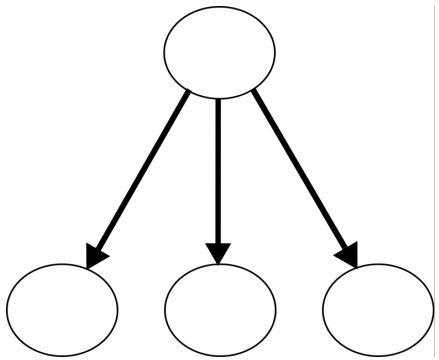
Probabilistic Graphical Models: Principles and Applications

Chapter 4: BAYESIAN CLASSIFIERS

L. Enrique Sucar, INAOE



Outline

Classification

Classification consists in assigning classes or labels to objects. There are two basic types of classification problems:

Unsupervised: in this case the classes are unknown, so the problem consists in dividing a set of objects into *n* groups or clusters, so that a class is assigned to each different group. It is also known as *clustering*.

Supervised: the possible classes or labels are known *a priori*, and the problem consists in finding a function or rule that assigns each object to one of the classes.

Probabilistic Classification

- Supervised classification consists in assigning to a particular object described by its attributes, $A_1, A_2, ..., A_n$, one of m classes, $C = \{c_1, c_2, ..., c_m\}$, such that the probability of the class given the attributes is maximized:

$$Arg_C[MaxP(C \mid A_1, A_2, ..., A_n)] \quad (1)$$

- If we denote the set of attributes as $\mathbf{A} = \{A_1, A_2, ..., A_n\}$: $Arg_C[MaxP(C \mid \mathbf{A})]$

Classifier Evaluation

Accuracy: it refers to how well a classifier predicts the correct class for unseen examples (that is, those not considered for learning the classifier).

Classification time: how long it takes the classification process to predict the class, once the classifier has been trained.

Training time: how much time is required to learn the classifier from data.

Memory requirements: how much space in terms of memory is required to store the classifier parameters.

Clarity: if the classifier is easily understood by a person.

Cost Imbalance

- In general we want to maximize the classification accuracy; however, this is only *optimal* if the cost of a wrong classification is the same for all the classes
- When there is imbalance in the costs of misclassification, we must then minimize the expected cost (EC). For two classes, this is given by:

$$EC = FN \times P(-)C(- \mid +) + FP \times P(+)C(+ \mid -) \quad (2)$$

Where: FN is the false negative rate, FP is the false positive rate, $P(+)$ is the probability of positive, $P(-)$ is the probability of negative, $C(- \mid +)$ is the cost of classifying a positive as negative, and $C(+ \mid -)$ is the cost of classifying a negative as positive

Bayes Classifier (I)

- Applying Bayes rule:

$$P(C \mid A_1, A_2, ..., A_n) = \frac{P(C)P(A_1, A_2, ..., A_n \mid C)}{P(A_1, A_2, ..., A_n)} \quad (3)$$

- Which can be written more compactly as:

$$P(C \mid \mathbf{A}) = P(C)P(\mathbf{A} \mid C)/P(\mathbf{A}) \quad (4)$$

Bayes Classifier (II)

- The classification problem can be formulated as:

$$Arg_C[Max[P(C \mid \mathbf{A}) = P(C)P(\mathbf{A} \mid C)/P(\mathbf{A})]] \quad (5)$$

- Equivalently:
  - $Arg_C[Max[P(C)P(\mathbf{A} \mid C)]]$
  - $Arg_C[Max[log(P(C)P(\mathbf{A} \mid C))]]$
  - $Arg_C[Max[(logP(C) + logP(\mathbf{A} \mid C)]]$

  Note that the probability of the attributes, $P(\mathbf{A})$, does not vary with respect to the class, so it can be considered as a constant for the maximization.

Complexity

- The direct application of the Bayes rule results in a computationally expensive problem
- The number of parameters in the likelihood term, $P(A_1, A_2, ..., A_n \mid C)$, increases exponentially with the number of attributes
- An alternative is to consider some independence properties as in graphical models, in particular that all attributes are independent given the class, resulting in the *Naive Bayesian Classifier*

Naive Bayes

- The naive or simple Bayesian classifier (NBC) is based on the assumption that all the attributes are independent given the class variable:

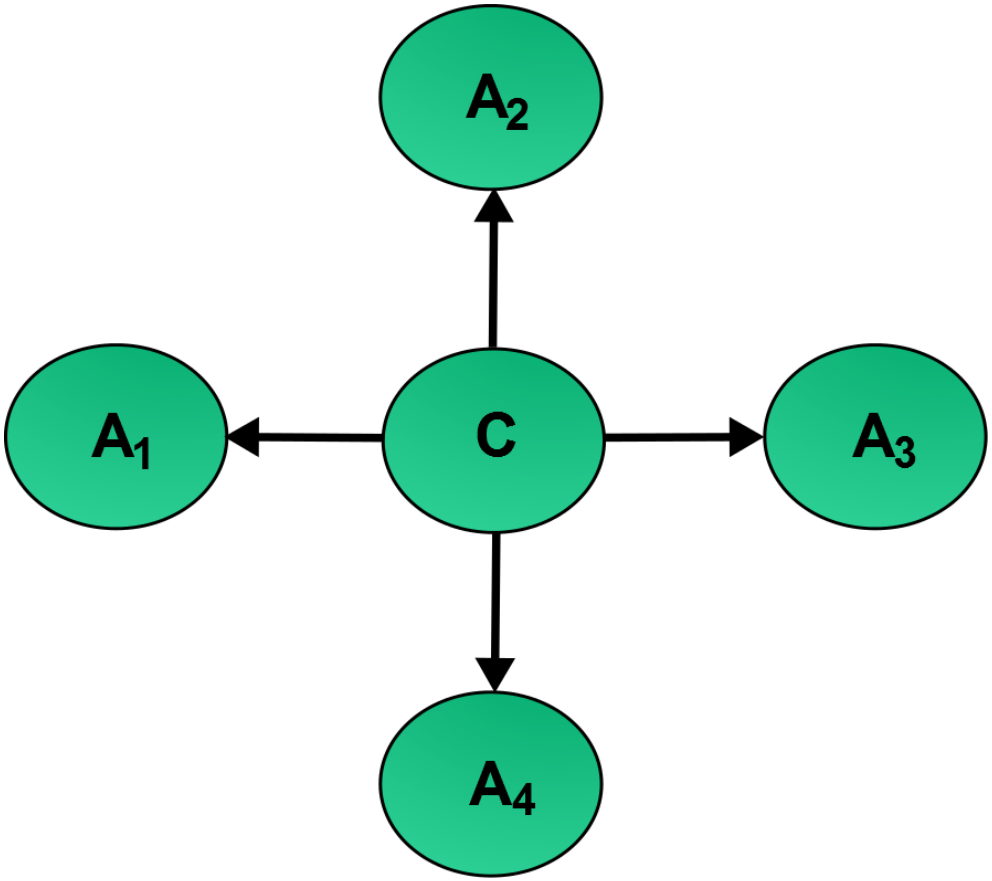
$$P(C \mid A_1, A_2, ..., A_n) = \frac{P(C)P(A_1 \mid C)P(A_2 \mid C)...P(A_n \mid C)}{P(\mathbf{A})} \tag{6}$$

where $P(\mathbf{A})$ can be considered, as mentioned before, a normalization constant.

Complexity

- The naive Bayes formulation drastically reduces the complexity of the Bayesian classifier, as in this case we only require the prior probability (one dimensional vector) of the class, and the n conditional probabilities of each attribute given the class (two dimensional matrices)
- The space requirement is reduced from exponential to linear in the number of attributes
- The calculation of the posterior is greatly simplified, as to estimate it (unnormalized) only n multiplications are required

NBC: Graphical Model



Parameter Learning

- The probabilities can be estimated from data using, for instance, maximum likelihood estimation
- The prior probabilities of the class variable, C, are given by:

$$P(c_i) \sim N_i/N \tag{7}$$

- The conditional probabilities of each attribute, A_j can be estimated as:

$$P(A_{jk} \mid c_i) \sim N_{jki}/N_i \tag{8}$$

Inference

- The posterior probability can be obtained just by multiplying the prior by the likelihood for each attribute
- Given the values for m attributes, $a_1, ...a_m$, for each class c_i, the posterior is proportional to:

$$P(c_i \mid a_1, ...a_m) \sim P(c_i)P(a_1 \mid c_i)...P(a_m \mid c_i) \quad (9)$$

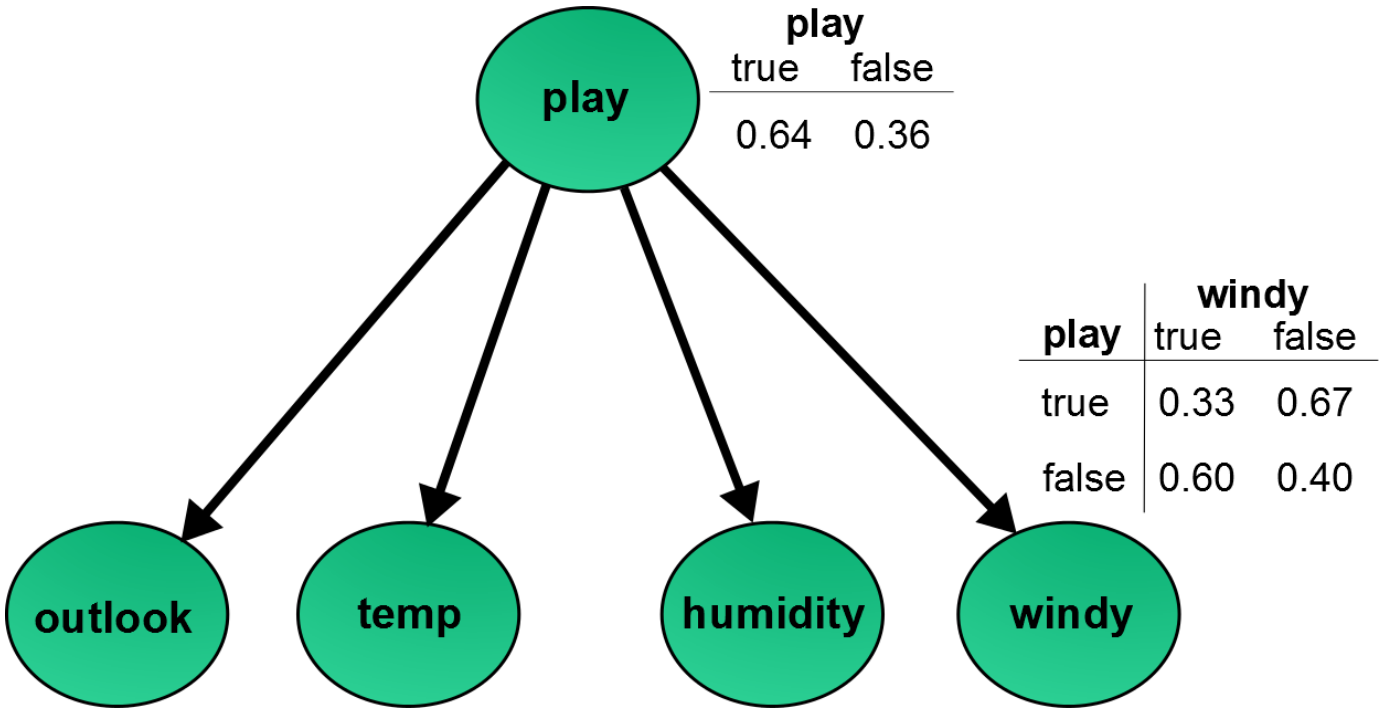
- The class c_k that maximizes the previous equation will be selected (for equal costs)

Example - Classifier for Golf

Outlook	Temperature	Humidity	Windy	Play
sunny	high	high	false	no
sunny	high	high	true	no
overcast	high	high	false	yes
rain	medium	high	false	yes
rain	low	normal	false	yes
rain	low	normal	true	no
overcast	low	normal	true	yes
sunny	medium	high	false	no
sunny	low	normal	false	yes
rain	medium	normal	false	yes
sunny	medium	normal	true	yes
overcast	medium	high	true	yes
overcast	high	normal	false	yes
rain	medium	high	true	no

Example - NBC for Golf

Graphical model and some parameters:



Example - inference for Golf

Given:
Outlook=rain, Temperature=high, Humidity=normal, Windy=no

$$P(Play = yes \mid overcast, medium, normal, no) =$$
$$k \times 0.64 \times 0.33 \times 0.22 \times 0.67 \times 0.67 = k \times 0.021$$
$$P(Play = no \mid overcast, medium, normal, no) =$$
$$k \times 0.36 \times 0.40 \times 0.40 \times 0.2. \times 0.40 = k \times 0.0046$$

$$k = 1/(0.021 + 0.0046) = 39.27$$
$$P(Play = yes \mid overcast, medium, normal, no) = 0.82$$
$$P(Play = no \mid overcast, medium, normal, no) = 0.18$$

Analysis

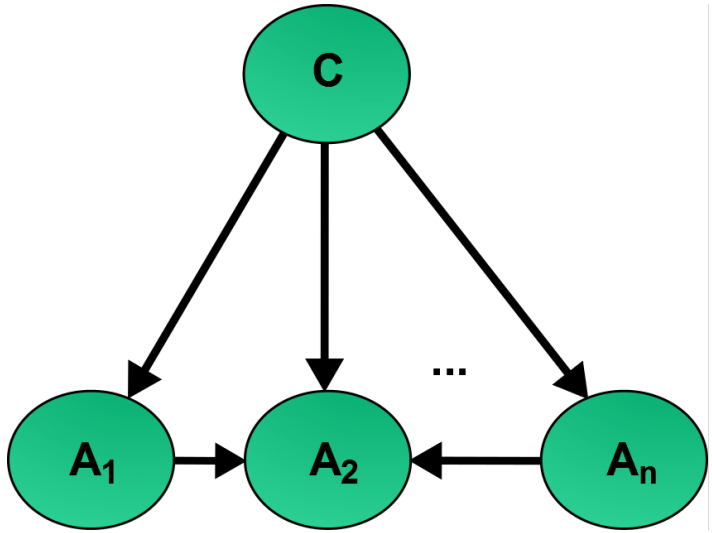
- Advantages:
  - The low number of required parameters, which reduces the memory requirements and facilitates learning them from data.
  - The low computational cost for inference (estimating the posteriors) and learning.
  - The relatively good performance (classification precision) in many domains.
  - A simple and intuitive model.
- Limitations:
  - In some domains, the performance is reduced given that the conditional independence assumption is not valid.
  - If there are continuous attributes, these need to be discretized (or consider alternative models such as the linear discriminator).

Alternative Bayesian Classifiers

- *Tree augmented Bayesian Classifier* or TAN
- *Bayesian Network augmented Bayesian Classifier* or BAN
- Semi–Naive Bayesian Classifiers

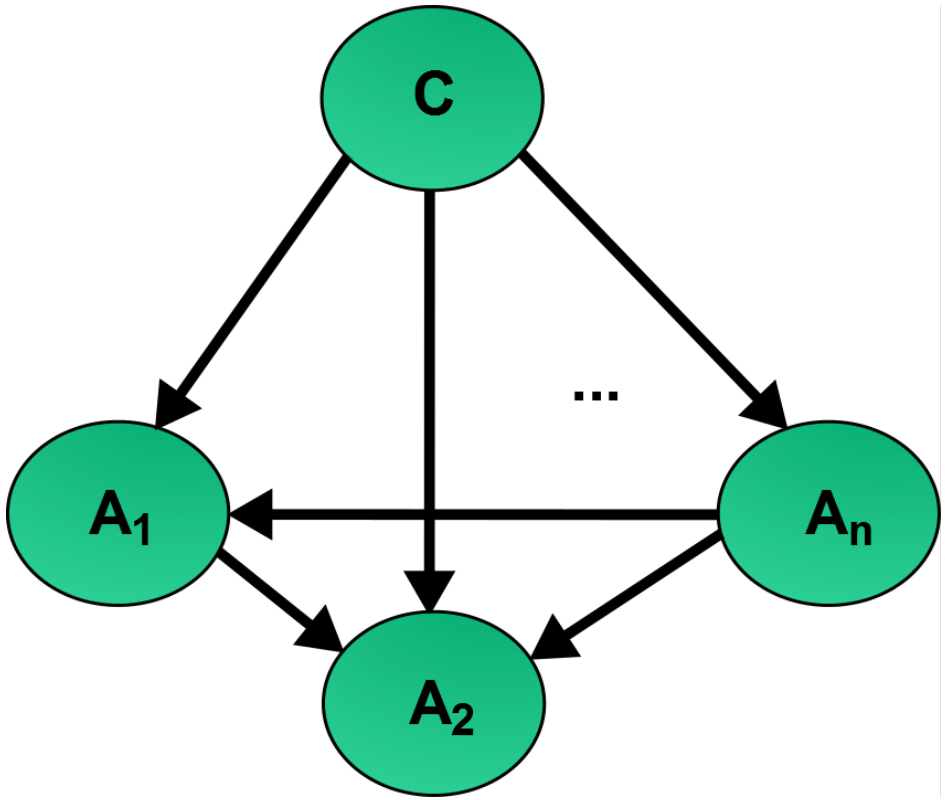
TAN

- The *Tree augmented Bayesian Classifier*, or TAN, incorporates some dependencies between the attributes by building a directed tree among the attribute variables
- The *n* attributes form a graph which is restricted to a directed tree that represents the dependency relations between the attributes



BAN

- The *Bayesian Network augmented Bayesian Classifier*, or BAN, which considers that the dependency structure among the attributes constitutes a directed acyclic graph (DAG)



Inference and Learning

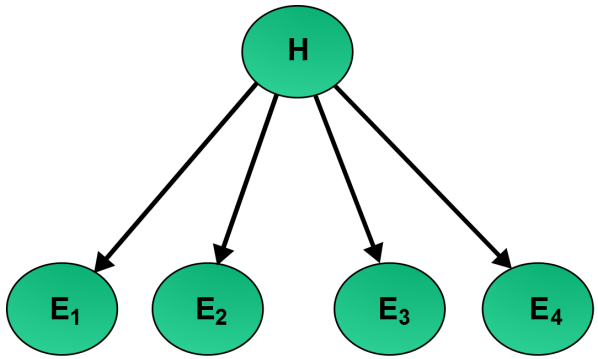
- The TAN and BAN classifiers can be considered as particular cases of a more general model, that is, Bayesian networks (see Chapter 7)
- Bayesian networks' inference and learning techniques can be applied to obtain the posterior probabilities (inference) and the model (learning) for the TAN and BAN classifiers

Semi-Naive Bayesian Classifiers (SNBC)

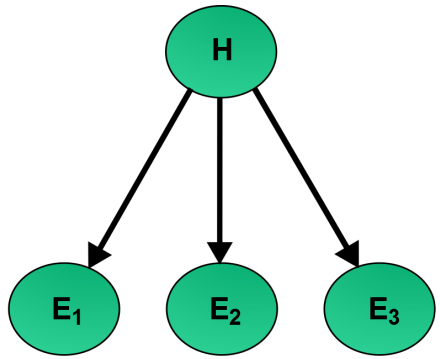
- Another alternative to deal with dependent attributes is to transform the basic structure of a naive Bayesian classifier, while maintaining a star or tree-structured network
- The basic idea of the SNBC is to eliminate or *join* attributes which are not independent given the class
- This is analogous to *feature selection* in machine learning

Structural Improvement

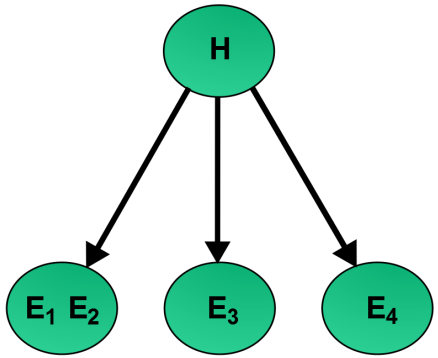
- Two alternative operations to modify the structure of a NBC: (i) node elimination, and (ii) node combination, considering that we start from a full structure



(a) Initial structure



(b) Elimination of E_4



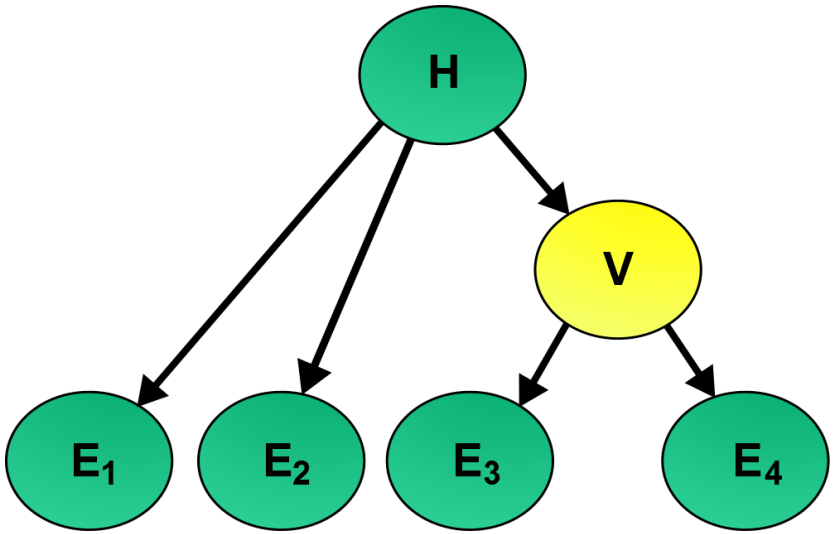
(c) Join E_1, E_2

Node elimination and combination

- Node elimination consists in simply eliminating an attribute, A_i, from the model, this could be because it is not relevant for the class (A_i and C are independent); or because the attribute A_i and another attribute, A_j, are not independent given the class
- Node combination consists in merging two attributes, A_i and A_j, into a new attribute A_k, such that A_k has as possible values the cross product of the values of A_i and A_j (assuming discrete attributes). For example, if $A_i = a, b, c$ and $A_j = 1, 2$, then $A_k = a1, a2, b1, b2, c1, c2$. This is an alternative when two attributes are not independent given the class

Node Insertion

- A third alternative when two attributes are not independent given the class, consists in adding a new attribute that makes them independent
- This new attribute is a kind of virtual or hidden node in the model, for which we do not have any data
- An alternative for estimating the parameters of hidden variables in Bayesian networks, such as in this case, is based on the Expectation-Maximization (EM) procedure (see Chapter 7)



Multidimensional classification

- Several important problems need to predict several classes simultaneously
- For example: text classification, where a document can be assigned to several topics; gene classification, as a gene may have different functions; image annotation, as an image may include several objects

Definition

- The *multi-dimensional classification* problem corresponds to searching for a function h that assigns to each instance represented by a vector of m features $\mathbf{X} = (X_1, \ldots, X_m)$ a vector of d class values $\mathbf{C} = (C_1, \ldots, C_d)$:

$$ArgMax_{c_1,\ldots,c_d} P(C_1 = c_1, \ldots, C_d = c_d | \mathbf{X}) \qquad (10)$$

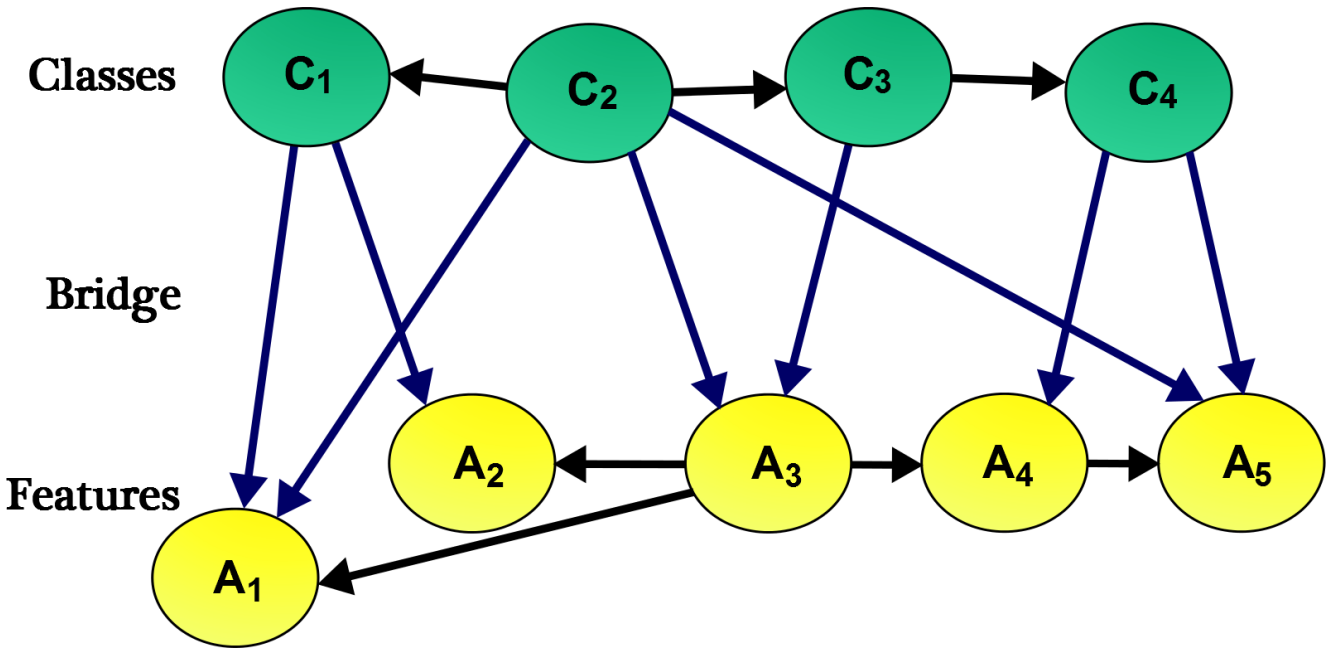
Multi-label classification

- *Multi-label* classification is a particular case of multi-dimensional classification, where all class variables are binary
- Two basic approaches:
  - The **binary relevance** approach transforms the multi-label classification problem into d independent binary classification problems, one for each class variable, $C_1, \ldots, C_d$ (does not consider dependence between classes)
  - The **label power-set** approach transforms the multi-label classification problem into a single-class scenario by defining a new compound class variable whose possible values are all the possible combinations of values of the original classes (too complex when there are many classes)

Multidimensional Bayesian Network Classifiers

- A multidimensional Bayesian network classifier (MDBNC) s a Bayesian network with a particular structure
- The set $\mathbf{V}$ of variables is partitioned into two sets $\mathbf{V_C} = \{\mathbf{C_1}, \ldots, \mathbf{C_d}\}$, $d \geq 1$, of class variables and $\mathbf{V_X} = \{\mathbf{X_1}, \ldots, \mathbf{X_m}\}$, $m \geq 1$, of feature variables $(d + m = n)$
- The set $\mathbf{A}$ of arcs is also partitioned into three sets, $\mathbf{A_C}$, $\mathbf{A_X}$, $\mathbf{A_{CX}}$, such that $\mathbf{A_C} \subseteq \mathbf{V_C} \times \mathbf{V_C}$ is composed of the arcs between the class variables, $\mathbf{A_X} \subseteq \mathbf{V_X} \times \mathbf{V_X}$ is composed of the arcs between the feature variables, and finally, $\mathbf{A_{CX}} \subseteq \mathbf{V_C} \times \mathbf{V_X}$ is composed of the arcs from the class variables to the feature variables

MDBNC - Graphical Model



Inference

- The problem of obtaining the classification of an instance with a MDBNC, that is, the most likely combination of classes, corresponds to the MPE (Most Probable Explanation) or *abduction* problem
- This is a complex problem with a high computational cost (see Chapters 7 and 8)

Chain Classifiers

- Chain classifiers are an alternative method for multi-label classification that incorporate class dependencies, while keeping the computational efficiency of the binary relevance approach
- A chain classifier consists of d base binary classifiers which are linked in a chain, such that each classifier incorporates the classes predicted by the previous classifiers as additional attributes
- The feature vector for each binary classifier, L_i, is extended with the labels $(0/1)$ for all previous classifiers in the chain

Chain Classifiers - example

Rock	Pop	Latin	E1	E2	E3
1	1	0	...	...	...
1	1	1	...	...	...
0	0	1	...	...	...

Learning and Inference

- Each classifier in the chain is trained to learn the association of label l_i given the features augmented with all previous class labels in the chain
- For classification, it starts at L_1, and propagates the predicted classes along the chain, such that for $L_i \in \mathcal{L}$ (where $\mathcal{L} = \{L_1, L_2, \ldots, L_d\}$) it predicts $P(L_i \mid \mathbf{X}, L_1, L_2, \ldots, L_{i-1})$
- The class vector is determined by combining the outputs of all the binary classifiers

Bayesian chain classifiers

- *Bayesian chain classifiers* are a type of chain classifier under a probabilistic framework:

$$ArgMax_{C_1,\ldots,c_d} P(C_1|C_2,\ldots,C_d,\mathbf{X})\ldots P(C_d|\mathbf{X}) \quad (11)$$

- If we consider the dependency relations between the class variables, and represent these relations as a directed acyclic graph (DAG), then we can simplify the previous Equation:

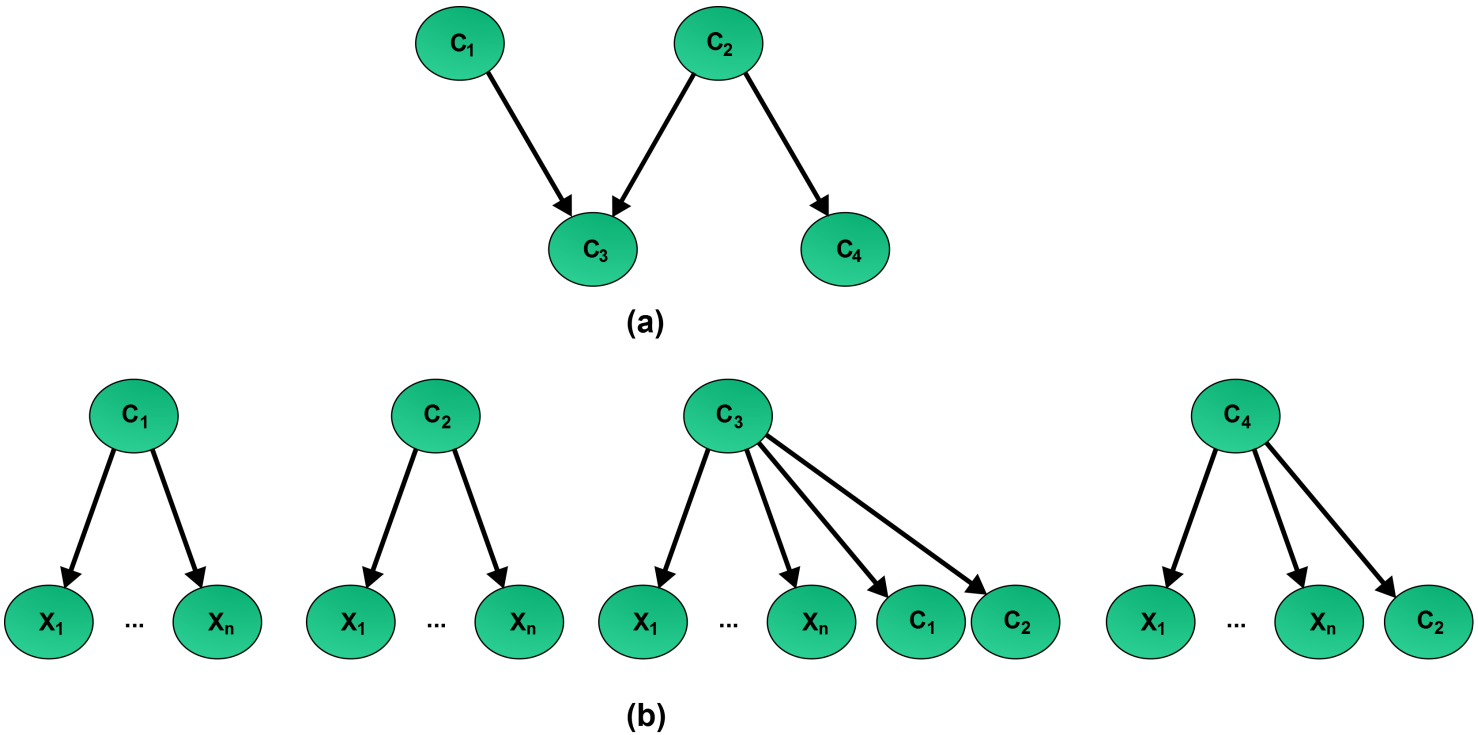
$$ArgMax_{C_1,\ldots,C_d} \prod_{i=1}^{d} P(C_i|\mathbf{Pa}(C_i),\mathbf{x}) \quad (12)$$

- A further simplification is obtained by assuming that the most probable joint combination of classes can be approximated by concatenating the individual most probable classes

$$
\begin{aligned}
&ArgMax_{C_1} P(C_1|\mathbf{Pa}(C_1), \mathbf{X}) \\
&ArgMax_{C_2} P(C_2|\mathbf{Pa}(C_2), \mathbf{X}) \\
&\qquad\qquad \ldots\ldots\ldots\ldots\ldots\ldots \\
&ArgMax_{C_d} P(C_d|\mathbf{Pa}(C_d), \mathbf{X})
\end{aligned}
$$

- This last approximation corresponds to a Bayesian chain classifier (BCC)
- For the *base* classifier we can use any of the Bayesian classifiers presented in the previous sections, for instance a NBC.

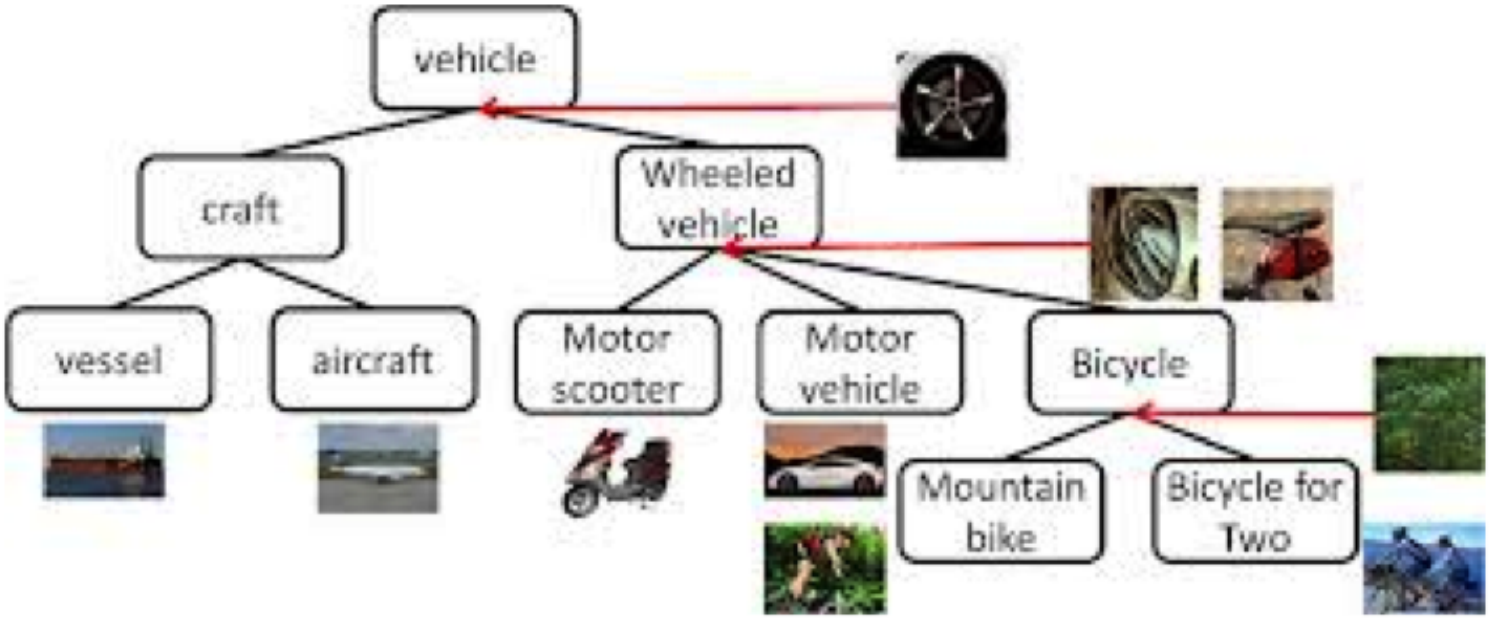
BCC: Graphical Model



Hierarchical Classification

- Hierarchical classification is a type of multidimensional classification in which the classes are ordered in a predefined structure, typically a tree, or in general a directed acyclic graph (DAG)
- In hierarchical classification, an example that belongs to certain class automatically belongs to all its superclasses
- Hierarchical classification has application in several areas, such as text categorization, protein function prediction, and object recognition

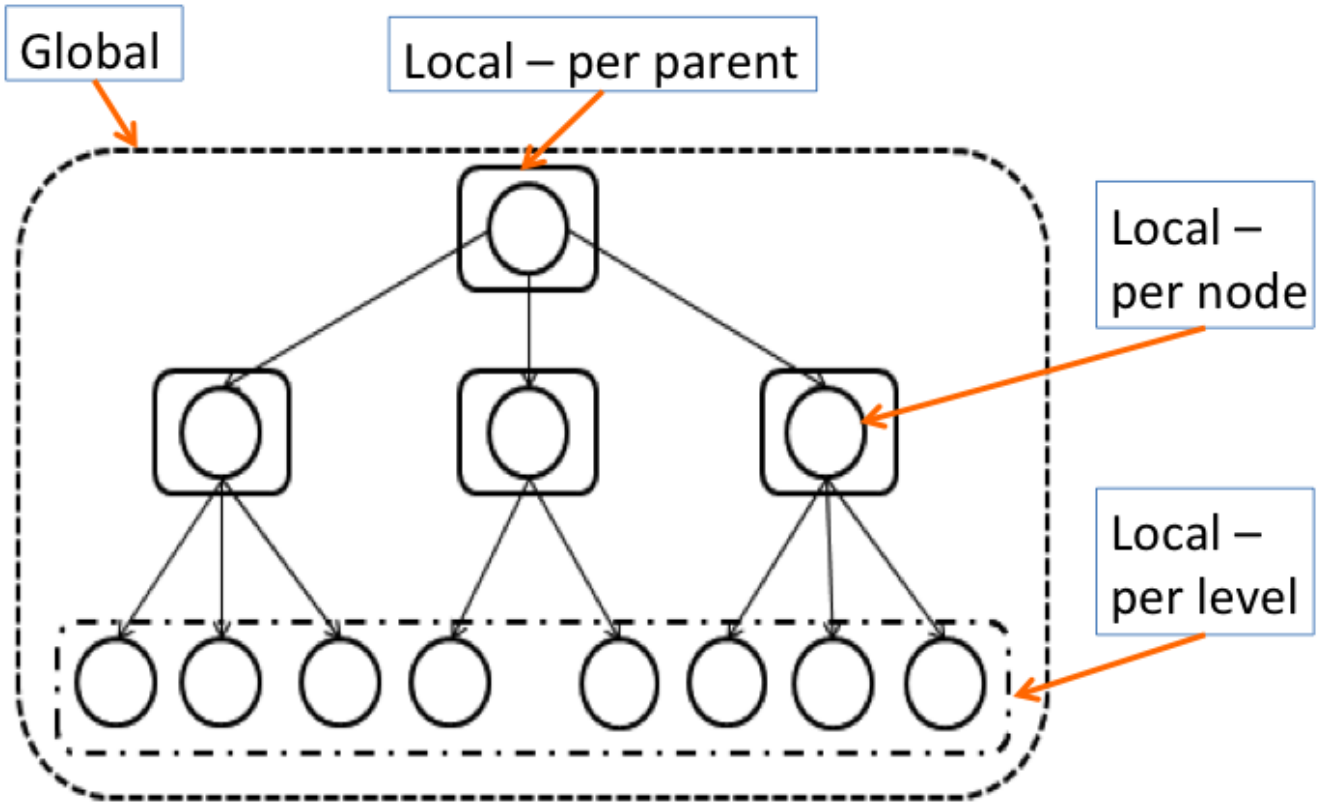
Hierarchical Classification - example



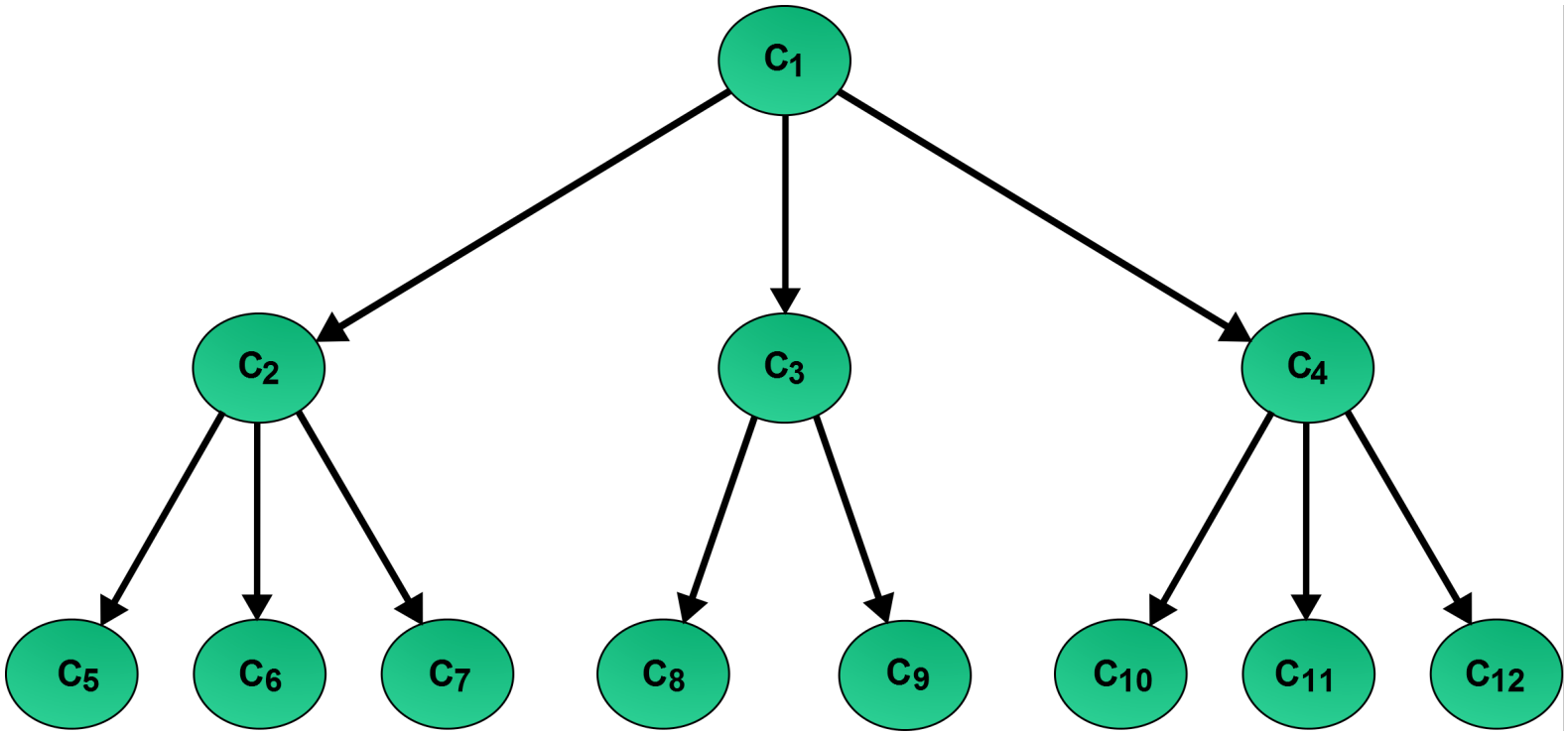
Basic approaches

- The **global approach** builds a classifier to predicts all the classes at once; this becomes too complex computationally for large hierarchies.
- The **local approaches** train several classifiers and combine their outputs:
  - Local Classifier per Hierarchy Level, that trains one multi-class classifier for each level of the class hierarchy
  - Local binary Classifier per Node, in which a binary classifier is built for each node (class) in the hierarchy, except the root node
  - Local Classifier per Parent Node (LCPN), where a multi-class classifier is trained to predict its child nodes
- Local methods commonly use a top-down approach for classification

Types of Methods



Hierarchical Structure



Chained path evaluation

- A disadvantage of the top–down approaches is that an error in the top levels of the hierarchy is propagated down and can not be recovered; an alternative is to evaluate each path in the hierarchy
- **Chained Path Evaluation** (CPE) analyzes each possible path from the root to a leaf node in the hierarchy, taking into account the level of the predicted labels to give a score to each path and finally return the one with the best score
- Additionally, it considers the relations of each node with its ancestors in the hierarchy, based on chain classifiers

Training

- A local classifier is trained for each node, C_i, in the hierarchy, except the leaf nodes, to classify its child nodes
- The classifier for each node, C_i, for instance a naive Bayes classifier, is trained considering examples from all it child nodes, as well as some examples of it sibling nodes in the hierarchy
- To consider the relation with other nodes in the hierarchy, the class predicted by the parent (tree structure) or parents (DAG), are included as additional attributes in each local classifier

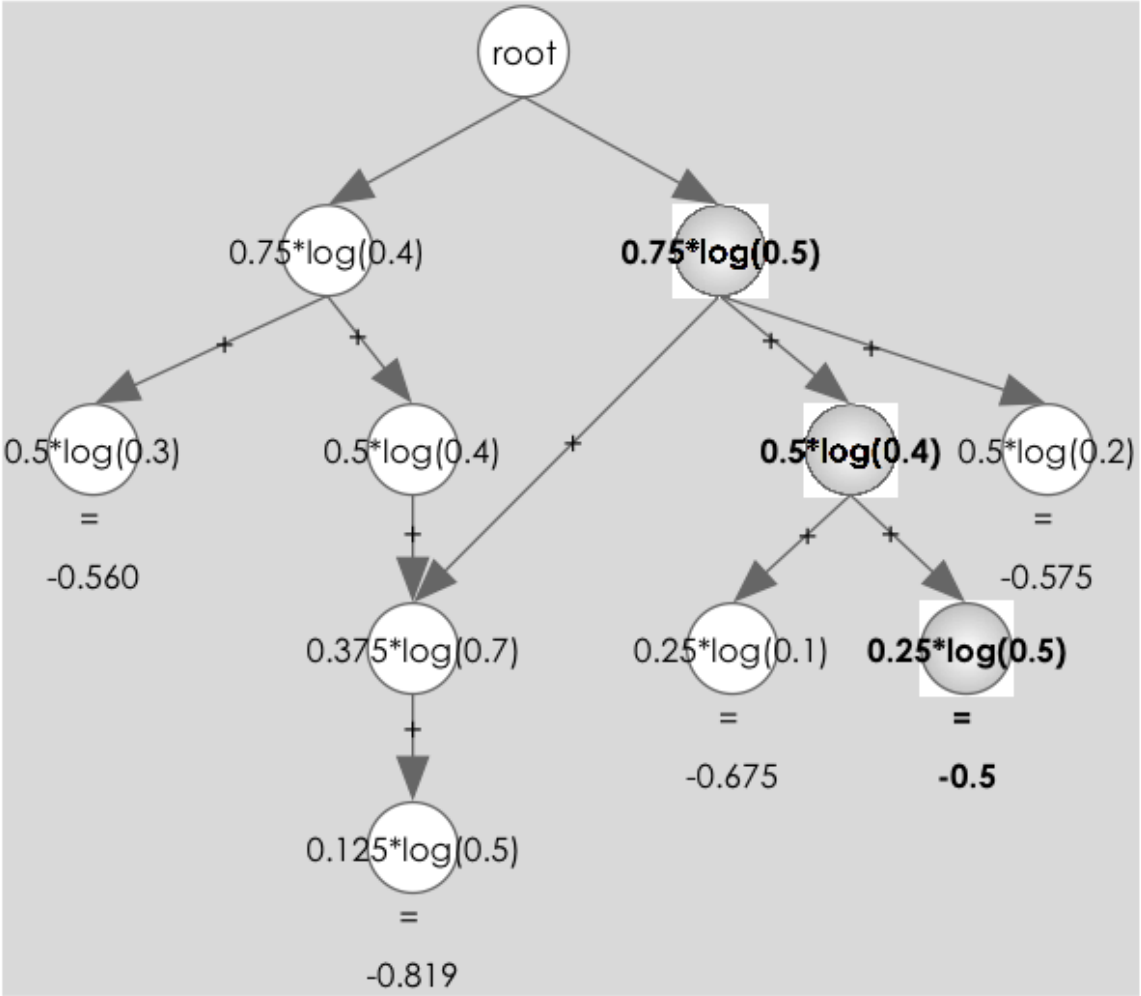
Classification

- In the classification phase, the probabilities for each class for all local classifiers are obtained based on the input data
- The *score* for each path in the hierarchy is calculated by a weighted sum of the log of the probabilities of all local classifiers in the path:

$$score = \sum_{i=0}^{n} w_{C_i} \times \log(P(C_i|X_i, pa(C_i))) \qquad (13)$$

- The purpose of these weights is to give more importance to the upper levels of the hierarchy
- Once the scores for all the paths are obtained, the path with the highest score will be selected as the set of classes corresponding to certain instance

CPE - Classification Example



Application Examples

- Visual skin detection: semi-naive Bayesian classifiers
- HIV drug selection: multidimensional Bayesian network classification

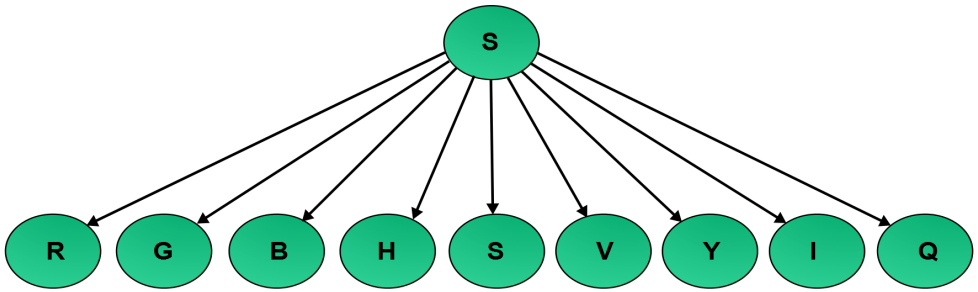
Visual skin detection

- Skin detection is a useful pre-processing stage for many applications in computer vision, such as person detection and gesture recognition, among others
- A simple and very fast way to obtain an approximate classification of pixels in an image as *skin* or $not - skin$ is based on the color attributes of each pixel
- Usually, pixels in a digital image are represented as the combination of three basic (primary) colors: Red (R), Green (G) and Blue (B), in what is known as the *RGB* model. There are alternative color models, such as *HSV*, *YIQ*, etc.

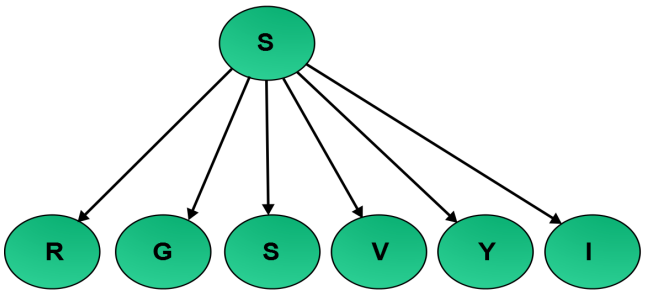
SNBC

- An alternative is to consider a semi-naive Bayesian classifier and *select* the best attributes from the different color models for skin classification by eliminating or joining attributes
- Then an initial NBC was learned based on data - examples of skin and not-skin pixels taken from several images. This initial classifier obtained a 94% accuracy when applied to other (test) images.
- The classifier was then *optimized*. Starting from the *full* NBC with 9 attributes, the method applies the variable elimination and combination stages until the *simplest* classifier with maximum accuracy is obtained. With this final model accuracy improved to 98%.

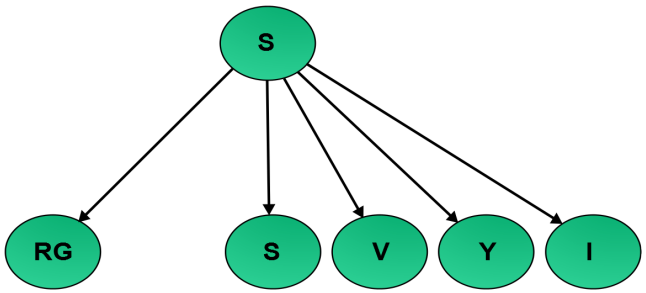
Optimization



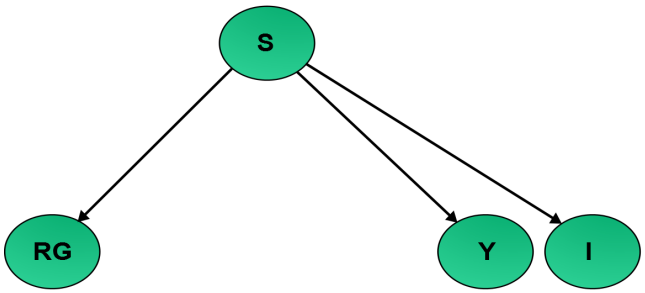
(a) Initial structure with color models RGB, HSV and YIQ



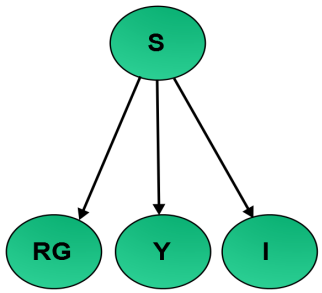
(b) Eliminate B, Q and H



(c) Join R and G



(d) Eliminate V and S



(e) Final model

Example

(a) Original image

(b) Image with skin pixels detected

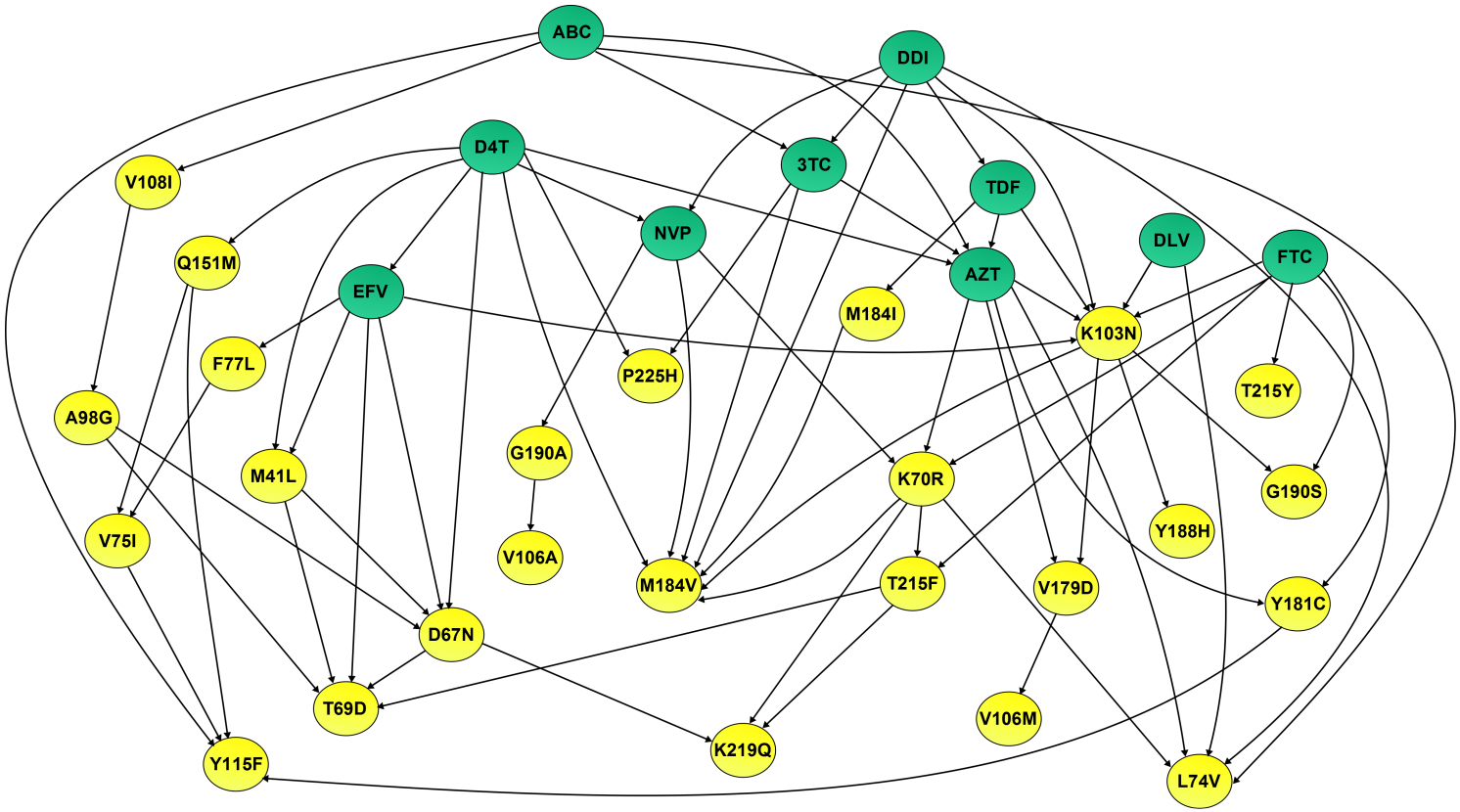
HIV drug selection

- The Human Immunodeficiency Virus (HIV) is the causing agent of AIDS, a condition in which progressive failure of the immune system allows opportunistic life-threatening infections to occur
- The Human Immunodeficiency Virus (HIV) is the causing agent of AIDS, a condition in which progressive failure of the immune system allows opportunistic life-threatening infections to occur
- To combat HIV infection several antiretroviral (ARV) drugs belonging to different drug classes that affect specific steps in the viral replication cycle have been developed. It is important to select the best drug combination according to the virus' mutations in a patient.

Multilabel Classification

- Selecting the best group of antiretroviral drugs for a patient can be seen as an instance of a multi-label classification problem
- This particular problem can be accurately modeled with a multi-dimensional Bayesian network classifiers
- This model can be learned from data and then use it for selecting the set of drugs according to the features (virus mutations)

MBC for HIV drug selection



Book

Sucar, L. E, *Probabilistic Graphical Models*, Springer 2015 – Chapter 4

Additional Reading – Bayesian classifiers / Semi-naive Bayesian classifiers


Cheng, J., Greiner, R.: Comparing Bayesian Network Classifiers. Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence, 101–108 (1999)


Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian Network Classifiers. Machine Learning **29**, 131–163 (1997)


Kwoh, C.K, Gillies, D.F.: Using Hidden Nodes in Bayesian Networks. Artificial Intelligence **88**, Elsevier, Essex, UK, 1–38 (1996)


Martinez M., Sucar, L.E.: Learning an optimal naive Bayes classifier. International Conference on Pattern Recognition (ICPR), Vol. 3, 1236–1239 (2006)

Additional Reading – Multidimensional and Hierarchical Classifiers (I)


Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier chains for multi-label classification. Proceedings ECML/PKDD, 254–269 (2009)


Bielza, C., Li, G., Larrañaga, P.: Multi-Dimensional Classification with Bayesian Networks. International Journal of Approximate Reasoning, (2011)


Borchani, H., Bielza, C., Toro, C., Larrañaga, P.: Predicting human immunodeficiency virus inhibitors using multi-dimensional Bayesian network classifiers. Artificial Intelligence in Medicine **57**, 219–229 (2013)


Ramírez, M., Sucar, L.E., Morales, E.: Path Evaluation for Hierarchical Multi-label Classification. Proceedings of the Twenty-Seventh International Florida Artificial Intelligence Research Society Conference (FLAIRS), pp. 502–507 (2014)

Additional Reading – Multidimensional and Hierarchical Classifiers (II)


Silla-Jr., C. N., Freitas, A. A.: A survey of hierarchical classification across different application domains. Data Min. and Knowledge Discovery, 22(1-2), 31–72 (2011).


Sucar, L.E, Bielza, C., Morales, E., Hernandez P., Zaragoza, J., Larrañaga, P.: Multi-label Classification with Bayesian Network-based Chain Classifiers. Pattern Recognition Letters **41**, 14–22 (2014)


Tsoumakas, G., Katakis, I.: Multi-Label Classification: An Overview. International Journal of Data Warehousing and Mining, **3**, 1–13 (2007)


van der Gaag L.C., de Waal, P.R.: Multi-dimensional Bayesian Network Classifiers. Third European Conference on Probabilistic Graphic Models, 107–114, Prague, Czech Republic (2006)