Unsupervised Learning

Content:

- comparison with supervised learning
- market basket analysis
- association rules (Apriori algorithm)
- cluster analysis
- K-means algorithm
- hierarchical clustering

Supervised versus unsupervised learning

- Supervised learning: "learning *with* a teacher"

  $P(X_1, \ldots, X_p, Y)$

  where $\mathbf{X} = \{X_1, \ldots, X_p\}$ are inputs, and Y is output or class variable

  Problems:

  - find most frequent value for Y given $\mathbf{X}$
  - find the average value of Y as a function of $\mathbf{X}$

- Unsupervised learning: "learning *without* a teacher"

  $P(X_1, \ldots, X_p)$

  where $\mathbf{X} = \{X_1, \ldots, X_p\}$ are variables in $\mathbf{X}$-space describing the problem

  Problem: what is the structure of $\mathbf{X}$-space?

Market basket analysis

Aims:

- Trying to understand customer behaviour
- Collect check-out counter information for each customer
- Classical example: "A convenient store in USA found out that beer and diapers sell together on Thursday evenings."
- Try to discover associations
- Results are used for:
  - improved stocking of shelves
  - cross-marketing in sales
  - sales promotion
  - catalogue design
  - consumer segmentation

Example: association rules

Age	Spectacle prescription	Ast	Tear production rate	Lens
young	myope	no	reduced	none
young	myope	no	normal	soft
young	myope	yes	reduced	none
young	hypermetrope	no	reduced	none
young	hypermetrope	no	normal	soft
young	hypermetrope	yes	reduced	none
pre-presbyopic	myope	no	reduced	none
pre-presbyopic	myope	no	normal	soft
pre-presbyopic	myope	yes	reduced	none
pre-presbyopic	hypermetrope	no	reduced	none
pre-presbyopic	hypermetrope	no	normal	soft
pre-presbyopic	hypermetrope	yes	normal	none
presbyopic	myope	no	reduced	none
presbyopic	myope	no	normal	none
presbyopic	myope	yes	reduced	none
presbyopic	hypermetrope	no	normal	soft
presbyopic	hypermetrope	yes	normal	none

Tear-prod-rate = *reduced* $\rightarrow$
Contact-lenses = *none*

Contact-lenses = *soft* $\rightarrow$
(Astigmatism = *no* $\wedge$ Tear-prod-rate = *normal*)

Learning association rules: Apriori

X_1	X_2	$\cdots$	X_p
$\vdots$	$\vdots$	$\vdots$	$\vdots$

$P(X_1, X_2, \ldots, X_p)$

- Aim: find values for $X_1, X_2, \ldots, X_p$ such that

  $$P(x_1, x_2, \ldots, x_p)$$

  is large

- Simplification: find values x_j for X_j, such that

  $$P\left(\bigwedge_{j=1}^{p} \bigvee_{x_j \in S_j} (X_j = x_j)\right)$$

  is large, with

  $$S_j \subseteq \text{Domain}(X_j)$$

  for $j = 1, \ldots, p$

Other simplifications

Original formulation:

$$P\left(\bigwedge_{j=1}^{p} \bigvee_{x_j \in S_j} (X_j = x_j)\right)$$

Choices:

1. assume that $S_j = \text{Domain}(X_j)$, then

   $$\bigvee_{x_j \in S_j} (X_j = x_j) \equiv \top$$

   or,

2. assume that $|S_j| = 1$, with subset of variables from $\{X_1, \ldots, X_p\}$, then

   $$\bigvee_{x_j \in S_j} (X_j = x_j) \equiv (X_j = x_j)$$

Choosing between (1) or (2) for each variable, yields for each variable either $(X_j = x_j)$ or $\top$ (variable is removed)

Final formulation

Find $\mathcal{J} \subseteq \{1, \ldots, p\}$, such that

$$\hat{P}\left(\bigwedge_{j \in \mathcal{J}} (X_j = x_j)\right) = \frac{1}{N} \sum_{i=1}^{N} \prod_{j \in \mathcal{J}} \iota(X_j = x_{i,j}) = T(\mathcal{J})$$

is large, where $\iota(P) = \begin{cases} 1 & \text{if } P = \top \\ 0 & \text{otherwise} \end{cases}$ and D is a dataset with $N = |D|$, and $X_j = x_{i,j}$ is the value of X_j in instance i. The set

$$\mathcal{I} = \{X_j = x_j \mid j \in \mathcal{J}\}$$

is called the item set, and $T(\mathcal{J})$ is called the support

Further simplification: assume that variables X_j are binary (non-essential simplification)

Apriori algorithm: item sets

- Choose support threshold t, and only consider item sets $\mathcal{J}$ with $T(\mathcal{J}) > t$
- If $\mathcal{L} \subseteq \mathcal{J}$ then $T(\mathcal{L}) \geq T(\mathcal{J})$ (the more conditions, the less support)
- This implies that any item set $\mathcal{J} \supset \mathcal{L}$ with $\mathcal{L}$ deleted, can also be deleted

Examples for $t = 3/17$:

- some single-item sets:

  {Age = *young*}, $T = 6/17$
  {Spectacle = *hypermetrope*}, $T = 8/17$
  {Contact-lenses = *none*}, $T = 12/17$

- some two-item sets:

  {Age = *young*, Spectacle = *hypermetrope*}, $T = 3/17$ (deleted)
  {Age = *young*, Contact-lenses = *none*}, $T = 4/17$
  {Spectacle = *hypermetrope*, Contact-lenses = *none*}, $T = 5/17$

Apriori algorithm: rules

Steps in the algorithm:

1. **generate item sets** with minimum support as required
2. **generate rules** with minimum accuracy a (confidence) where accuracy $\alpha(r)$ is defined as:

$$\alpha(\phi \to \psi) = \frac{T(\phi \wedge \psi)}{T(\psi)}$$

which can be seen as an estimate of $P(\psi \mid \phi)$. Final ruleset $\mathcal{R}$

$$\mathcal{R} = \{r \mid \alpha(r) > a\}$$

Example of rules:

Spectacle = *hypermetrope* $\to$
  Contact-lenses = *none*, $\alpha = 5/12$
Contact-lenses = *none* $\to$
  Age = *young*, $\alpha = 4/6$

Apriori: example from WEKA

```
Minimum support: 0.25
Minimum metric <accuracy>: 0.9

Size of set of large itemsets L(1): 11
Size of set of large itemsets L(2): 20
Size of set of large itemsets L(3): 6

Best rules found:
 1. tear-prod-rate=reduced 9
    ==> contact-lenses=none 9 alpha:(1)
 2. spectacle-prescrip=myope tear-prod-rate=reduced 6
     ==> contact-lenses=none 6    alpha:(1)
 3. astigmatism=yes 6 ==> contact-lenses=none 6 alpha:(1)
 4. contact-lenses=soft 5 ==>
    astigmatism=no tear-prod-rate=normal 5 alpha:(1)
 5. astigmatism=no contact-lenses=soft 5
    ==> tear-prod-rate=normal 5 alpha:(1)
 6. tear-prod-rate=normal contact-lenses=soft 5
    ==> astigmatism=no 5 alpha:(1)
 7. astigmatism=no tear-prod-rate=reduced 5
    ==> contact-lenses=none 5 alpha:(1)
 8. contact-lenses=soft 5
    ==> tear-prod-rate=normal 5 alpha:(1)
 9. contact-lenses=soft 5 ==> astigmatism=no 5 alpha:(1)
10. astigmatism=yes tear-prod-rate=reduced 4
    ==> contact-lenses=none 4 alpha:(1)
```

Note: there can be arbitrary conjunctions in premises and consequences of rules

Apriori: tricks

Suppose the the three-item set $\mathcal{I}$ contains the following elements (with support greater than the threshold):

$\{A, B, C\}$
$\{A, C, D\}$
$\{A, B, E\}$
$\{B, C, E\}$

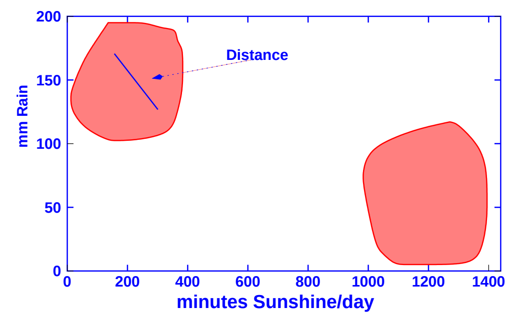
where elements are of the form $X_j = x_j$

Then, the four-item set

$\{A, B, C, D\}$

is not accepted, as for example $\{B, C, D\}$ is below the support threshold, and therefore lacking in the three-item sets

Cluster analysis



- Grouping of related objects into subsets (clusters)
- Sometimes: ordering of clusters into a hierarchy
- Required: degree of (dis)similarity
- Top-down and bottom-up approaches

Dissimilarity

Let $\mathbf{X} = \{X_1, \ldots, X_p\}$ be a set of variables, where the variable X_j attains a value $x_{i,j}$ within instance $\mathbf{x}_i \in D$ (dataset)

Dissimilarity $d(x_{i,j}, x_{k,j})$ between values $x_{i,j}$ and $x_{k,j}$ of variable X_j:

- quantitative variable, various examples:
  - *squared distance* $d(x_{i,j}, x_{k,j}) = (x_{i,j} - x_{k,j})^2$
  - *absolute value* $d(x_{i,j}, x_{k,j}) = f(|x_{i,j} - x_{k,j}|)$, where f is a monotonously increasing function, e.g. $f(x) = x^p, p \in \mathbb{N}$
- qualitative (categorical) variables: if X_j has m values, then define vector $\mathbf{x}_j$, with

$$x_{i,j} = \begin{cases} 1 & \text{if } X_j = x_{i,j} \\ 0 & \text{otherwise} \end{cases}$$

Multi-variable dissimilarity

- Difference between two instances $\mathbf{x}_i, \mathbf{x}_k \in D$:

$$\Delta(\mathbf{x}_i, \mathbf{x}_k) = \sum_{j=1}^{p} \omega_j \cdot d(x_{i,j}, x_{k,j})$$

  with weights ω_j, and $\sum_{j=1}^{p} \omega_j = 1$

- Average dissimilarity for dataset D, with $N = |D|$:

$$\bar{\Delta} = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{k=1}^{N} \Delta(\mathbf{x}_i, \mathbf{x}_k) = \sum_{j=1}^{p} \omega_j \cdot \bar{d}_j$$

  with

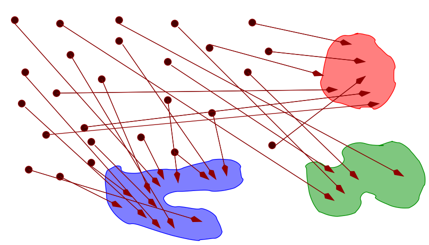
$$\bar{d}_j = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{k=1}^{N} d(x_{i,j}, x_{k,j})$$

- Equal contribution of variables to dissimilarity: $\omega_j = \frac{1}{\bar{d}_j}$, which is normally undesirable

Some remarks

- Choice of appropriate (dis)similarity measure is more important than the choice of the algorithm
- This choice is dependent of the problem domain
- Incorporating domain characteristics into the weight vector ω is the difficult part
- Normally, matters are complicated by:
  - mixture of qualitative and quantitative variables
  - missing values
- Alternative: correlation $\rho(\mathbf{x}_i, \mathbf{x}_k)$ (similarity)

Combinatorial clustering algorithm



Let D be a dataset with $N = |D|$, and let K be the prespecified number of clusters

Clustering problem: Find function

$$C : \{1, \ldots, N\} \rightarrow \{1, \ldots, K\}$$

called encoder with $\forall \mathbf{x}_i \in D : C(i) = k$, fulfilling some measure of optimality

Example measure: total point scatter

$$T = \frac{1}{2} \sum_{i=1}^{N} \sum_{k=1}^{N} \Delta(\mathbf{x}_i, \mathbf{x}_k)$$

Decomposition of total scatter

$$\begin{aligned} T &= \frac{1}{2}\sum_{i=1}^{N}\sum_{k=1}^{N}\Delta(\mathbf{x}_i,\mathbf{x}_k) \\ &= \frac{1}{2}\sum_{l=1}^{K}\sum_{C(i)=l}\left(\sum_{C(k)=l}\Delta_{i,k} + \sum_{C(k)\neq l}\Delta_{i,k}\right) \\ &= W(C) + B(C) \end{aligned}$$

where $\Delta_{i,k} = \Delta(\mathbf{x}_i,\mathbf{x}_k)$; T is *constant* for dataset D

Components:

- within-cluster point scatter:

$$W(C) = \frac{1}{2}\sum_{l=1}^{K}\sum_{C(i)=l}\sum_{C(k)=l}\Delta(\mathbf{x}_i,\mathbf{x}_k)$$

- between-cluster point scatter:

$$B(C) = \frac{1}{2}\sum_{l=1}^{K}\sum_{C(i)=l}\sum_{C(k)\neq l}\Delta(\mathbf{x}_i,\mathbf{x}_k)$$

Algorithm: minimise $W(C) = T - B(C)$

Basic ideas K-means algorithm

Basic approach:

- greedy approach (so, fast – cluster oriented)
- dissimilarity: squared Euclidean distance

$$\Delta(\mathbf{x}_i,\mathbf{x}_k) = \sum_{j=1}^{p}(x_{i,j} - x_{k,j})^2 = ||\mathbf{x}_i - \mathbf{x}_k||^2$$

- within cluster point scatter:

$$\begin{aligned} W(C) &= \frac{1}{2}\sum_{l=1}^{K}\sum_{C(i)=l}\sum_{C(k)=l}||\mathbf{x}_i - \mathbf{x}_k||^2 \\ &= \sum_{l=1}^{K}\sum_{C(i)=l}||\mathbf{x}_i - \bar{x}_l||^2 \end{aligned}$$

  where $\bar{x}_l$ is the average (cluster centroid) of cluster l

- optimisation problem: determine

$$C^* = \min_C \sum_{l=1}^{K}\sum_{C(i)=l}||\mathbf{x}_i - \bar{x}_l||^2$$

K-means in WEKA

```
K-means K = 2
=============

Cluster centroids:
Cluster 0: pre-presbyopic hypermetrope yes reduced none
Cluster 1: young myope no reduced none

Clustered Instances: C0 14 (58%), C1 10 (42%)

K-means K = 3
=============

Cluster centroids:
Cluster 0: pre-presbyopic hypermetrope yes reduced none
Cluster 1: young myope no reduced none
Cluster 2: young myope yes normal hard

Clustered Instances: C0 11 (46%), C1 9 (38%), C2 4 (17%)

K-means K = 4
=============

Cluster centroids:
Cluster 0: pre-presbyopic hypermetrope yes reduced none
Cluster 1: young myope no reduced none
Cluster 2: young myope yes normal hard
Cluster 3: pre-presbyopic hypermetrope no normal soft

Clustered Instances: C0 9 (38%), C1 7 (29%), C2 4 (17%),
                     C3 4 (17%)
```

K-means (continued)

Solution of

$$C^* = \min_C \sum_{l=1}^{K}\sum_{C(i)=l}||\mathbf{x}_i - \bar{x}_l||^2$$

Note that for the average $\bar{x}_S$ of the data in S it holds that

$$\bar{x}_S = \arg\min_m \sum_{i\in S}||x_i - m||^2$$

Hence, solving

$$\min_{C,\{m_l\}_{l=1}^{K}} \sum_{l=1}^{K}\sum_{C(i)=l}||\mathbf{x}_i - m_l||^2$$

yields C^* (this is a local optimum)

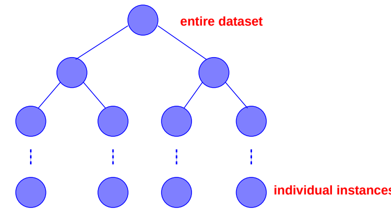
K-means algorithm

K-means(D, K)

{
  initialise C
  for $l = 1, \ldots, K$ **do**
    $m_l \leftarrow$ initial-value
  until C is stable **do**
    for $l = 1, \ldots, K$ **do**
      $m_l \leftarrow \arg\min_{m_l} \sum_{C(i)=l} ||\mathbf{x}_i - m_l||^2$
    for $i = 1, \ldots, N$ **do**
      $C(i) \leftarrow \arg\min_{1 \leq l \leq K} ||\mathbf{x}_i - m_l||^2$
}

- Iteration continues until the assignments made by the encoder C do not change anymore
- Initial choices for means m_l affect results; solution:
  - take random choices for m_l
  - determine the m_l's for which C is minimal
- Experimentation with different number of clusters K is normally required

Hierarchical clustering



Dendrogram:

- binary tree, where
  - root represents entire dataset, and leaves individual instances
  - from leaves to root, dissimilary between merged clusters in increasing
- single-linkage clustering:

$$\Delta(G, H) = \min_{\mathbf{x} \in G, \mathbf{x}' \in H} \Delta(\mathbf{x}, \mathbf{x}')$$

  is the difference between clusters G and H (other possibilities: max and cluster average)

Microarray example