Information theory in ML
Feature Selection

For *efficiency* of the classifier and to *suppress noise* choose subset of all possible features.

→ Selected features should be
  • frequent to *avoid overfitting* the classifier to the training data,
  • but not too frequent in order to be characteristic.

Features should be good *discriminators* between classes (i.e. frequent/characteristic in one class but infrequent in other classes).

**Approach:**
- compute measure of discrimination for each feature
- select the top k most discriminative features in greedy manner

*tf*idf is usually not a good discrimination measure, and may give undue weight to terms with high *idf* value (leading to the danger of overfitting)
Entropy: idea

\[ P(D) = \frac{7}{8} \]
\[ P(K) = \frac{1}{8} \]

Die Toten Hosen:

„Sieben fuhren nach Düsseldorf und einer fuhr nach Köln“

\[ H_{koelsch} = H_{alt} = (\frac{7}{8} \cdot \log_2 \frac{7}{8} + \frac{1}{8} \cdot \log_2 \frac{1}{8}) \approx 0.19 \]

\[ H_{water} = - \left( \frac{1}{2} \cdot \log_2 \frac{1}{2} + \frac{1}{2} \cdot \log_2 \frac{1}{2} \right) = 1 \]

\[ \lim_{p_i \to 0} p_i \cdot \log_2 p_i = 0 \]
### Example for Feature Selection

#### Feature Selection Table

<table>
<thead>
<tr>
<th></th>
<th>f1</th>
<th>f2</th>
<th>f3</th>
<th>f4</th>
<th>f5</th>
<th>f6</th>
<th>f7</th>
<th>f8</th>
</tr>
</thead>
<tbody>
<tr>
<td>d1:</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d2:</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>d3:</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d4:</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d5:</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d6:</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d7:</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d8:</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d9:</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>d10:</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>d11:</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>d12:</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

#### Class Tree:

- **Entertainment**
  - **Math**
    - **Calculus**
    - **Algebra**

#### Training Docs:
- d1, d2, d3, d4 → Entertainment
- d5, d6, d7, d8 → Calculus
- d9, d10, d11, d12 → Algebra
Discriminative Classifiers: Decision Trees

given: a multiset of \textbf{m-dimensional training data records} \subseteq \text{dom}(A_1) \times \ldots \times \text{dom}(A_m) \text{ with numerical, ordinal, or categorial attributes } A_i \text{ (e.g. term occurrence frequencies } \subseteq \mathbb{N}_0 \times \ldots \times \mathbb{N}_0) \text{ and with class labels} \\

wanted: a tree with
• \textbf{attribute value conditions} of the form
  • \( A_i \leq \text{value} \) for numerical or ordinal attributes  \\
  or  \\
  • \( A_i \in \text{value set} \) or \( A_i \cap \text{value set} = \emptyset \)  \\
  for categorial attributes  \\
  or  \\
  • linear combinations of this type  \\
  for several numerical attributes \[ \sum k_i A_i \leq \text{value} \]
  as \textbf{inner nodes} and
• \textbf{labeled classes as leaf nodes}
Examples for Decision Trees (1)

\[
\begin{align*}
tf(\text{homomorphism}) & \geq 2 \\
tf(\text{vector}) & \geq 3 \\
tf(\text{limit}) & \geq 2
\end{align*}
\]

\[
\begin{align*}
\text{Lineare Algebra} & \quad \text{Calculus} & \quad \text{Other} \\
\text{Algebra} & \quad &
\end{align*}
\]

\[
\begin{align*}
\text{has read Tolkien} & \\
\text{has read Eco} & \quad \text{boring}
\end{align*}
\]

\[
\begin{align*}
\text{intellectual} & \quad \text{uneducated}
\end{align*}
\]

\[
\begin{align*}
\text{salary} & \geq 100000 \\
\text{credit worthy} & \quad \text{university degree} & \quad \text{salary} & \geq 50000
\end{align*}
\]

\[
\begin{align*}
\text{credit worthy} & \quad \text{not credit worthy}
\end{align*}
\]
Top-Down Construction of Decision Tree

Input: decision tree node $k$ that represents one partition $D$ of $\text{dom}(A_1) \times \ldots \times \text{dom}(A_m)$

Output: decision tree with root $k$

1) BuildTree ($\text{root, dom}(A_1) \times \ldots \times \text{dom}(A_m)$)
2) PruneTree: reduce tree to appropriate size

with:

procedure BuildTree ($k, D$):
    if $k$ contains only training data of the same class then terminate;
    determine split dimension $A_i$;
    determine split value $x$ for most suitable partitioning of $D$ into
    $D_1 = D \cap \{d | d.A_i \leq x\}$ and $D_2 = D \cap \{d | d.A_i > x\}$;
    create children $k_1$ and $k_2$ of $k$;
    BuildTree ($k_1, D_1$); BuildTree ($k_2, D_2$);
Split Criterion: Information Gain

Goal is to split current node such that the resulting partitions are as pure as possible w.r.t. class labels of the corresponding training data. Thus we aim to minimize the **impurity** of the partitions.

An approach to define impurity is via the entropy-based (statistical) **information gain** (referring to the distribution of class labels within a partition)

\[
G (k, k1, k2) = H(k) – ( p1*H(k1) + p2*H(k2) )
\]

where:

- \( n_k \): # training data records in k
- \( n_{k,j} \): # training data records in k that belong to class j
- \( p1 = \frac{n_{k1}}{n_k} \) and \( p2 = \frac{n_{k2}}{n_k} \)

\[
H(k) = -\sum_j \frac{n_{k,j}}{n_k} \log_2 \frac{n_{k,j}}{n_k}
\]
Example for Decision Tree for Text Classification

<table>
<thead>
<tr>
<th></th>
<th>f1</th>
<th>f2</th>
<th>f3</th>
<th>f4</th>
<th>f5</th>
<th>f6</th>
<th>f7</th>
<th>f8</th>
</tr>
</thead>
<tbody>
<tr>
<td>d1:</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>d2:</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d3:</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d4:</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>d5:</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>d6:</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

C1: Algebra
- d1, d2
C2: Calculus
- d3, d4
C3: Stochastics
- d5, d6

G = H(k) – (2/6*H(k1) + 4/6*H(k2))

H(k) = \frac{1}{3} \log_3 3 + \frac{1}{3} \log_3 3 + \frac{1}{3} \log_3 3

H(k1) = \log 1 + 0 + 0

H(k2) = 0 + \frac{1}{2} \log 2 + \frac{1}{2} \log 2

G = \log 3 - 0 - \frac{2}{3} \times 1 \approx 1.6 - 0.66 = 0.94

f_2 > 0

Algebra

f_7 > 1

Stochastics

Calculus
Simple (Class-unspecific) Criteria for Feature Selection

**Document Frequency Thresholding:**
Consider for class Cj only terms ti that occur in at least \( \delta \) training documents of Cj.

**Term Strength:**
For decision between classes C1, ..., Ck select (binary) features Xi with the highest value of

\[
s(X_i) := P[X_i \text{ occurs in doc } d \mid X_i \text{ occurs in similar doc } d']
\]

To this end the set of similar doc pairs (d, d') is obtained
- by thresholding on pairwise similarity or
- by clustering/grouping the training docs.

+ further possible criteria along these lines
**Information gain:**
For discriminating classes $c_1, ..., c_k$ select the (binary) features $X_i$ with the largest gain in entropy

\[
G(X_i) = \sum_{j=1}^{k} P[c_j] \log_2 \frac{1}{P[c_j]} \\
- P[X_i] \sum_{j=1}^{k} P[c_j | X_i] \log_2 \frac{1}{P[c_j | X_i]} \\
- P[\overline{X_i}] \sum_{j=1}^{k} P[c_j | \overline{X_i}] \log_2 \frac{1}{P[c_j | \overline{X_i}]}
\]

Generalization for non-binary features is straightforward..
Feature Selection Based on Mutual Information

Mutual information (Kullback-Leibler distance, relative entropy):

for class $c_j$ select those (binary) features $X_i$ with the largest value of

$$MI(X_i, c_j) = \sum_{X \in \{X_i, \bar{X}_i\}} \sum_{C \in \{c_j, \bar{c}_j\}} P[X \land C] \log \frac{P[X \land C]}{P[X] P[C]}$$

and for discriminating classes $c_1, \ldots, c_k$:

$$MI(X_i) = \sum_{j=1}^{k} P[c_j] MI(X_i, c_j)$$

Generalization for non-binary features is straightforward
Conditional mutual information

Idea: construct the set of features iteratively (e.g. by adding features one by one)

For a new candidate, compute score $s(n, k)$ which depends of ist individual discriminative power, given class labels and previously chosen features.

By taking the feature $X_n$ with the maximum score $s(n, k)$ we ensure that the new feature is both informative and different than the preceding ones - at least in terms of predicting the class label $Y$.

Instrument for constructing $s(n, k)$: conditional mutual information.

$$I(X; Y \mid Z) = \mathbb{E}_Z (I(X; Y) \mid Z) = \sum_{z \in Z} p_Z(z) \sum_{y \in Y} \sum_{x \in X} p_{X,Y\mid Z}(x, y \mid z) \log \frac{p_{X,Y\mid Z}(x, y \mid z)}{p_{X\mid Z}(x \mid z)p_{Y\mid Z}(y \mid z)},$$
Comparing distributions

We are given: two probability distributions P and Q.
Question: how (dis)similar are they to each other?

Common example: distributions of two documents over (some) latent topics, Expressed by means of multinomials..

We can compute $\text{MI}(P,Q)$ or $\text{MI}(Q,P)$ – but they are both asymmetric

Workaround: take as a dissimilarity measure between the value of

$\text{DIS}(P,Q) = \text{DIS}(Q,P) = 0.5 \times \text{MI}(P,Q) + 0.5 \times \text{MI}(Q,P)$ or

$\text{JSD}(P,Q) = \text{JSD}(Q,P) = 0.5 \times \text{MI}(P,M) + 0.5 \times \text{MI}(Q,M)$ \hspace{1cm} M = 0.5 \times (P+Q)

- JSD also known as „Jensen-Shannon divergence“
- Not yet a „real“ distance (JS is not a proper metric), but indicator
- Becomes a metric with further regularization (e.g. square root of JSD)