Bayesian Networks and Decision Graphs

Chapter 8

Classification: Text Categorization

The Association for Computing Machinery (ACM) maintains a subject classification scheme for computer science research papers. Part of the subject hierarchy (1998 version):

I. Computing Methodologies
I.2 Artificial Intelligence
I.2.6 Learning
- Analogies
- Concept learning
- Connectionism and neural nets
- Induction
- Knowledge acquisition
- Language acquisition
- Parameter learning

Papers are manually classified by authors or editors.

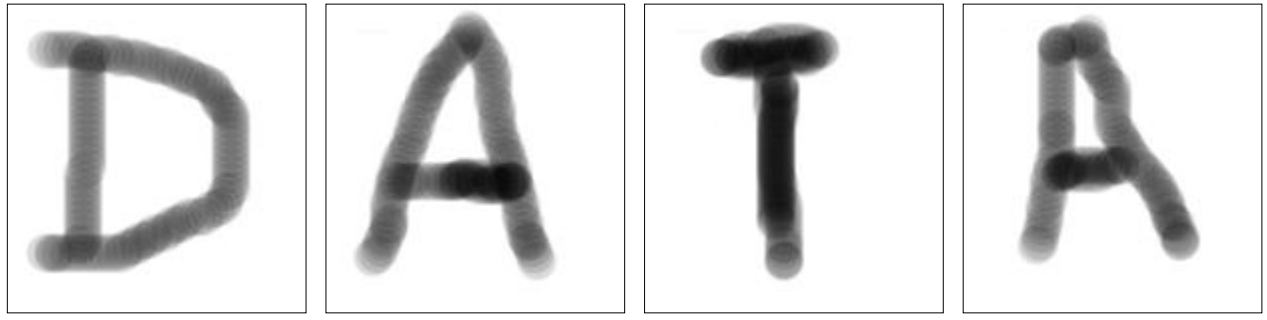
Data: collection of classified papers (full text or abstracts)
Task: build a *classifier* that automatically assigns a subject index to new, unclassified papers.

Classification: Spam Filtering

Spam filtering in Mozilla: user trains the mail reader to recognize spam by manually labeling incoming mails as spam/no spam.

Data: collection of user-classified emails (full text).
Task: build a classifier that automatically categorizes an incoming email as spam/no spam

Classification: Character Recognition



Data: collection of handwritten characters, correctly labeled.

Task: build a classifier that identifies new handwritten characters.

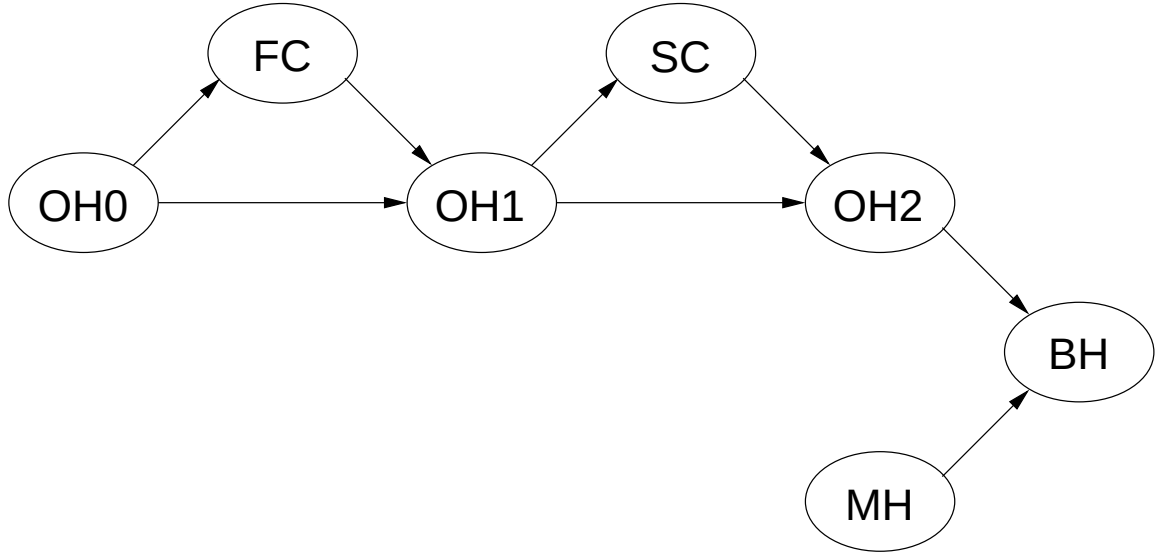
Classification: Credit Rating

From existing customer data predict whether a person applying for a new loan will repay or default on the loan.

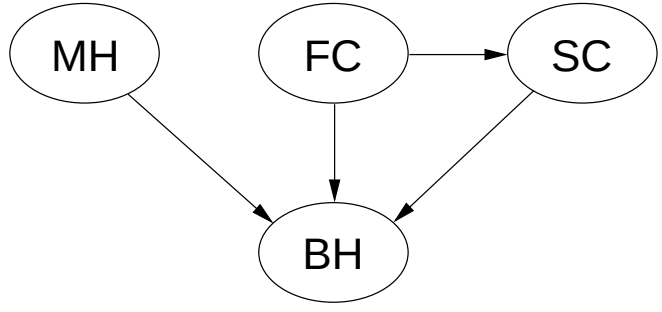
Data: existing customer records with attributes like age, employment type, income, . . . and information on payback history.
Task: build a classifier that predicts whether a new customer will repay the loan.

Classification using Bayesian networks

Consider the poker domain and assume that we want a classifier for predicting BH based on FC, SC, and MH.



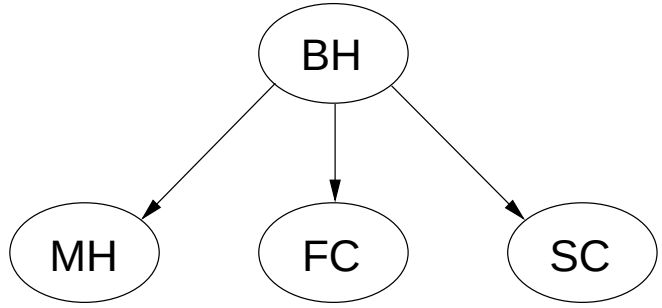
When learning the model from data (over FC, SC, MH, BH), we are likely to get:



This is not a compact model!

Naive Bayes

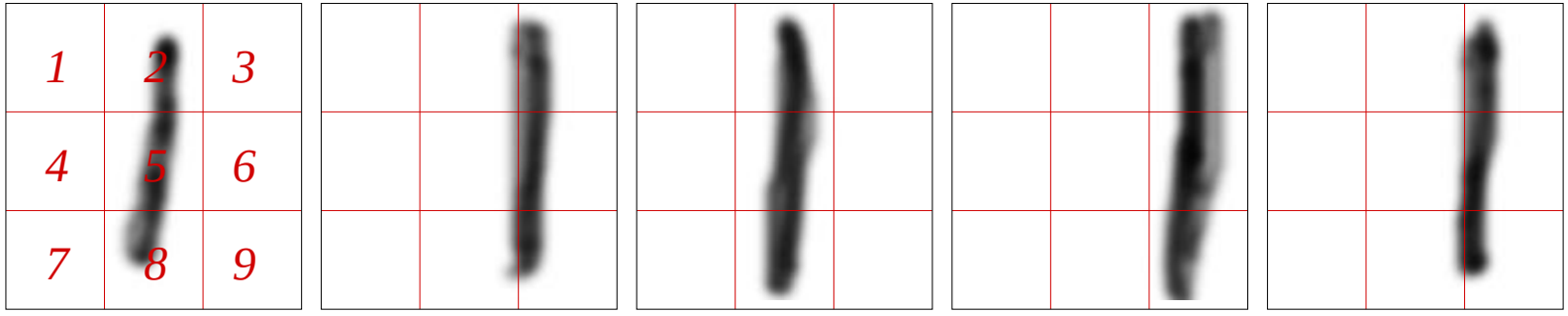
A compact type of model is the naive Bayes model:



Interpretation: Given the true class labels, the different attributes take their value independently.

Naive Bayes

The naive Bayes assumption I



For example:

$$P(\textit{Cell-2} = b \mid \textit{Cell-5} = b, \textit{Symbol} = 1) > P(\textit{Cell-2} = b \mid \textit{Symbol} = 1)$$

Attributes not independent given *Symbol*=1!

Naive Bayes

The naive Bayes assumption II

For spam example e.g.:

$$P(\text{Body'nigeria'=y} \mid \text{Body'confidential'=y}, \text{Spam=y})$$
$$\gg$$
$$P(\text{Body'nigeria'=y} \mid \text{Spam=y})$$

Attributes not independent given *Spam*=yes!

$\rightsquigarrow$ Naive Bayes assumption often not realistic. Nevertheless, Naive Bayes often successful.

Naive Bayes

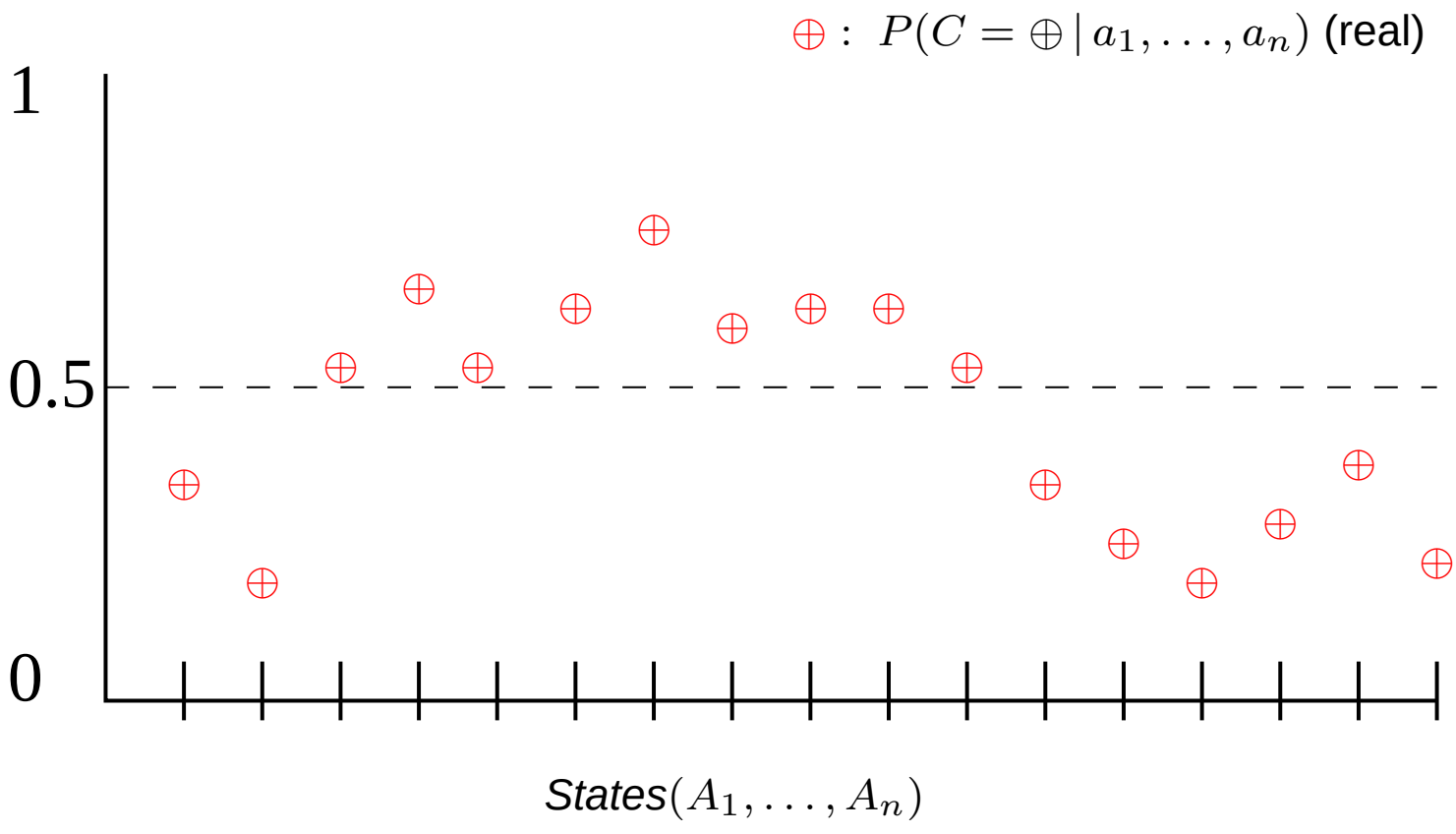
Learning a Naive Bayes Classifier

- Determine parameters $P(a_i \mid c)$ ($a_i \in \mathit{States}(A_i), c \in \mathit{States}(C)$) from empirical counts in the data.
- Missing values are easily handled: instances for which A_i is missing are ignored for $P(a_i \mid c)$.
- Discrete and continuous attributes can be mixed.

Naive Bayes

The paradoxical success of Naive Bayes

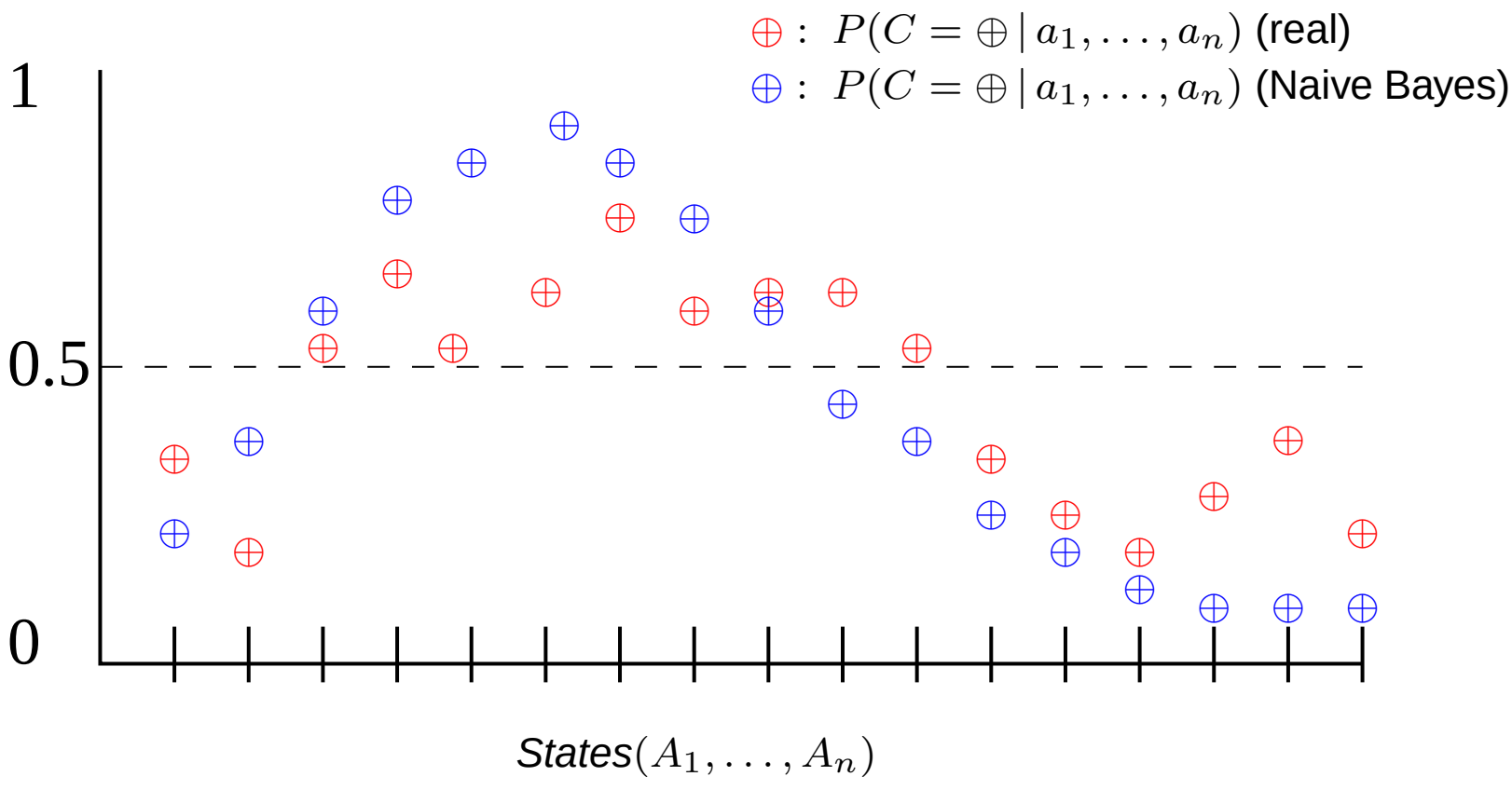
One explanation for the surprisingly good performance of Naive Bayes in many domains: do not require exact distribution for classification, only the right decision boundaries



Naive Bayes

The paradoxical success of Naive Bayes

One explanation for the surprisingly good performance of Naive Bayes in many domains: do not require exact distribution for classification, only the right decision boundaries



Naive Bayes

When Naive Bayes must fail

No Naive Bayes Classifier can produce the following classification:

A	B	Class
yes	yes	$\oplus$
yes	no	$\ominus$
no	yes	$\ominus$
no	no	$\oplus$

because assume it did, then:

1. $P(A = y \mid \oplus)P(B = y \mid \oplus)P(\oplus) > P(A = y \mid \ominus)P(B = y \mid \ominus)P(\ominus)$
2. $P(A = y \mid \ominus)P(B = n \mid \ominus)P(\ominus) > P(A = y \mid \oplus)P(B = n \mid \oplus)P(\oplus)$
3. $P(A = n \mid \ominus)P(B = y \mid \ominus)P(\ominus) > P(A = n \mid \oplus)P(B = y \mid \oplus)P(\oplus)$
4. $P(A = n \mid \oplus)P(B = n \mid \oplus)P(\oplus) > P(A = n \mid \ominus)P(B = n \mid \ominus)P(\ominus)$

Naive Bayes

Multiplying the four left sides and the four right sides of these inequalities:

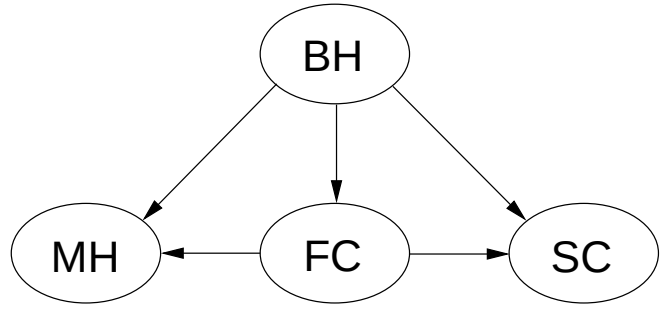
$$\prod_{i=1}^{4}(\textit{left side of } i.) > \prod_{i=1}^{4}(\textit{right side of } i.)$$

But this is false, because both products are actually equal.

Tree Augmented Naive Bayes

Model: all Bayesian network structures where

- The class node is a parent of each attribute node
- The substructure on the attribute nodes is a tree



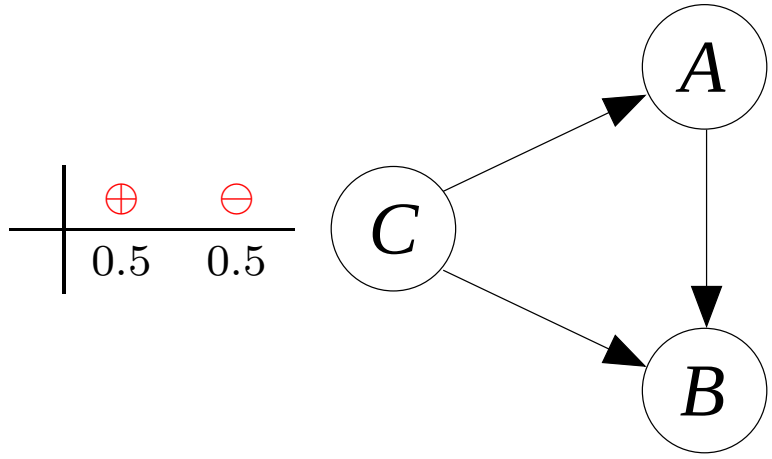
Learning TAN classifier: learning the tree structure and parameters. Optimal tree structure can be found efficiently.

Tree Augmented Naive Bayes

TAN classifier for

A	B	Class
yes	yes	$\oplus$
yes	no	$\ominus$
no	yes	$\ominus$
no	no	$\oplus$

:



$\oplus$	$\ominus$
0.5	0.5

C	yes	no
$\oplus$	0.5	0.5
$\ominus$	0.5	0.5

C	A	yes	no
$\oplus$	yes	1.0	0.0
$\oplus$	no	0.0	1.0
$\ominus$	yes	0.0	1.0
$\ominus$	no	1.0	0.0

Evaluating Classifiers

Validation

Evaluation: estimate of the performance of a classifier on future data. Estimate obtained by measuring performance on *validation set* (distinct from test set used for parameter tuning!); or by cross-validation.

Classification Error

Classifier $\mathcal{C}$ is used to classify instances $\mathbf{a}_1, \ldots, \mathbf{a}_N$ with true class labels $c_1, \ldots, c_N$. Class labels assigned by $\mathcal{C}$: $c'_1, \ldots, c'_N$. Classification error:

$$|\{i \in 1, \ldots, N \mid c_i \neq c'_i\}|/N$$

Evaluating Classifiers

Expected Loss

A more detailed picture is provided by the *confusion matrix* and a *cost function*:

		Pred		
		me	draw	op
True	me	0.25	0.05	0.1
	draw	0.05	0.1	0.05
	op	0.05	0.1	0.25

Confusion matrix

		Pred		
		me	draw	op
True	me	0	0	-1
	draw	0	0	-1
	op	-3	-1	0

Loss matrix

Expected Loss:

$$\sum_{\text{True,Pred.}} \textit{Confusion}(\text{True}, \text{Pred}) \cdot \textit{Loss}(\text{True}, \text{Pred}) = -0.4.$$

When cost function given, try to minimize expected loss (minimizing classification error is special case for *0-1 loss*: $\textit{Loss}(x, x) = 0$ and $\textit{Loss}(x, y) = 1$ for $x \neq y$)!