

Distributional properties of Italian NN compounds: An Exploration with R

Marco Baroni¹ and Stefan Evert²

¹Center for Mind/Brain Sciences
University of Trento

²Cognitive Science Institute
University of Osnabrück

Potsdam, 3-14 September 2007

Outline

Introduction

Data

Clustering

k-means

Dimensionality reduction with PCA

NN Compounds

- ▶ Part of ongoing work with Emiliano Guevara (University of Bologna) and Vito Pirrelli (CNR/ILC, Pisa)
- ▶ Three-way classification inspired by theoretical (Bisetto and Scalise, 2005) and psychological work (e.g., Costello and Keane, 2001)
 - ▶ Relational (*computer center, angolo bambini*)
 - ▶ Attributive (*swordfish, esperimento pilota*)
 - ▶ Coordinative (*singer-songwriter, bar pasticceria*)

Relational compounds

- ▶ Express relation between two entities
- ▶ Heads are typically information containers, organizations, places, aggregators, pointers, etc.
- ▶ **M** “grounds” generic meaning of, or fills slot of **H**
- ▶ E.g., *stanza server* (“server room”), *fondo pensioni* (“pension fund”), *centro città* (“city center”)

Attributive compounds

- ▶ Interpretation of **M** is reduced to a “salient” property of its full semantic content, and this property is *attributed* to **H**:
- ▶ *presidente fantoccio* (“puppet president”), *progetto pilota* (“pilot project”)

Coordinative compounds

- ▶ Head and modifier denote similar/compatible entities, compound has coordinative reading
- ▶ **HM** is both **H** and **M**
- ▶ *viaggio spedizione* (“expedition travel”), *cantante attore* (“singer actor”)
- ▶ Ignored here

Ongoing exploration

- ▶ Data-set of frequent compounds: 24 **ATT** / 100 **REL**
- ▶ All **ATT** and **REL** compounds with $\text{freq} \geq 1,000$ in itWaC (2 billion token Italian Web-based corpus)
- ▶ Will the distinction between **ATT** and **REL** emerge from combination of distributional cues (also extracted from itWaC)?

Ongoing exploration

- ▶ Data-set of frequent compounds: 24 **ATT** / 100 **REL**
- ▶ All **ATT** and **REL** compounds with $\text{freq} \geq 1,000$ in itWaC (2 billion token Italian Web-based corpus)
- ▶ Will the distinction between **ATT** and **REL** emerge from combination of distributional cues (also extracted from itWaC)?
- ▶ Cues:
 - ▶ Semantic similarity between head and modifier
 - ▶ Explicit syntactic link
 - ▶ Relational properties of head and modifier
 - ▶ “Specialization” of head and modifier

Outline

Introduction

Data

Clustering

k-means

Dimensionality reduction with PCA

The data

H Compound head (Italian compounds are left-headed!)

M Modifier

TYPE attributive or relational

COS Cosine similarity between **H** and **M**

DELLL Log-likelihood ratio score based comparison between observed frequency of **H del M** (“**H** of the **M**”) and expected frequency under independence

HDELP Proportion of times **H** occurs in context **H del NOUN** over total occurrences of **H**

DELM Proportion of times **M** occurs in context **NOUN DEL M** over total occurrences of **M**

HNP Proportion of times **H** occurs in context **H NOUN** over total occurrences of **H**

NMP Proportion of times **M** occurs in context **NOUN M** over total occurrences of **M**

Cue statistics

- ▶ Read the file `comp.stats.txt` into a data-frame named `d` and “attach” the data-frame
 - ▶ `read.delim()` won't work here, use `read.table()`!
- ▶ Compute basic statistics
- ▶ Look at the distribution of each cue among compounds of type attributive (`at`) vs. relational (`re`)
- ▶ Using a t- or Mann-Whitney-test, determine for which cues the distinction between attributive and relational is significant
- ▶ Also, which cues are correlated? (use `cor()` on the subset of the data-frame that contains the cues)

Outline

Introduction

Data

Clustering

k-means

Dimensionality reduction with PCA

Outline

Introduction

Data

Clustering

k-means

Dimensionality reduction with PCA

Clustering

- ▶ *k-means*: one of the simplest and most widely used hard flat clustering algorithms
- ▶ For more sophisticated options, see the *cluster* and *e1071* packages

k-means

- ▶ The basic algorithm
 1. Start from k random points as cluster centers
 2. Assign points in data-set to cluster of closest center
 3. Re-compute centers (means) from points in each cluster
 4. Iterate cluster assignment and center update steps until configuration converges
- ▶ Given random nature of initialization, it pays off to repeat procedure multiple times (or to start from “reasonable” initialization)

Illustration of the k -means algorithm

See `help(iris)` for more information about the data set used

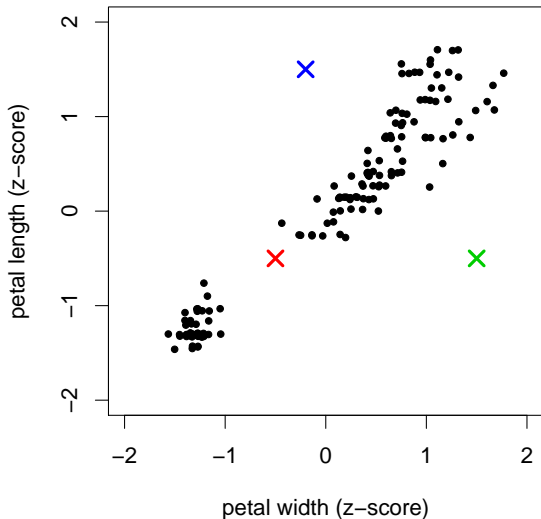


Illustration of the k -means algorithm

See `help(iris)` for more information about the data set used

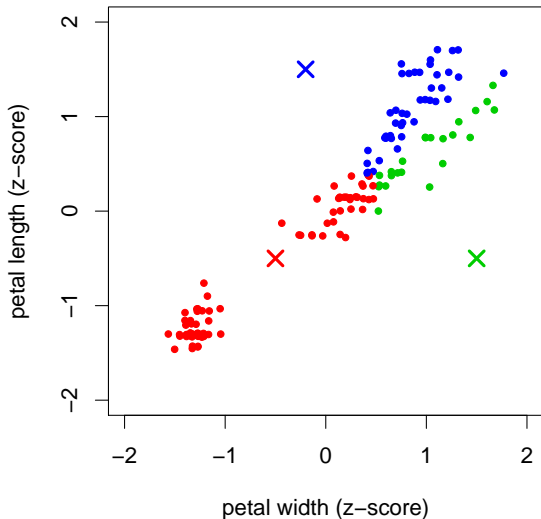


Illustration of the k -means algorithm

See `help(iris)` for more information about the data set used

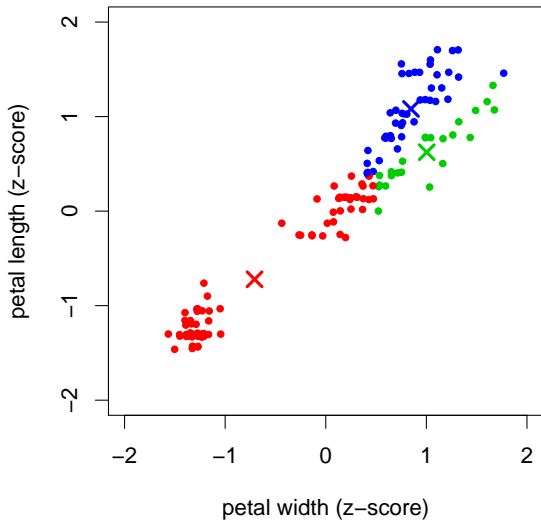


Illustration of the k -means algorithm

See `help(iris)` for more information about the data set used

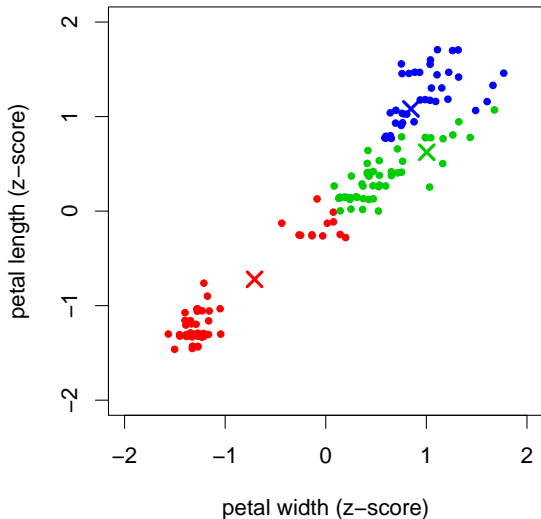


Illustration of the k -means algorithm

See `help(iris)` for more information about the data set used

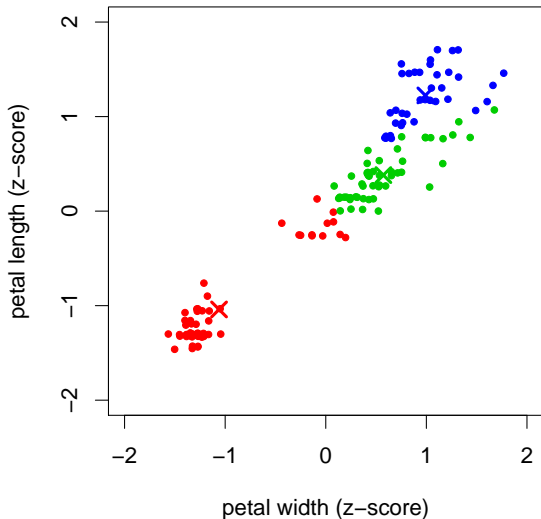


Illustration of the k -means algorithm

See `help(iris)` for more information about the data set used

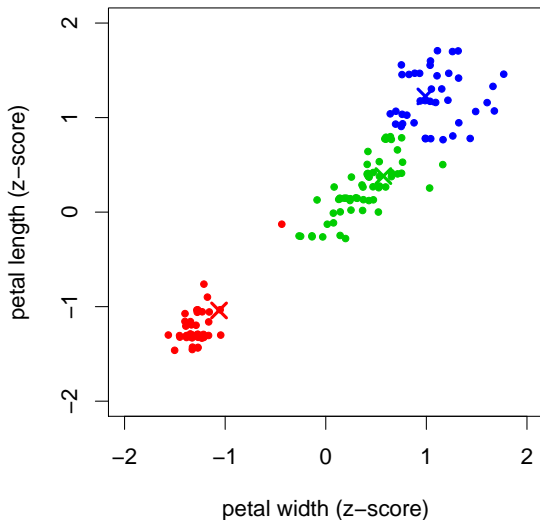


Illustration of the k -means algorithm

See `help(iris)` for more information about the data set used

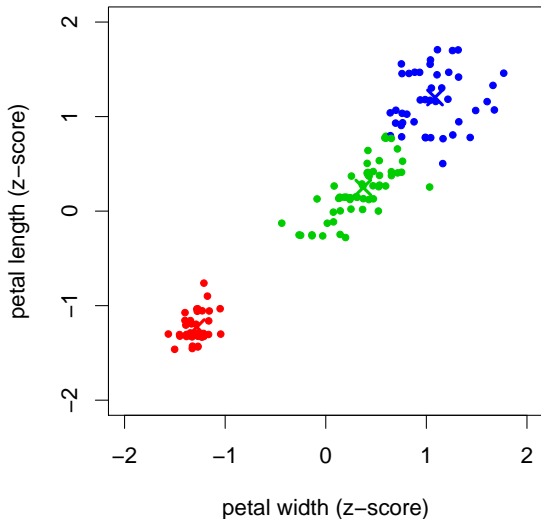


Illustration of the k -means algorithm

See `help(iris)` for more information about the data set used

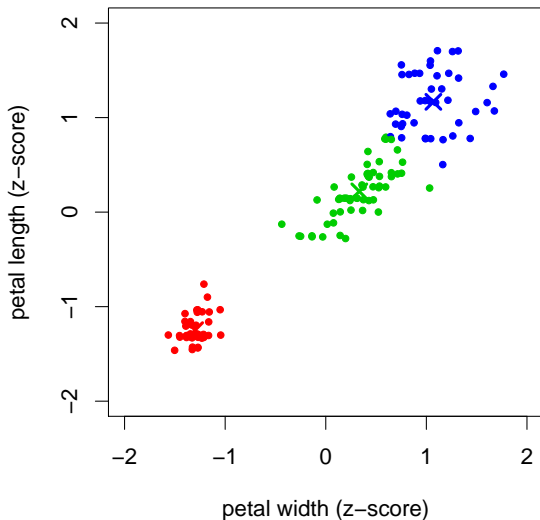


Illustration of the k -means algorithm

See `help(iris)` for more information about the data set used

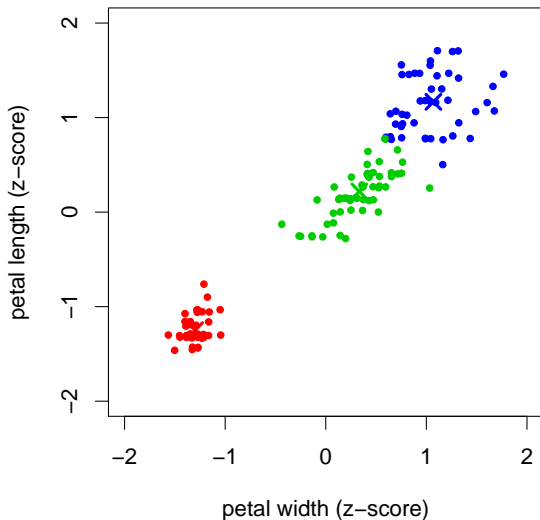
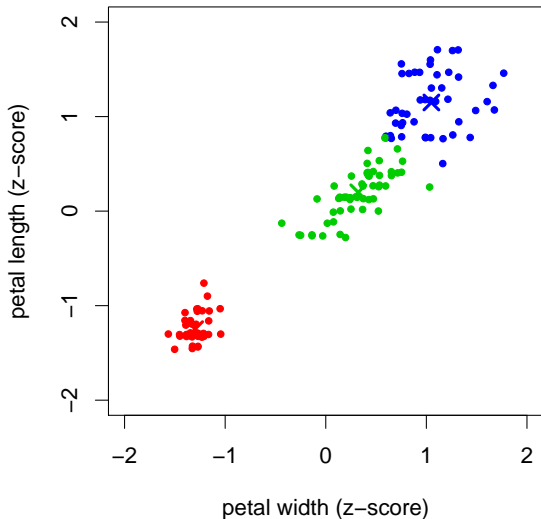


Illustration of the k -means algorithm

See `help(iris)` for more information about the data set used



k-means, first try

cues are in columns 4 to 9

```
> km <- kmeans(d[,4:9], 2, nstart=10)
> km
```

problem: extreme DELLL values dominate the clustering
(relevant small cluster might be cluster 2 in your solution)

```
> DELLL[km$cluster==1]
```

```
> head(sort(DELLEL, decreasing=TRUE))
```

Scaling and trying again

```
> scaled <- scale(d[,4:9])
> summary(d[4:9])    # distribution of original data
> summary(scaled)    # after scaling

> km <- kmeans(scaled, 2, nstart=10)
> km

> table(km$cluster, d$TYPE) # confusion matrix
```

Outline

Introduction

Data

Clustering

k-means

Dimensionality reduction with PCA

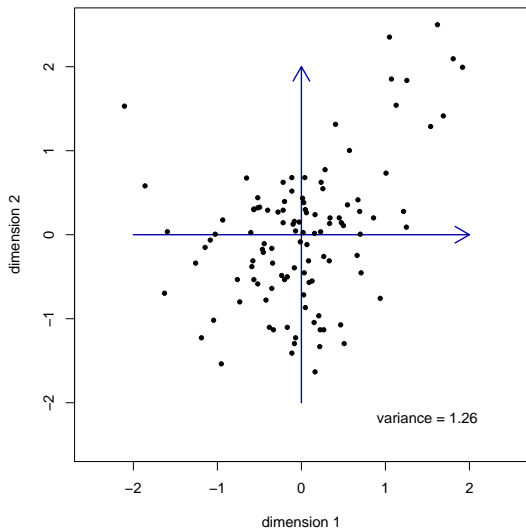
Dimensionality reduction

- ▶ To find “latent” variables
- ▶ To reduce random noise
- ▶ For easier visualization

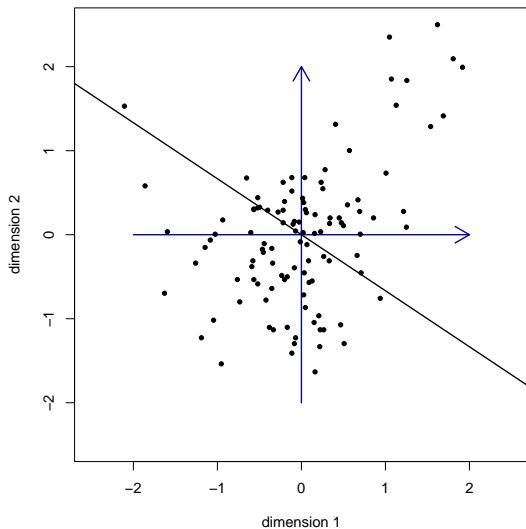
Principal component analysis (PCA)

- ▶ Find a set of orthogonal dimensions such that the first dimension “accounts” for the most *variance* in the original data-set, the second dimension accounts for as much as possible of the remaining variance, etc.
- ▶ The top k dimensions (principal components) are the best sub-set of k dimensions to approximate the spread in the original data-set
- ▶ Correlation of original variables to principal components might reveal interesting underlying factors

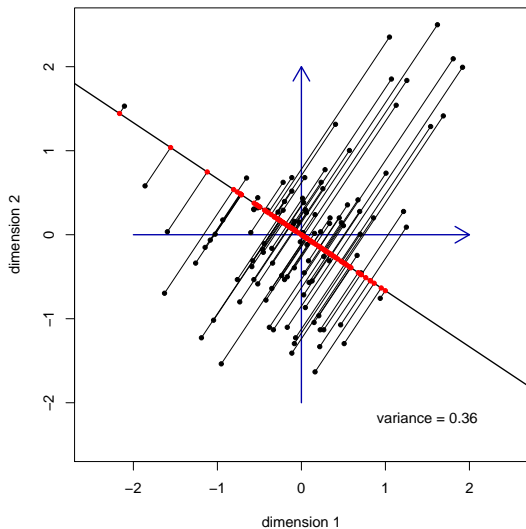
Preserved variance: examples



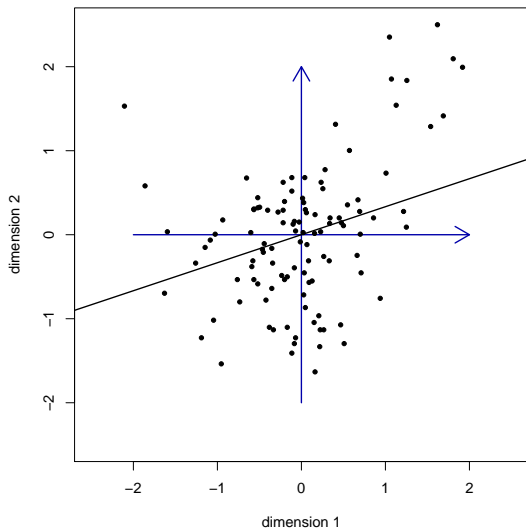
Preserved variance: examples



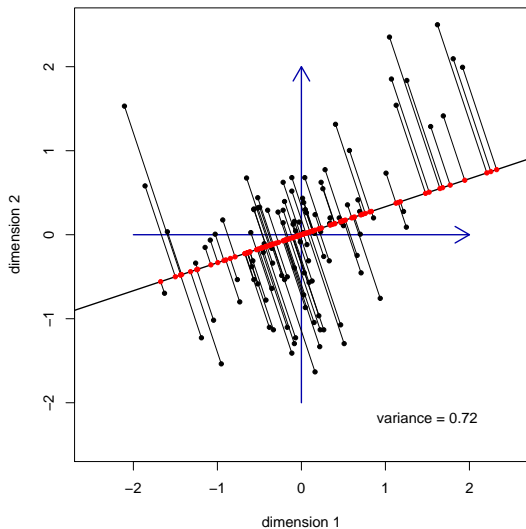
Preserved variance: examples



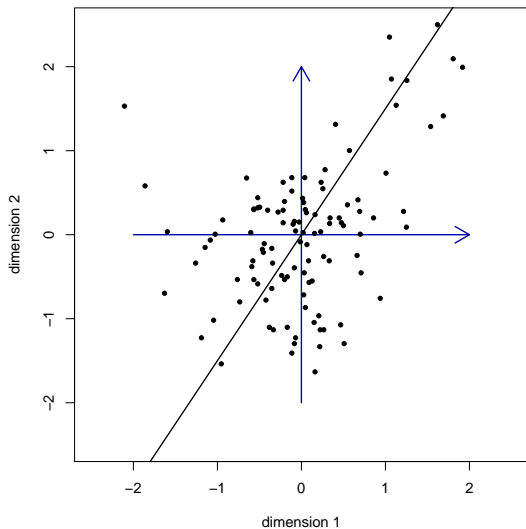
Preserved variance: examples



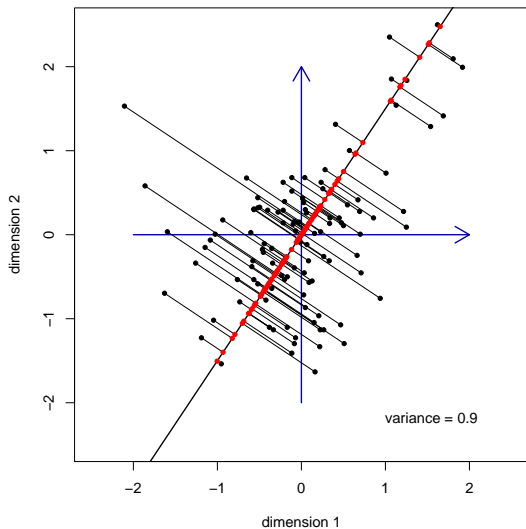
Preserved variance: examples



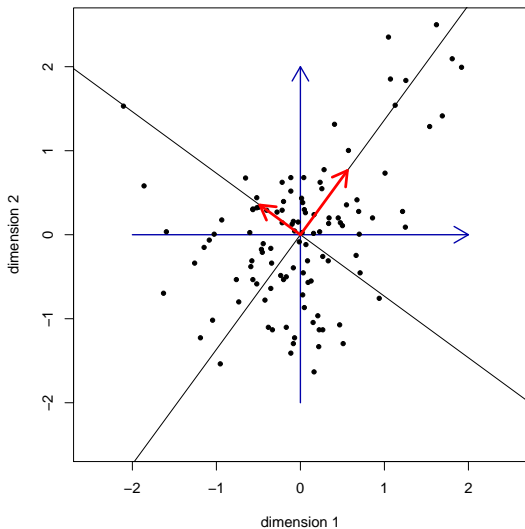
Preserved variance: examples



Preserved variance: examples



Adding an orthogonal dimension



PCA in R

```
> temp <- subset(d, select=c(HNPROP, NMPROP,  
  DELLL, HDELPROP, DELMPROP, COS))  
  
> pr <- prcomp(temp, scale=TRUE)  
> pr  
  
> plot(pr)  
  
> biplot(pr)  
> biplot(pr, xlab=TYPE,  
  xlim=c(-.25, .25), ylim=c(-.25, .25))
```


More refined plotting

- ```
> plot(pr$x[,1:2], type="n",
 xlim=c(min(pr$x[,1]),4),
 ylim=c(min(pr$x[,2]),4)) # only sets up plot region
```
- ```
> points(subset(pr$x, TYPE=="re"),  
        col="blue", pch=19, lwd=2) # blue points for type "re"
```
- ```
> points(subset(pr$x, TYPE=="at"),
 col="red", pch=19, lwd=2) # red points for type "at"
```
- ```
> legend("topright", inset=.05,  
      fill=c("red","blue"), cex=1.5,  
      legend=c("ATT","REL"))      # legend explains colors
```

Adding the cues

- > text (pr\$rotation[1,1]*4, pr\$rotation[1,2]*4,
label="H N", cex=1.7)
- > text (pr\$rotation[2,1]*4, pr\$rotation[2,2]*4,
label="N M", cex=1.7)
- > text (pr\$rotation[3,1]*4, pr\$rotation[3,2]*4,
label="H DEL M", cex=1.7)
- > text (pr\$rotation[4,1]*4, pr\$rotation[4,2]*4,
label="H DEL", cex=1.7)
- > text (pr\$rotation[5,1]*4, pr\$rotation[5,2]*4,
label="DEL M", cex=1.7)
- > text (pr\$rotation[6,1]*4, pr\$rotation[6,2]*4,
label="COS", cex=1.7)

Trying k-means again

```
> km <- kmeans(pr$x[,1:4], 2, nstart=10)
> table(km$cluster, d$TYPE)
```

what happens with more/fewer dimensions?

```
> plot(pr$x[,1:2], type="n",
      xlim=c(min(pr$x[,1]), 4),
      ylim=c(min(pr$x[,2]), 4))

> text(pr$x[,1], pr$x[,2],
      col=km$cluster, labels=TYPE)
```

now refine this plot as on previous slides