492
493	494	495	496
497	498	499	500
501	502	503	504
505	506	507	508
509	510	511	512
513	514	515	516
517	518	519	520
521	522	523	524
525	526	527	528
529	530	531	532
533	534	535	536
537	538	539	540
541	542	543	544
545	546	547	548
549	550	551	552
553	554	555	556
557	558	559	560
561	562	563	564
565	566	567	568
569	570	571	572
573	574	575	576
577	578	579	580
581	582	583	584
585	586	587	588
589	590	591	592
593	594	595	596
597	598	599	600
601	602	603	604
605	606	607	608
609	610	611	612
613	614	615	616
617	618	619	620
621	622	623	624
625	626	627	628
629	630	631	632
633	634	635	636
637	638	639	640
641	642	643	644
645	646	647	648
649	650	651	652
653	654	655	656
657	658	659	660
661	662	663	664
665	666	667	668
669	670	671	672
673	674	675	676
677	678	679	680
681	682	683	684
685	686	687	688
689	690	691	692
693	694	695	696
697	698	699	700
701	702	703	704
705	706	707	708
709	710	711	712
713	714	715	716
717	718	719	720
721	722	723	724
725	726	727	728
729	730	731	732
733	734	735	736
737	738	739	740
741	742	743	744
745	746	747	748
749	750	751	752
753	754	755	756
757	758	759	760
761	762	763	764
765	766	767	768
769	770	771	772
773	774	775	776
777	778	779	780
781	782	783	784
785	786	787	788
789	790	791	792
793	794	795	796
797	798	799	800
801	802	803	804
805	806	807	808
809	810	811	812
813	814	815	816
817	818	819	820
821	822	823	824
825	826	827	828
829	830	831	832
833	834	835	836
837	838	839	840
841	842	843	844
845	846	847	848
849	850	851	852
853	854	855	856
857	858	859	860
861	862	863	864
865	866	867	868
869	870	871	872
873	874	875	876
877	878	879	880
881	882	883	884
885	886	887	888
889	890	891	892
893	894	895	896
897	898	899	900
901	902	903	904
905	906	907	908
909	910	911	912
913	914	915	916
917	918	919	920
921	922	923	924
925	926	927	928
929	930	931	932
933	934	935	936
937	938	939	940
941	942	943	944
945	946	947	948
949	950	951	952
953	954	955	956
957	958	959	960
961	962	963	964
965	966	967	968
969	970	971	972
973	974	975	976
977	978	979	980
981	982	983	984
985	986	987	988
989	990	991	992
993	994	995	996
997	998	999	1000

例1: 結果を記憶する

```
setwd("D:/R/Sample")
x <- read.table("cwlist.txt", as.is=1)
y <- "縦横無尽"

s <- c()
for ( i in 1:length(x[,1]) ) {
  score <- 0
  for ( j in 1:4 ) {
    if ( substr(x[i],j,j) == substr(y,j,j) ) score <- score +1
  }
  if ( score >= 3 ) s <- c(s,i)
}
print( s ); print( x[s,] )
```

これはskipしましょう

例2: Rの機能活用

```
setwd("D:/R/Sample")
x <- read.table("cwlist.txt", as.is=1)
y <- "縦横無〇"

s <- c()
for ( i in 1:length(x[,1]) ) {
  if ( length(agrep( y, x[i,] )) != 0 ) s <- c(s,i)
}
print( s ); print( x[s,] )
```

第一引数の文字列にほぼ等しい(defaultでは一文字間違いまで)文字列が引数2にあるとき、第二引数での位置を返す。ないとき、長さ0のベクトルを返す。

出力

[1] 1219

[1] "縦横無尽"

これは学習か？

- 単なるプログラム！
 - 否定するわけではないが
- 学習の基本であることは確かである。
人間でも、
 - 文字を覚える
 - 数を覚える
 - 九九を覚える、
- このプログラムでは、学習らしくない。
- しかし、後で使いやすいように、データの記憶方法を工夫をすると、学習らしくなる。
 - つまり、記憶するにあたって、データの性質を(データの内なる構造)を利用する。これこそ、学習である。

目次

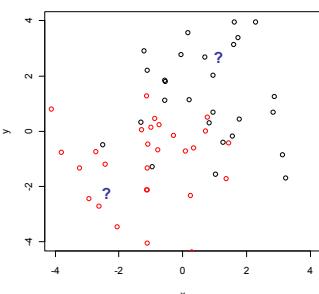
- 丸暗記(?)
- 作ってみよう(ぴったり一致するもの)
 - 簡単なプログラム例
- 少し発展させよう(ぴったりでなくとも)
 - 近さとは
 - 次元と単位と距離
 - どの事例を参考に
 - 複数個と多数決
 - プログラム例

少し発展させよう

- 右の問題で、穴埋めする時
- 説明変数値が
 - 知っているものと同じか
 - かなり近いときに、
被説明変数の値を答えにする
という方法を考えてみよう



図で見ると分りやすい



目次

- 丸暗記(?)
- 作ってみよう(ぴったり一致するもの)
 - 簡単なプログラム例
- 少し発展させよう(ぴったりでなくとも)
 - 近さとは
 - 次元と単位と距離
 - どの事例を参考に
 - 複数個と多数決
 - プログラム例

小目次

- 近さとは
 - 定義することの困難さ
 - 人間は、似ているものの発見が得意
 - しかし、言葉にできない
 - 「距離」でいいのか？
 - 単位の問題
 - 次元の問題
 - 正規化で解決

「似ている」の困難さ



http://gigazine.net/news/20071006_cat_similar_baby/

なかなか難しい



脱線: 逆手にとって: CAPTCHA

Completely Automated Public Turing Test To Tell Computers and Humans Apart



この画像の中に表示されている文字を入力してください

画像: D5B0382U
8文字
*文字の入力: _____

<http://www.captcha.net/>

似ているとは近いこと(?)

- 似ているとは近いことと、考えよう
- 近いとは「距離」が小さいこと。つまり、「近さ」とは「距離」である。
 - 同語反復。でもない。
 - 「距離」というと、数学では、距離の公理を満たす概念。
 - 距離の公理とは？
 - 非負性: $d(x,y) \geq 0$
 - 同一性: $x=y \Leftrightarrow d(x,y)=0$
 - 対称性: $d(x,y)=d(y,x)$
 - 三角不等式: $d(x,y)+d(y,z) \geq d(x,z)$
 - 代表的なものは:
 - ユークリッド距離: $\sqrt{(x_1-y_1)^2+(x_2-y_2)^2+(x_3-y_3)^2}$
 - マンハッタン距離: $|x_1-y_1|+|x_2-y_2|+|x_3-y_3|$

小目次

- 近さとは
 - 定義することの困難さ
 - 人間は、似ているものの発見が得意
 - しかし、言葉にできない
 - 「距離」でいいのか？
 - 単位の問題
 - 次元の問題
 - 正規化で解決

「距離」でいいのか？

- 例として、身長と体重で、AさんとBさんの近さをみると
ことにしよう
 - Aさんは (h_A, w_A) , Bさんは (h_B, w_B) としよう
- 距離は？
 - $\sqrt{((h_A-h_B)^2+(w_A-w_B)^2)}$? or $|h_A-h_B|+|w_A-w_B|$? or ...
- そもそも単位は？
 - 身長は、cm? mm? μ m? m? km?
 - 体重は、kg? g? mg? pg? t?

閉話休題。
△の横棒がないけどおかしくはない。そもそもその横棒って何だ？

単位の問題とは？

- 身長と体重を2個の属性とし、次の三人の距離をみよう
 - A: (170cm, 65kg)
 - B: (180cm, 60kg)
 - C: (175cm, 70kg)
- マンハッタン距離を考えよう
 - $d(A,B)=10+5=15$
 - $d(B,C)=5+10=15$
 - $d(C,A)=5+5=10$
- cmとkgを用いるのは不公平。単位の名称からいって、mとgだろう。となると、
 - $d(A,B)=0.01+5000=5000.01$
 - $d(B,C)=0.005+10000=10000.005$
 - $d(C,A)=0.005+5000=5000.005$
- 他にも、いくらでも、考えられる。さて、どうしたものか。

え？ そう？

単位の問題だけではない

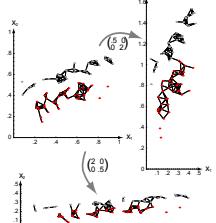
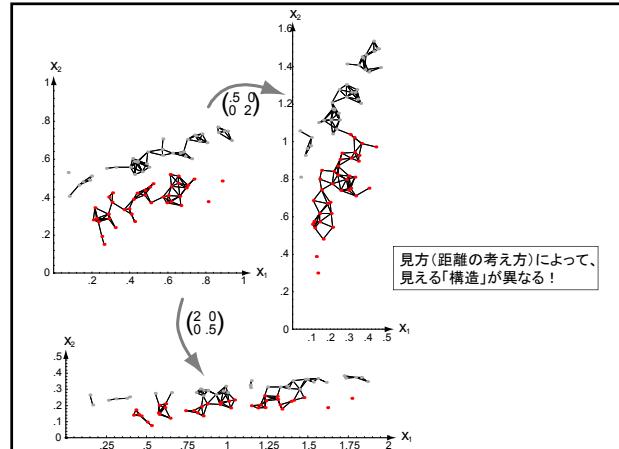


FIGURE 10.8. Scaling axes affects the clustering in a minimum distance cluster method. The original data and minimum distance clusters are shown in the upper left; points in one cluster are shown in red, and the others are shown in gray. When the vertical axis is expanded by a factor of 0.5 and the horizontal axis is expanded by a factor of 2.0, the clustering is altered (as shown in the middle). 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 are formed (as shown in the bottom). In both cases, the arrangement 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.



つまり: 次元と単位

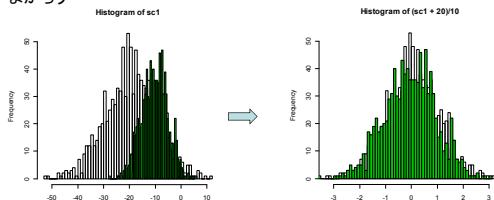
- 問題は2つある: 次元と単位
- 次元: 異なる物理量を区別する概念。次元が異なる物理量は比較さえできない。
 - 「1kgと1mとは、どちらがより偉いか？」は有意味か
 - なお、この例に表れる次元は3個。重さ、長さ、偉さ。
 - 「偉さ」が物理量かというと、まあ、そうではないが。
- 単位: 物理量を数値で表すとき、数値の1に対応する物理量
 - 長さの単位の例: m, mile, 尺, Å
 - 勿論、1cmを単位にしても、3.14kmを単位にしてもよい

次元と単位と距離

- 次元解析: 物理量を、質量、長さ、時間、電荷、温度などの次元の有理数幕で表現し、方程式や仮説の妥当性を調べること
- 目下の話題は、「次元」
 - 異なる次元の量は、比較・加算ができない。
 - 当然、(170cm, 65kg)と(180cm, 60kg)の距離をユークリッド距離的に、また、マンハッタン距離的には定義できない。
 - では、どうする？
- もう一つの話題は、「単位」
 - 異なる単位の量は、比較・加算ができない。
 - 身長と指の長さが属性のとき、(1.7m, 10cm)と(1.8m, 12cm)の距離をユークリッド距離的に、また、マンハッタン距離的には定義できない。
 - では、どうする？

正規化

- もしデータが大量にあれば、そして、各属性値が独立に正規分布していれば(ある確率分布に従っていれば)、「値/標準偏差」という量を使えばよい
 - この量は、無次元量。なぜなら、「値」もその「標準偏差」も同じ次元。
 - 「無次元量だから比較・加算できる」という説ではないが、
 - 無次元の上に、各属性値の標準偏差が1に規格化されたのだから、よからう

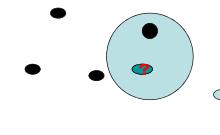


目次

- 丸暗記(?)
- 作ってみよう(ぴったり一致するもの)
 - 簡単なプログラム例
- 少し発展させよう(ぴったりでなくとも)
 - 近さとは
 - 次元と単位と距離
 - どの事例を参考に
 - 複数個と多数決
 - プログラム例

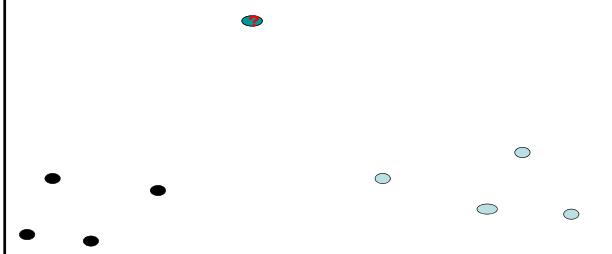
どの事例を参考に？

- 「最も近い事例」にしよう。そうすれば、
- 「どのくらい近ければよい」という基準を考えなくてすむ。
 - つまり、(適当に決めた基準で)見つからなかった場合にも対処できる



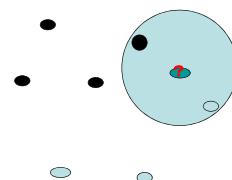
最も近いものが遠いと？

- 「どのくらい近ければよい」という基準を考えなくてすむ。」とはいえ



近い事例が複数あると？

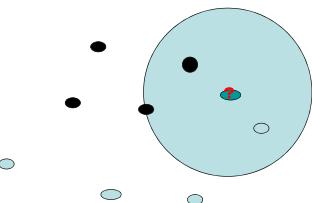
- 「一つ」に決めてしまったが、ちょっと不安。
- 例えば、



– どっち？

多数決はどうか？

- 近い方から3人(とか5人とか7人とか...)の意見を聞くことにしたらどうだろう。
- そして多数決をとればよいだろう
- 一般には、近くの K 人から聞くことにする、この方法を k -nearest neighbor 法という



目次

- 丸暗記(?)
- 作ってみよう(ぴったり一致するもの)
 - 簡単なプログラム例
- 少し発展させよう(ぴったりでなくとも)
 - 近さとは
 - 次元と単位と距離
 - どの事例を参考に
 - 複数個と多数決
 - プログラム例

深みに: 多数決のプログラムは?

- まず、「多数」を選ばないといけない。
 - プログラムはちょっと複雑
 - 例えば、集まった100人のうち、身長が最も低い5人を選ぶ方法を考えることになる。
- どうすればいい?
- 一つの方法は、昔とった杵柄(のはず)のバブルソートです。
 - 100人をバブルソートする途中で、つまり、上位5人が決まった時点で、バブルソートを中止すればよい。

深みに: 多数決のプログラムは? (続)

- しかし、長いプログラムを書くのはやさることにしよう。
 - その代わりプログラム実行時間はかかるようになるのだが、まあ、我慢。
- テストする x との距離を、すべての学習データに対して計算し、それを、全部(ここが無駄)ソートしてしまい(R内の関数を使えばプログラムは書かなくてすむので)、それから先頭の必要個数(今回は k 個)を使えばよい。
- 朗報! R で sort の help を見てください。部分的にソートしてくれそうです。partial=5 とすると、最小値5個が得られたところで、ソートは終了するようです。
- しかし、index(何番目の要素かを示す値)は返してくれない。返すように指定する(index.return=TRUE)のは完全ソートのときのみ可能
- partial は諦めて、index.return=TRUE としよう。
- なお、rank という関数もある。

実プログラム例: 2次元の場合

- 2次元かつ 0 or 1 に分類する場合の k-NN プログラムを作ってみよう。
- 入力の近さ判定にはユークリッド距離、出力の近似には、多数決を使う
- 実験用のデータとして、平均が (1,1) と (-1,-1)、分散共分散行列が $1.5^2 I$ (I は単位行列) である2個の正規分布を考える。二つのクラスの事例は、それぞれの正規分布に従い生成されると考える。
- 何個かデータを生成する

実験用データの作成

```
class1<-data.frame(x=rnorm(30,mean=1,sd=1.5),y=rnorm(30,mean=1,sd=1.5),c=rep(1,30))
class2<-data.frame(x=rnorm(30,mean=-1,sd=1.5),y=rnorm(30,mean=-1,sd=1.5),c=rep(2,30))
data<-rbind(class1,class2)

plot(subset(data,data$c==1)[,1:2],xlim=c(-4,4),ylim=c(-4,4))
points(subset(data,data$c==2)[,1:2],col="red",xlim=c(-4,4),ylim=c(-4,4))

> head(class1)
   x     y c
1 1.5121442 -1.721736 1
2 -0.3993179  1.716194 1
3 -0.6811447 -1.406001 1
4 -0.1252901  1.114100 1
5  0.5958607  3.326750 1
6  2.2467210  3.636144 1
> head(class2)
   x     y c
1 -1.1141637 -3.1630000 2
2 -0.2754737  0.08535921 2
3 -0.1761440  1.38123139 2
4 -0.9974249 -1.31036347 2
5 -0.4343134 -4.01635416 2
6  1.9064339  0.74648529 2
```

何をやっているか

```
c という名の列に1を30個
class1<-data.frame(x=rnorm(30,mean=1,sd=1.5),y=rnorm(30,mean=1,sd=1.5),c=rep(1,30))
データフレームにするあたり x といふ名の列に平均1、標準偏差1.5の正規乱数30個
y といふ名の列に平均1、標準偏差1.5の正規乱数30個

data<-rbind(class1,class2) 行方向につなげる。 subset(data,data$c==1) は
data の中で data$c==1 が成立しているものだけからなる部分集合
plot(subset(data,data$c==1)[,1:2],xlim=c(-4,4),ylim=c(-4,4))
points(subset(data,data$c==2)[,1:2],col="red",xlim=c(-4,4),ylim=c(-4,4))

> head(class1)
   x     y c
1 1.5121442 -1.721736 1
2 -0.3993179  1.716194 1
3 -0.6811447 -1.406001 1
4 -0.1252901  1.114100 1
5  0.5958607  3.326750 1
6  2.2467210  3.636144 1
> tail(head(data,n=33),n=6)
   x     y c
28 -0.6473127  1.35403476 1
29  0.5357084  4.11995931 1
30  1.9639820  0.26578581 1
31 -1.1141637 -3.16300000 2
32 -0.2754737  0.08535921 2
33 -0.1761440  1.38123139 2
```

多数決はどうしよう？

- table() 関数を用いて、数えることにする。
 - table() は分割表 (contingency table, cross tabulation, or cross tab) を作る関数

```

> x <- c(0,1,0,1,1,0,0,0)
> (tbl<-table(x))
          0 が5個, 1が3個
> as.data.frame(tbl)
  x Freq
1 0    5
2 1    3
> (count1<-as.data.frame(tbl)$Freq[as.data.frame(tbl)$x==1])
[1] 5
> (count2<-as.data.frame(tbl)$Freq[as.data.frame(tbl)$x==2])
[1] 3
>

```

多数決

- 頻度を求め、その名前 (factor の level) からクラスを表す数値を取り出したい。

```

# majority(c(0,1,2,2,0,1,0,2,1,2,0,1,1)) = 1 となるような関数
majority <- function( x ) {
  return( as.numeric(names(which.max(table(x)))) )
}

```

まとめれば k-NN

多数決関数の定義

```

# majority(c(0,1,2,2,0,1,0,2,1,2,0,1,1)) = 1 となるような関数
majority <- function( x ) {
  return( as.numeric(names(which.max(table(x)))) )
}

```

実験用データの作成

```

class1<-data.frame(x1=rnorm(30,mean=1, sd=1.5),
                     x2=rnorm(30,mean=1, sd=1.5), c=rep(1,30))
class2<-data.frame(x1=rnorm(30,mean=-1, sd=1.5),
                     x2=rnorm(30,mean=-1, sd=1.5), c=rep(2,30))
data<-rbind(class1, class2)

```

Main program

```

# k-NN majority
k <- 3
x1<- 0      # テスト点の x1 座標
x2<- 0      # テスト点の x2 座標
sorted<- sort( ( x1 - data$x1 )^2 + ( x2 - data$x2 )^2, index.return=TRUE)
majority( data$c[ sorted$ix[ 1:k ] ] )

```

注意: これはスカラー、これはベクトル、引算の結果はベクトル
ベクトルの要素の2乗からなるベクトル

境界線が表示できます

```

# k of k-NN
k <- 3
# test data which, in this case, are used for the plot
xtest1 <- seq(min(data$x1),max(data$x1),length=30)
xtest2 <- seq(min(data$x2),max(data$x2),length=30)
xtest <- expand.grid(xtest1,xtest2)
# use k-NN to get predictions on the grid
cPredicted <- rep(0,30*30)
for ( i in 1:length(xtest[,1]) ) {
  x1 <- xtest$Var1[i]
  x2 <- xtest$Var2[i]
  sorted <- sort( ( x1 - data$x1 )^2 + ( x2 - data$x2 )^2, index.return=TRUE)
  cPredicted[i] <- majority( data$c[ sorted$ix[ 1:k ] ] )
}
# cPredicted in a matrix form to be used for "plot"
cPredictedMatrix <- matrix( cPredicted, 30, 30 )

```

何を行おうとしているか？へのヒント

```

plot(subset(data, data$c==1)[,1:2], xlim=c(-4,4), ylim=c(-4,4))
points(subset(data, data$c==2)[,1:2], col="red", xlim=c(-4,4), ylim=c(-4,4))
points(xtest, pch=".", col="blue")

```

プログラム(続)

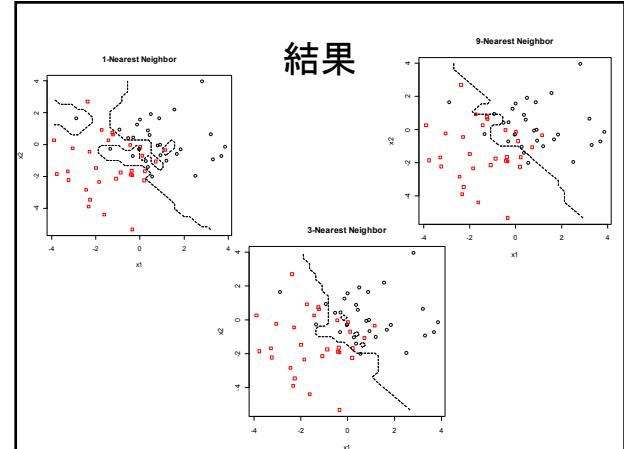
```

# plot "data"
plot(data[,1:2],
      xlab="x1", ylab="x2",           # labels of axes
      main=paste(k, "-Nearest Neighbor", sep=""), # title of the graph
      col=data$c, pch=as.real(data$c)+20) # color and characters to be used

# draw boundaries
for(i in 1:max(cPredicted))
  contour(
    xtest1, xtest2, cPredictedMatrix==i, # x, y, and z where z is 0 or 1
    levels=0.5,                         # levels=0.5 i.e. the middle of 0 and 1
    drawlabels=FALSE, add=TRUE,
    col=1, lty=3)                      # color is black and line type is broken

```

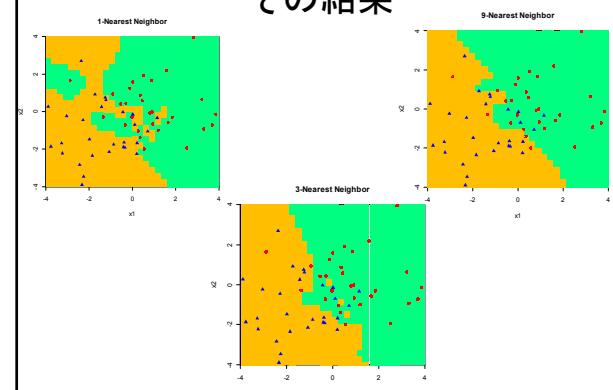
結果



別のプロット

```
# another way to plot
# the following is for just two classes
image( xtest1, xtest2, cPredictedMatrix==2,
       col=c("#00FF80", "#FFBF00"), axes=TRUE,
       xlim=c(-4,4), ylim=c(-4,4),
       xlab="x1", ylab="x2", main=paste(k, "-Nearest Neighbor", sep=""))
points( data[,1:2], col=ifelse(data$c==1, "red", "blue"),
       pch=ifelse(data$c==1, 16, 17) )
```

その結果



まとめにかえて(後半)

目次

- どの「穴埋め」か?
- k-NN (k nearest neighbor, k 最近傍法)とは
- なぜ「境界」がでこぼこか?
- k はどう決めるか

解いた問題

- 穴埋め問題1
 - 穴埋め問題2-3
 - 説明変数値が
 - ・ 知っているものと同じか
 - ・ かなり近いときに、
- 被説明変数の値を答えにする
という方法



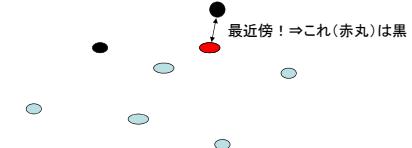
目次

- どの「穴埋め」か?
- k-NN (k nearest neighbor, k 最近傍法)とは
- なぜ「境界」がでこぼこか?
- k はどう決めるか

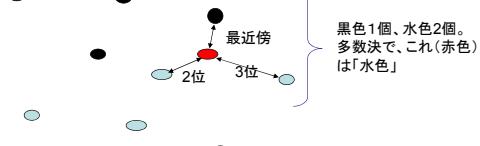
最近傍法

- 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 (fを近似)する
- k-Nearest neighbor
 - k個の最近接データの間で、多数決、または
 - k個の最近接データの間で、平均値、または、..

1-Nearest Neighbor



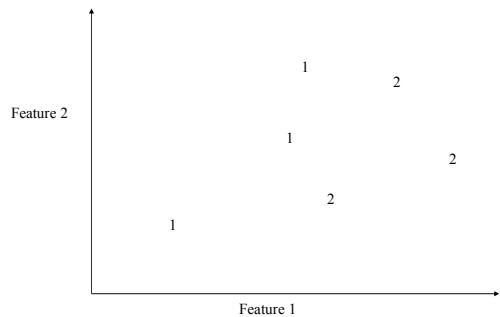
3-Nearest Neighbor



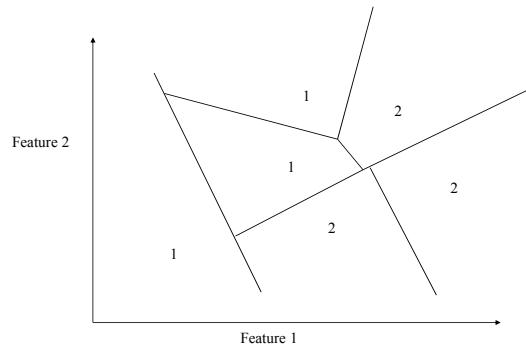
目次

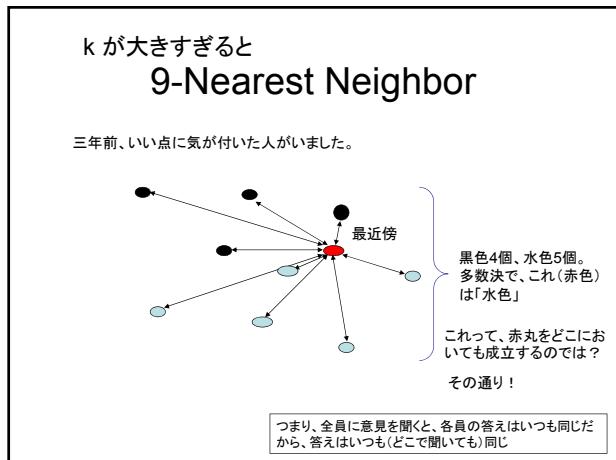
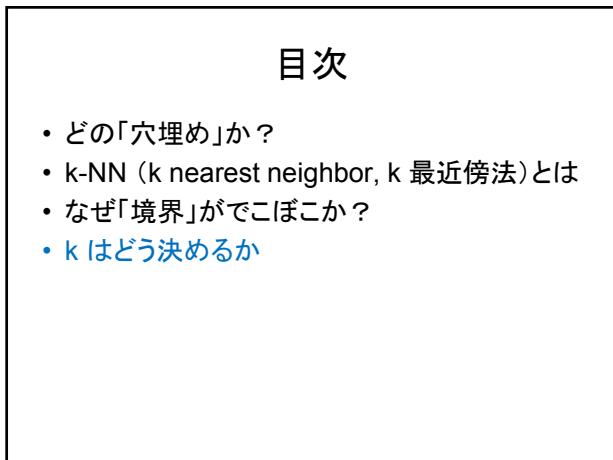
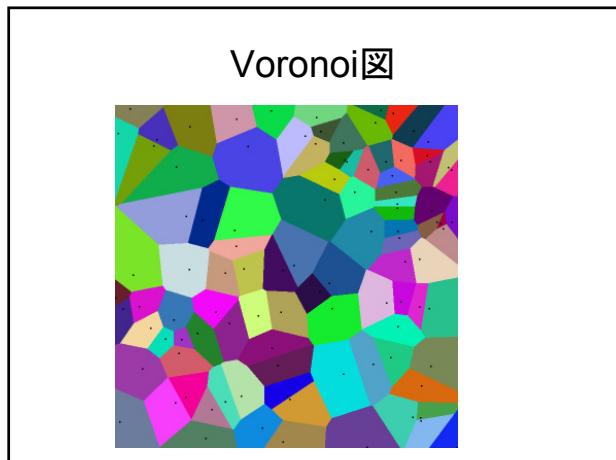
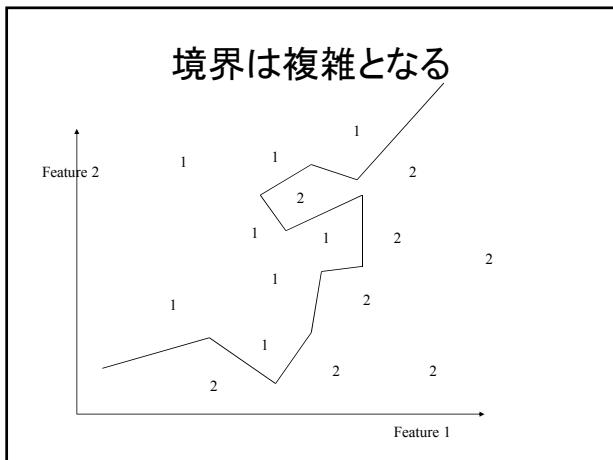
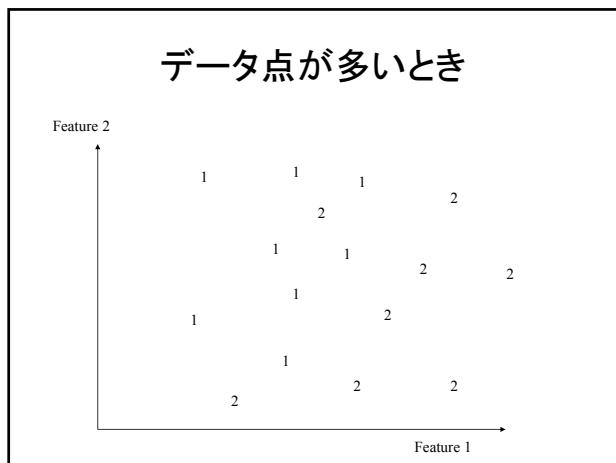
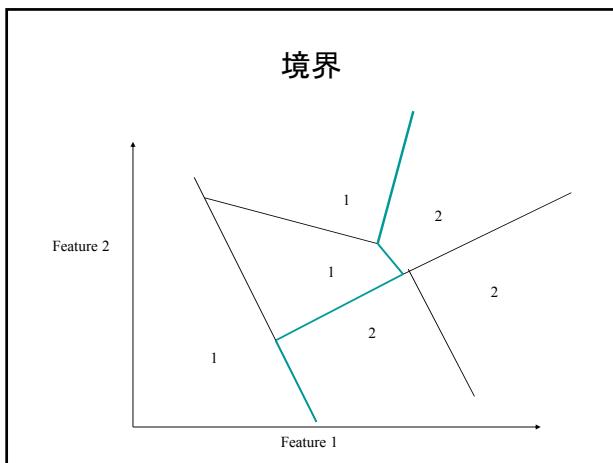
- どの「穴埋め」か？
- k-NN (k nearest neighbor, k 最近傍法) とは
- なぜ「境界」がでこぼこか？
- k はどう決めるか

1-NNの境界線の幾何的解釈



境界





k の選択

- こうしたパラメータの設定(この講義でもこれから頻繁に出てくる)は、一般に、難しい
- 理論的根拠をもって、または、データ(validation data set)を用いて決めることがある
- k-NNについては、bootstrapによる方法が提案されている

Peter Hall, Byeong U. Park, and Richard J. Samworth.
Choice of neighbor order in nearest-neighbor classification.
The Annals of Statistics, 2008, Vol. 36, No. 5, 2135–2152.

即レポ2

練習問題

- この講義で作成した k-NN を用いて、データもこの講義で作成したものを用いて、k を大きくしたとき、境界は、(1,1) と (-1,-1) の垂直2等分線に近くなるか？
 - k を変えてみるだけでなく、データ数も変えてみよ。
 - なぜ「この特別な2点の垂直2等分線」なのだろうか、その理由を考えてみてください。

—昨年、全然垂直2等分線らしくないのに、「垂直2等分線になりました」とレポートに書いた人がいました。これは「白を黒という」に等しい。