Chapter 12 –
Discriminant Analysis 判别分析

Instructor: Zach Zhizhong ZHOU,
Shanghai Jiao Tong University
主讲教师：周志中，上海交通大学

Data Mining for Business Intelligence

Shmueli, Patel & Bruce

Discriminant Analysis: Background
判别分析：背景

A classical statistical technique 经典统计技术

Used for classification long before data mining 数据挖掘出现前用了很久的分类 方法

- Classifying organisms into species 将生物分类到不同种属中
- Classifying skulls 对颅骨进行分类
- Fingerprint analysis 指纹分析

And also used for business data mining (loans, customer types, etc.) 也可以用于商业数据挖掘（如贷款、客户类别等）

Can also be used to highlight aspects that distinguish classes (profiling) 也可以用于突显用于区分不同类别的显著特征。（剖绘）

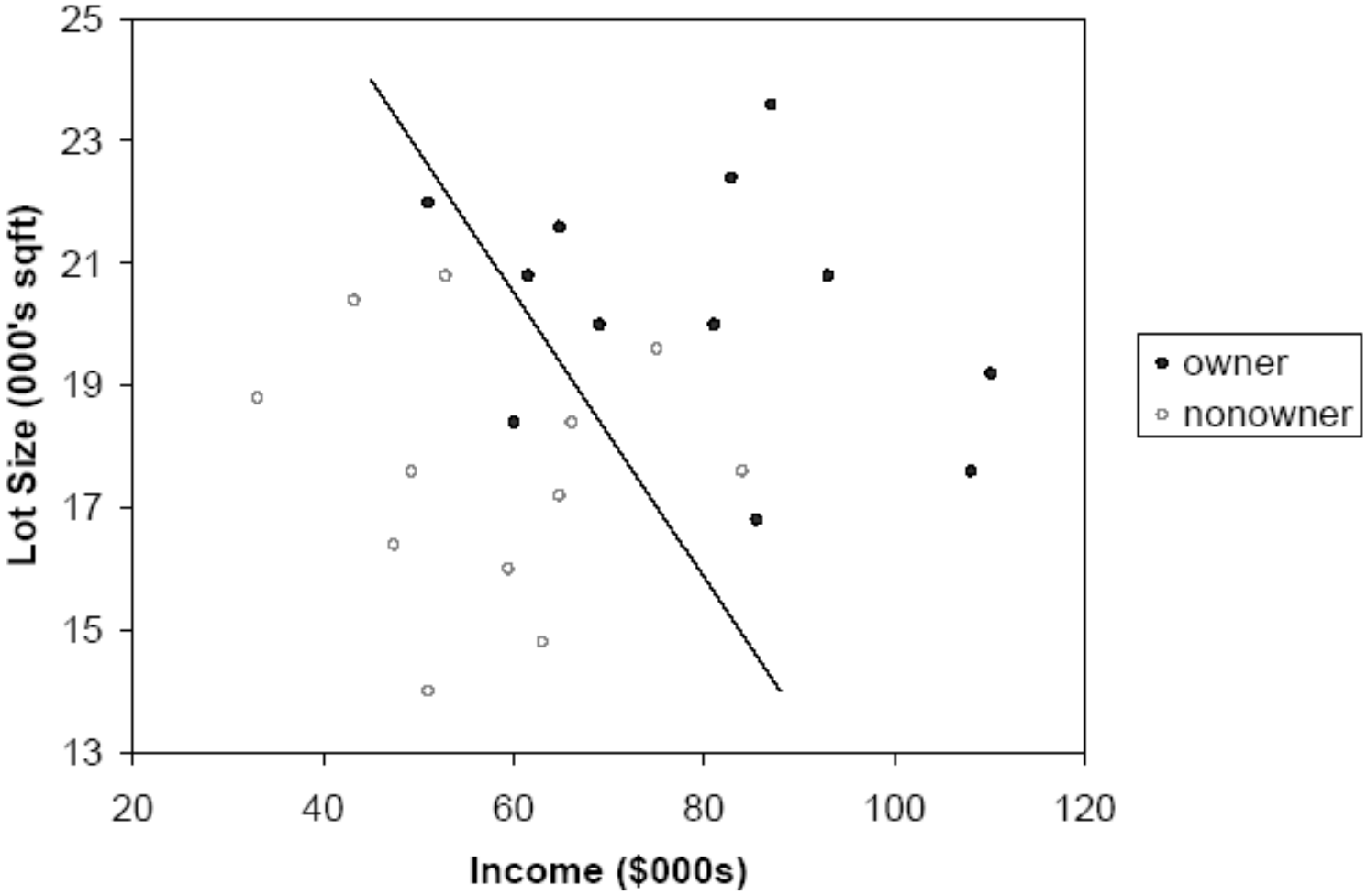
Small Example: Riding Mowers

Goal: classify purchase behavior (buy/no-buy) of riding mowers based on income and lot size

Outcome: owner or non-owner (0/1)

Predictors: lot size, income

Can we manually draw a line that separates owners from non-owners?



Example: Loan Acceptance
例子：接受贷款申请

In the prior small example, separation is clear. 上面的小例子中，如何分割是非常清楚的。

In data mining applications, there will be more records, more predictors, and less clear separation. 在数据挖掘应用中，会有更多的记录、预测因子和没那么清晰的分割。

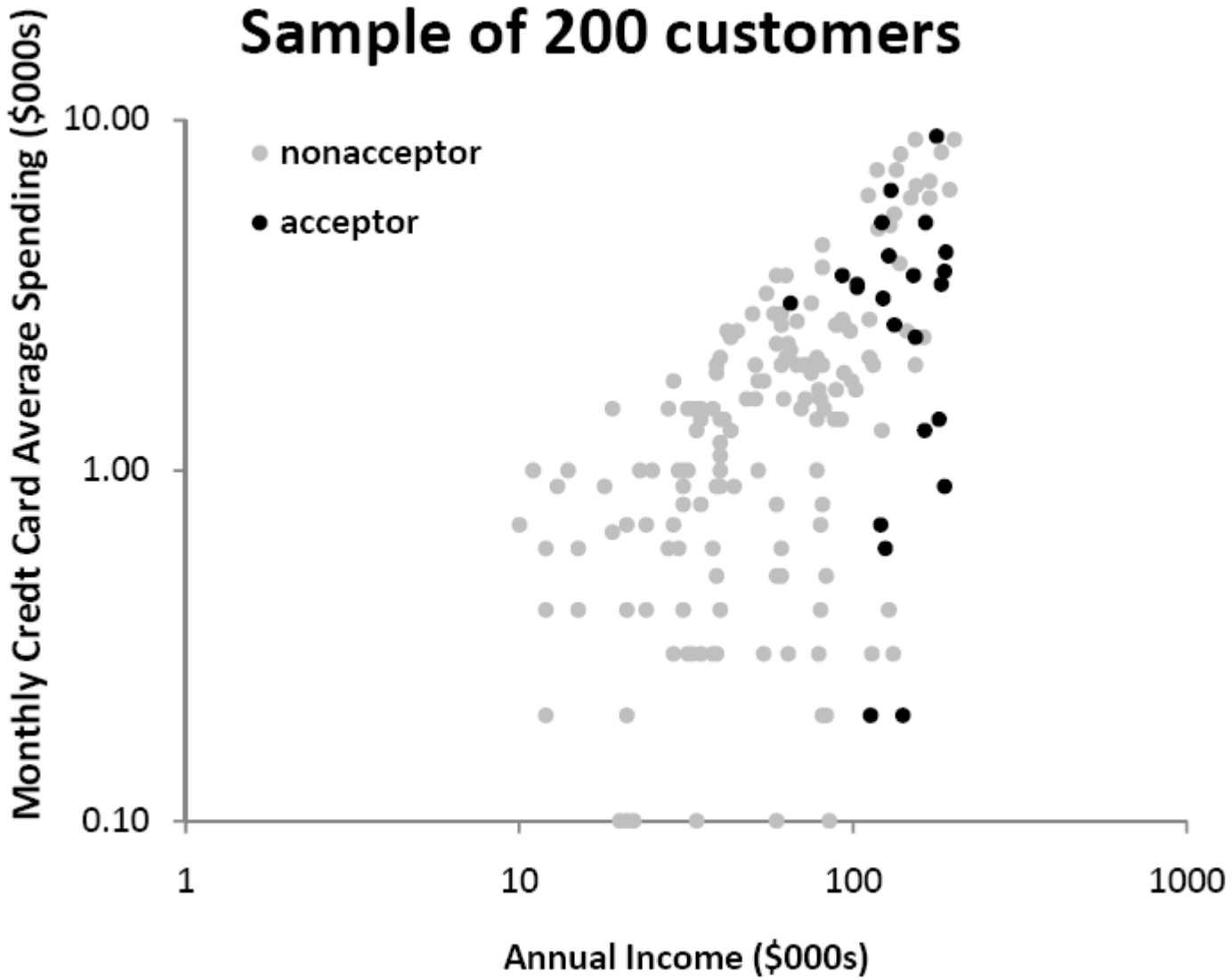
Consider Universal Bank example with only 2 predictors: 考虑全能银行（德国）的例子，假设只有2个预测因子：

Outcome: accept/don’t accept loan 结果：接受/不接受贷款申请

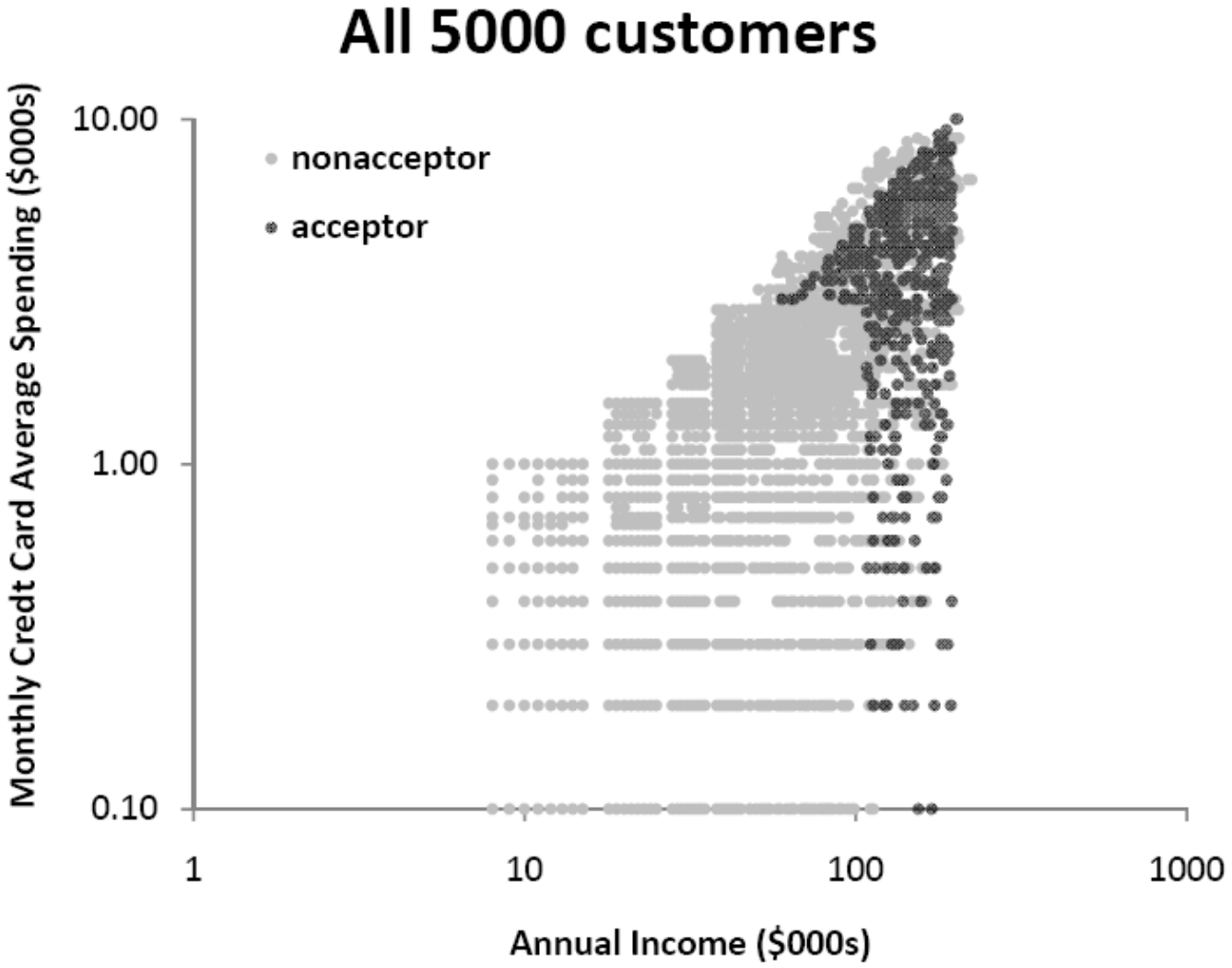
Predictors: 预测因子：

- Annual income (Income) 年收入
- Avg. monthly credit card spending (CCAvg) 每月信用卡消费支出

Sample of 200 customers



5000 customers



Algorithm for Discriminant Analysis

判别分析算法

The Idea 思路

To classify a new record, measure its distance from the center of each class

对一个新记录进行分类时，衡量他与每个类别的中心之间的距离

Then, classify the record to the closest class

然后把它归入距离最近的类别中

Step 1: Measuring Distance
第一步：衡量距离

Need to measure each record’s distance from the center of each class 需要衡量每个记录与每个类别中心之间的距离

The center of a class is called a *centroid* $\overline{X}$ 一个类别的中心被称为形心。

The centroid is simply a vector (list) of the means of each of the predictors. This mean is computed from all the records that belong to that class. 形心是一个由预测因子平均值组成的向量，平均值是属于该类别的记录的各个预测因子的平均值。

Step 1: Measuring Distance – cont.
衡量距离（续）

A popular distance metric is Euclidean Distance (used with KNN). We can use it to measure the distance of a record from a class centroid:

常用的距离指标是欧拉距离（用在KNN中）。可用来衡量一个记录与一个类别的距离。

$$D_{Euclidean}(\mathbf{x}, \bar{\mathbf{x}}) = \sqrt{(x_1 - \bar{x}_1)^2 + \cdots + (x_p - \bar{x}_1)^2}$$

Drawbacks: 缺点：

- Sensitive to scale, variance (can normalize to correct)

  对量纲和方差敏感，可通过标准化数据解决此问题。

- Ignores correlation between variables

  忽略了变量之间的相关性。

Instead, use “Statistical (Mahalanobis) Distance” 使用统计（马氏）距离

transpose (convert column to row)

$$D_{\text{Statistical}}(\mathbf{x}, \overline{\mathbf{x}}) = [\mathbf{x} - \overline{\mathbf{x}}]' S^{-1} [\mathbf{x} - \overline{\mathbf{x}}]$$

inverse of covariance matrix S 这个S矩阵是属于某个类别的记录生成的方差矩阵。中间是S矩阵的逆。

(p-dimension extension of division)

- For a single predictor (p=1), this reduces to a z-score 如果只有1个预测因子，那么马氏距离就是z-值。
- When p > 1, statistical distance takes account of correlations among predictors (z-score doesn’t) 如果预测因子个数大于1，马氏距离考虑了因子之间的相关性而z-值未予以考虑。

Step 2: Classification Functions
步骤2：分类函数

- The idea is to create classification score that reflects the distance from each class 思路是生成一个用来进行分类的分数，该分数体现了记录与各个分类之间的距离。
- This is done by estimating “*classification functions*”, which are a function of the statistical distances. 这可以通过估计“分类函数”实现。分类函数是一个统计距离的函数。
- The estimation maximizes the ratio of between-class to within-class variability 估计过程使异类与同类之间的差异最大。
- Fisher’s linear classification functions: one for each class. Used to compute a classification score. Fisher线性分类函数：每个类别对应一个函数，用于计算用于分类的分数。
- Classify a record to class with highest score 计算记录对应每个类别的分类函数，该记录归入分类函数取值最高的一类。

Classification Functions (XLMiner output)

Variables	Classification Function	
	owner	non-owner
Constant	-73.16020203	-51.42144394
Income	0.42958561	0.32935533
Lot_Size	5.46674967	4.68156528

record #1: income = $60K, lot size = 18.4K

Owner score = -73.16 + (0.43)(60) + (5.47)(18.4) = 53.2

Non-owner score= -51.42+(0.33)(60)+(4.68)(18.4)= 54.48

"Non-owner" score is higher → (mis)classify as non-owner

Classification scores for part of Mowers data

Row Id.	Predicted Class	Actual Class	Classification Scores	
			owners	nonowners
1	nonowner	owner	53.2031285	54.48067701
2	owner	owner	55.4107621	55.38873348
3	owner	owner	72.7587384	71.04259149
4	owner	owner	66.9677061	66.21046668
5	owner	owner	93.2290383	87.71741038
6	owner	owner	79.0987673	74.72663127
7	owner	owner	69.449838	66.54448063
8	owner	owner	84.8646791	80.71623966
9	owner	owner	65.8161985	64.93537943
10	owner	owner	80.4996528	76.58515957
11	owner	owner	69.0171568	68.37011405
12	owner	owner	70.9712258	68.88764339
13	owner	nonowner	66.2070123	65.0388853
14	nonowner	nonowner	63.2303113	63.34507531
15	nonowner	nonowner	48.7050398	50.44370426
16	nonowner	nonowner	56.9195896	58.31063803
17	owner	nonowner	59.1397834	58.63995271

Step 3: Converting to Probabilities
转化成概率值

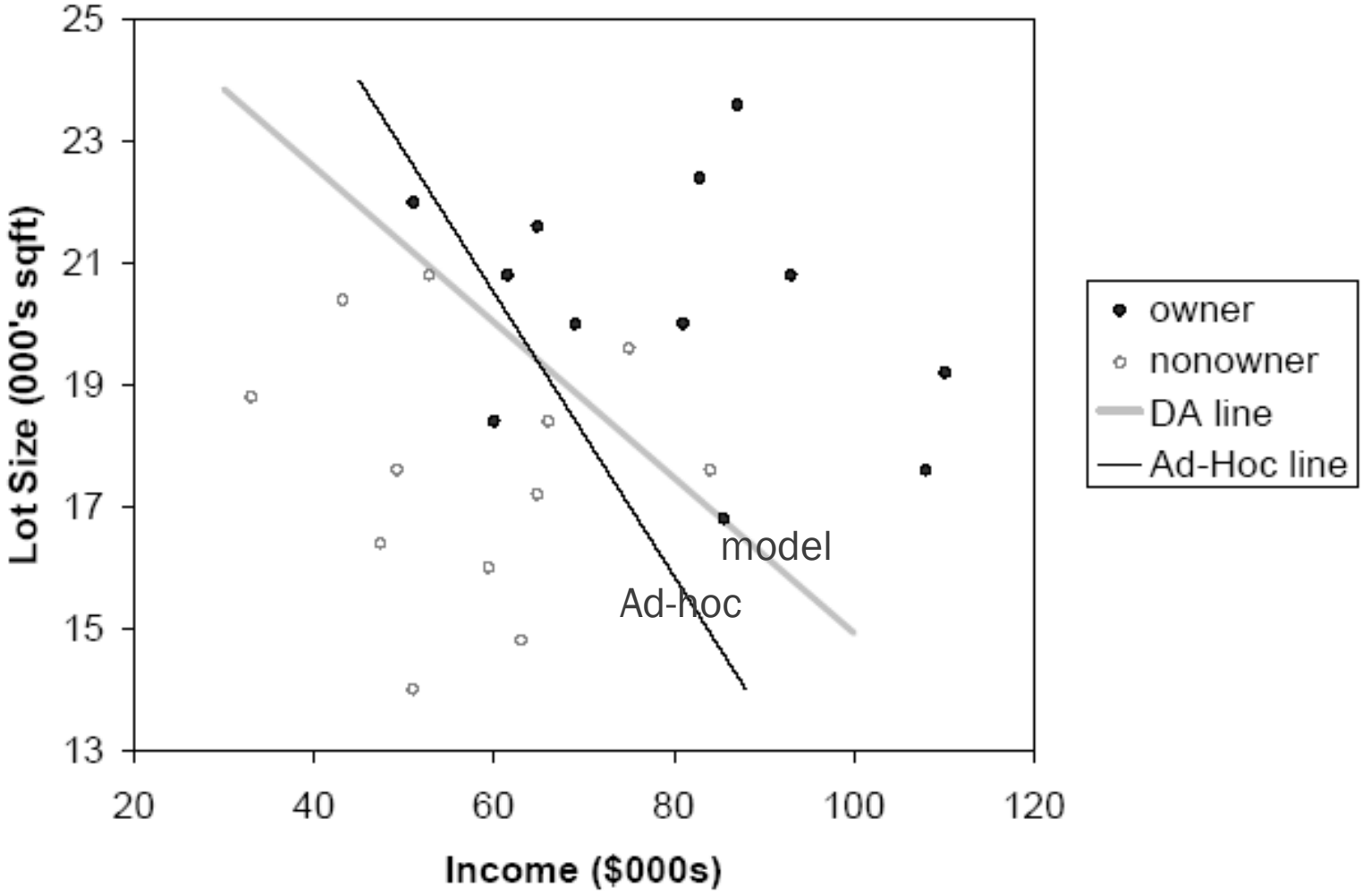
It is possible to convert classification scores to probabilities of belonging to a class: 我们可以将分类分数转化成属于某个类别的概率。

$$P = \frac{e^{c_k(i)}}{e^{c_1(i)} + e^{c_2(i)} + \cdots + e^{c_m(i)}}$$

Probability that record *i* (with predictor values *x1, x2, ⋯xp*) belongs to class *k*

The probability is then compared to the cutoff value in order to classify a record 可以将概率值与截值对比决定记录的分类。

Line from model (plus ad-hoc line)



Assumptions & Caveats of Discriminant Analysis
判别分析的假设和警示性说明

1. Assumes multivariate normality of predictors
   假设预测因子服从多项正态分布。

   When this condition is met, DA is more efficient than other methods (i.e. needs less data to obtain similar accuracy) 当这个假设满足，DA比其他方法更有效率（比如说，需要更少的数据来达到相同的正确率）

   Even when it is not met, DA is robust when we have enough cases in smallest class (> 20) . This means it *can be used with dummy variables*! 即便这个假设没有得到满足，DA仍然表现出鲁棒性（只要拥有最少记录的分类中有足够多的记录，比如大于20个）。这意味着DA可以用在有虚拟变量的情况下。

Assumptions & Caveats of Discriminant Analysis
判别分析的假设和警示性说明

2. Assumes correlation among predictors within a class is the same across all classes. (Compare correlation tables of each class by eye.) 假设预测因子之间的相关性在所有分类中都是相同的（这可以通过目测每个分类的相关系数矩阵判断该假设是否得到满足）。
3. Sensitive to outliers 对奇异值敏感。

Assessing Predictive Performance

As in other classification methods:

- Confusion matrix
- Lift

Based on validation data

Improving Classifications

Prior Probabilities 先验概率

If classes are not equally frequent, or their frequency in the sample does not reflect reality, then classification functions can be improved 如果类别不是以同样的频度出现，或者它们在样本中出现的频度与现实不同，那么分类函数可以进一步改善。

How? 如何改善?

Incorporate prior (or real) probabilities of class membership: 将某类成员出现的先验概率（或者真实概率）引入到模型中。

- Add $log(p_j)$ to the classification function for class j

  P_j is probability a case belongs to class j 在属于j的分类函数中加上$log(p_j)$。这里的P_j是记录属于分类j的概率。

Example - Mowers

Sample contains 50% owners, but suppose in population only 15% are owners (i.e. 0.15 probability of being an owner)

Existing classification function constants

- Owners: -73.16
- Nonowners: -51.42

Adjusted for prior probabilities:

- Owners: -75.06 + log(0.15) = -75.06
- Nonowners: -51.42 + log(0.85) = -50.58

Unequal Misclassification Costs
不同的分类错误成本

For the two-class (buyer/non-buyer) case, we can account for asymmetric costs of misclassification (C_1, C_2) in same fashion as for unequal prior probabilities

对于二元分类（购买/不购买）的情况，我们可以将不同分类错误成本按处理先验概率的方式加入模型。

How?

- Add log(C_1) and log (C_2) to constants在常数项加上log(C_1) 和 log (C_2)。

Often absolute costs are unknown. Instead, use cost ratio:

- Set C_1 = 1, C_2 = ratio 绝对成本往往未知，我们可以使用成本比率。
- Add log (C_2/C_1) to class 2’s constant 在类别2的常数项加上log (C_2/C_1) 。

Multiple Classes

Same procedure is used for multiple classes

One classification function for each class

Whichever function has highest value, case is assigned to that class

Example: Auto Accidents

Outcome: (3 classes)

- No injury
- Non-fatal injury
- Fatal injury

Predictors:

Time of day, day of week, weather, type of road, road surface conditions, ...

Accident Example: Data Sample

Accident #	RushHour	WRK_ZONE	WKDY	INT_HWY	LGTCON	LEVEL	SPD_LIM	SUR_COND	TRAF_WAY	WEATHER	MAX_SEV
1	1	0	1	1	dark_light	1	70	ice	one_way	adverse	no-injury
2	1	0	1	0	dark_light	0	70	ice	divided	adverse	no-injury
3	1	0	1	0	dark_light	0	65	ice	divided	adverse	non-fatal
4	1	0	1	0	dark_light	0	55	ice	two_way	not_adverse	non-fatal
5	1	0	0	0	dark_light	0	35	snow	one_way	adverse	no-injury
6	1	0	1	0	dark_light	1	35	wet	divided	adverse	no-injury
7	0	0	1	1	dark_light	1	70	wet	divided	adverse	non-fatal
8	0	0	1	0	dark_light	1	35	wet	two_way	adverse	no-injury
9	1	0	1	0	dark_light	0	25	wet	one_way	adverse	non-fatal
10	1	0	1	0	dark_light	0	35	wet	divided	adverse	non-fatal
11	1	0	1	0	dark_light	0	30	wet	divided	adverse	non-fatal
12	1	0	1	0	dark_light	0	60	wet	divided	not_adverse	no-injury
13	1	0	1	0	dark_light	0	40	wet	two_way	not_adverse	no-injury
14	0	0	1	0	day	1	65	dry	two_way	not_adverse	fatal
15	1	0	0	0	day	0	55	dry	two_way	not_adverse	fatal
16	1	0	1	0	day	0	55	dry	two_way	not_adverse	non-fatal
17	1	0	0	0	day	0	55	dry	two_way	not_adverse	non-fatal
18	0	0	1	0	dark	0	55	ice	two_way	not_adverse	no-injury
19	0	0	0	0	dark	0	50	ice	two_way	adverse	no-injury
20	0	0	0	0	dark	1	55	snow	divided	adverse	no-injury

Classification Functions

Variables	Classification Function		
	fatal	no-injury	non-fatal
Constant	-25.59584999	-24.51432228	-24.2336216
RushHour	0.92256236	1.95240343	1.9031992
WRK_ZONE	0.51786095	1.19506037	0.77056831
WKDY	4.78014898	6.41763353	6.11652184
INT_HWY	-1.84187829	-2.67303801	-2.53662229
LGTCON_day	3.70701218	3.66607523	3.7276206
LEVEL	2.62689376	1.56755066	1.71386576
SPD_LIM	0.50513172	0.46147966	0.45208475
SUR_COND_dry	9.99886131	15.8337965	16.25656509
TRAF_WAY_two_way	7.10797691	6.34214783	6.35494375
WEATHER_adverse	9.68802357	16.36388016	16.31727791

Scores for first few records

Row Id.	Predicted Class	Actual Class	Score for fatal	Score for no-injury	Score for non-fatal
2	no-injury	no-injury	25.94	31.42	30.93
56	no-injury	non-fatal	15.00	15.58	15.01
79	no-injury	no-injury	2.69	9.95	9.81
141	no-injury	no-injury	10.10	17.94	17.64
203	no-injury	non-fatal	2.42	11.76	11.41

- For row #2, "non-fatal" score is highest, so record is classified as non-fatal

- Next slide shows these scores plus the estimated probabilities

XLMiner output: scores & probabilities

Row Id.	Predicted Class	Actual Class	Score for fatal	Score for no-injury	Score for non-fatal	Prob. for class fatal	Prob. for class no-injury	Prob. for class non-fatal
2	no-injury	no-injury	25.94	31.42	30.93	0.002583566	0.618769909	0.378646525
56	no-injury	non-fatal	15.00	15.58	15.01	0.263257586	0.471205318	0.265537095
79	no-injury	no-injury	2.69	9.95	9.81	0.000376892	0.535717942	0.463905165
141	no-injury	no-injury	10.10	17.94	17.64	0.000226522	0.574000564	0.425772914
203	no-injury	non-fatal	2.42	11.76	11.41	5.18896E-05	0.586851481	0.413096629

Summary 总结

- Discriminant analysis is based on measuring the distance of a record from the class centers 判别分析建立在衡量记录与类别之间距离的基础上。
- The distance metric used is statistical distance, which takes into account the correlations between predictors 使用到的距离指标是统计距离。它考虑到了预测因子之间的相关性。
- Suitable for small datasets 判别分析对小数据集是适合的。
- Assumptions: equal correlations within each class, and normality (but fairly robust to violation of normality) 假设：各个分类中的预测因子之间有相同的相关性。预测因子满足多元正态分布（但违反这个假设情况下DA仍然有足够大的鲁棒性）。
- Sensitive to outliers (explore the data!) 对奇异值很敏感（需要进行数据探索）。
- Classification functions useful for profiling: can order predictors in terms of separating the classes 分类函数对剖绘有用：可根据预测因子对分类所起作用大小对预测因子进行排序。