Bayesian Learning
- based on Tom Mitchell's slides -

August 24, 2015

- Bayes Theorem
- *Maximum a posteriori Probability* (MAP) hypotheses
- *Maximum likelihood*(ML) hypotheses
- MAP learners
- Minimum description length principle
- Bayes optimal classifier
- Naive Bayes learner
- Example: Learning over text data
- Bayesian belief networks
- Expectation Maximization algorithm

1 Two Roles for Bayesian Methods

Provide practical learning algorithms:

- Naive Bayes learning
- Bayesian belief network learning

- Combine prior knowledge (prior probabilities) with observed data

- Requires prior probabilities

Provide useful conceptual framework

- Provides "gold standard" for evaluating other learning algorithms

- Additional insight into Occam's razor

2 Bayes Theorem

The basic result underlying the Bayesian approach is the the Bayes Theorem.

Idea : Under certain conditions reverse the arrow $X \longrightarrow Y$ to infer something about X given knowledge of Y. Model/express $\longrightarrow$ as conditional probability.

In the learning framework, we have, as usual D the set of examples for a concept, and h a (possible) hypothesis about this concept.

Bayes Theorem states:

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

where

- $P(h)$ = prior probability of hypothesis h
- $P(D)$ = prior probability of training data D
- $P(h|D)$ = probability of h given D (posterior probability of h given the data D)
- $P(D|h)$ = probability of D given h

3 Choosing Hypotheses

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

Generally, we want the most probable hypothesis given the training data, the *Maximum a posteriori* hypothesis h_{MAP} defined as that hypothesis which maximizes the posterior probability over all possible hypotheses:

$$\begin{aligned} h_{MAP} &= \arg\max_{h \in H} P(h|D) \\ &= \arg\max_{h \in H} \frac{P(D|h)P(h)}{P(D)} \\ &= \arg\max_{h \in H} P(D|h)P(h) \end{aligned} \quad (1)$$

If assume a uniform distribution on the hypothesis space, that is, $\forall h \in H$, $P(h) = \frac{1}{|H|}$, then (**??**) can be further simplified to

$$h_{MAP} = \arg\max_{h \in H} P(D|h) \quad (2)$$

The quantity $P(D|h)$ in (**??**) is called *likelihood*, and therefore the solution to (**??**) is referred to as the *Maximum Likelihood Hypothesis*(ML), or,

$$h_{ML} = \arg\max_{h \in H} P(D|h)$$

Example 1 *(Bayes Theorem for diagnosis)*
Does patient have cancer or not?

> *A patient takes a lab test and the result comes back positive. The test returns a correct positive result in only* 98% *of the cases in which the disease is actually present, and a correct negative result in only* 97% *of the cases in which the disease is not present. Furthermore,* .008 *of the entire population have this cancer.*

$$\begin{array}{rl} P(cancer) = 0.008 & P(\neg cancer) = 0.992 \\ P(+|cancer) = 0.98 & P(-|cancer) = 0.02 \\ P(+|\neg cancer) = 0.03 & P(-|\neg cancer) = 0.97 \end{array}$$

4 Basic Formulas for Probabilities

- *Product Rule*: probability $P(A \wedge B)$ of a conjunction of two events A and B:

$$P(A \wedge B) = P(A|B)P(B) = P(B|A)P(A)$$

- *Sum Rule*: probability of a disjunction of two events A and B:

$$P(A \vee B) = P(A) + P(B) - P(A \wedge B)$$

- *Theorem of total probability*: if events $A_1, \ldots, A_n$ are mutually exclusive with $\sum_{i=1}^{n} P(A_i) = 1$, then

$$P(B) = \sum_{i=1}^{n} P(B|A_i)P(A_i)$$

5 Brute Force MAP Hypothesis Learner

1. For each hypothesis h in H, calculate the posterior probability

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

2. Output the hypothesis h_{MAP} with the highest posterior probability

$$h_{MAP} = \operatorname*{argmax}_{h \in H} P(h|D)$$

6 Relation to Concept Learning

Consider our usual concept learning task

- instance space X, hypothesis space H, training examples D
- consider the FINDS learning algorithm (outputs most specific hypothesis from the version space $VS_{H,D}$)

What would Bayes rule produce as the MAP hypothesis?

Does $FindS$ output a MAP hypothesis ?

Assume fixed set of instances $\langle x_1, \ldots, x_m \rangle$
Assume D is the set of classifications $D = \langle c(x_1), \ldots, c(x_m) \rangle$
Choose $P(D|h)$:

- $P(D|h) = 1$ if h consistent with D
- $P(D|h) = 0$ otherwise

Choose $P(h)$ to be *uniform* distribution

- $P(h) = \frac{1}{|H|}$ for all h in H

Then,

$$P(h|D) = \begin{cases} \frac{1}{|VS_{H,D}|} & \text{if } h \text{ is consistent with } D \\ \\ 0 & \text{otherwise} \end{cases}$$

7 Characterizing Learning Algorithms by Equivalent MAP Learners

Discuss Figure 6.1 in the book

- Every hypothesis consistent with D is a MAP hypothesis.

- Define *consistent learners*: a learner that outputs only a consistent hypothesis.

- Therefore: **If the probability distribution on H uniform, i.e. $p(h) = c, \forall h \in H$, and if D is noise-free (no conflicting data), and D is deterministic, that is, $P(D|h) = 1$ is D and h are consistent, otherwise $P(D|h) = 0$, then every consistent learner outputs a MAP hypothesis**.

Example 2 *Find-S outputs a MAP hypothesis, even though it* **does not explicitely manipulate probabilities.**

Other distributions for $P(D|h)$ and $P(h)$ for which the same is true: any distributions that favor more specific hypotheses: $P(h_1) \geq P(h_2)$ if h_1 is more specific *than h_2.*

Therefore, by identifying $P(D|h)$ and $P(h)$ under which a learner outputs the MAP hypothesis, the Bayesian framework is one way to characterize the *behavior of learning algorithms*, even when no explicit use is made of these probabilities.

For characterization of Find-S and CandElim in terms of the Bayesian framework we use probabilities with values 0 or 1 only (capturing, deterministic aspect and noise-free data).

8 Learning A Real Valued Function

Consider any real-valued target function f. Under certain conditions, the ML hypothesis coincides with the Least Squared Errors hypothesis.

More precisely, we have the following result:

Proposition 1 *Let the training examples* $\langle x_i, d_i \rangle$, *where* d_i *is noisy training value*

- $d_i = f(x_i) + e_i$

- e_i *is random variable (noise) drawn independently for each* x_i *according to some Gaussian distribution with mean=0*

Then the maximum likelihood hypothesis h_{ML} *is the one that minimizes the sum of squared errors:*

$$h_{ML} = \arg\min_{h \in H} \sum_{i=1}^{m} (d_i - h(x_i))^2$$

Proof:

$$\begin{aligned}
h_{ML} &= \operatorname*{argmax}_{h \in H} p(D|h) \\
&= \operatorname*{argmax}_{h \in H} \prod_{i=1}^{m} p(d_i|h) \\
&= \operatorname*{argmax}_{h \in H} \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}(\frac{d_i - h(x_i)}{\sigma})^2}
\end{aligned}$$

Maximize natural log of this instead...

$$\begin{aligned}
h_{ML} &= \operatorname*{argmax}_{h \in H} \sum_{i=1}^{m} \ln \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2} \left(\frac{d_i - h(x_i)}{\sigma} \right)^2 \\
&= \operatorname*{argmax}_{h \in H} \sum_{i=1}^{m} -\frac{1}{2} \left(\frac{d_i - h(x_i)}{\sigma} \right)^2
\end{aligned}$$

$$
\begin{aligned}
&= \underset{h \in H}{\operatorname{argmax}} \sum_{i=1}^{m} -\left(d_i - h(x_i)\right)^2 \\
&= \underset{h \in H}{\operatorname{argmin}} \sum_{i=1}^{m} \left(d_i - h(x_i)\right)^2
\end{aligned}
$$

9 Learning to Predict Probabilities

We want to learn a function

$$f : X \to 0, 1$$

and we let

$$p_0(x) = Prob(f(x) = 0),\ p_1(x) = P(f(x) = 1) = 1 - p_0(x)$$

For example,

$$X = \{x;\ x\ is\ a\ patient\ with\ symptoms\}$$

and we let, for $x \in X$

$$\begin{aligned} f(x) &= 1\ if\ x\ survives \quad (3) \\ & \quad 0\ otherwise \\ & \quad (4) \end{aligned}$$

We want to learn

$$p(x) = P(f(x) = 1)$$

Training examples

$$\langle x_i, d_i \rangle$$

where d_i is 1 or 0. For each $x \in X$, let $F_1(x)$ denote the frequency of 1s and $F_0(x)$ the frequency of 0s.

State the problem as a MAP problem:

- What is $P(D|h)$? $D = \{(x_1, d_1), \ldots, (x_m, d_m)\}$, where $d_i = f(x_i) = 0$ or 1.

Here x_i, d_i are random variables, x_i and h are independent.

Claim: In general,

$$P(x_i, d_i|h) = P(d_i|h, x_i)P(x_i|h)$$

Proof:

$$rhs = \frac{P(d, h, x)}{h, x} \times \frac{P(x, h)}{P(h)}$$

$$lfs \equiv \frac{P(x, h, d)}{P(h)}$$

Now, independence of x and h means that $P(x|h) = P(x)$ (easy to show)

Therefore,

$$P(D|h) = \prod_{i=1}^{m} P(x_i, d_i|h) = \prod_{i=1}^{m} P(d_i|h_i, x_i)P(x_i)$$

Now, $P(d_i = 1|h, x_i) = h(x_i)$, so,

$$\begin{aligned} P(d_i|h, x_i) &= h(x_i)\ if\ d_i = 1 \\ &= 1 - h(x_i)\ if\ d_i = 0 \end{aligned} \tag{5}$$

which can be rewritten as

$$P(d_i|h, x_i) = (h(x_i))^{d_i}(1 - h(x_i))^{(1-d_i)}$$

and thus,

$$P(D|h_i) = \prod_{i=1}^{m} h(x_i)^{d_i}(a - h(x_i))^{(1-d_i)}P(x_i)$$

Now, h_{ML} is the hypothesis such that

$$P(D|h_{ML}) = max_{h_i}P(D|h_i).....$$

$$h_{ML} = \underset{h \in H}{\operatorname{argmax}} \sum_{i=1}^{m} d_i \ln h(x_i) + (1 - d_i) \ln(1 - h(x_i))$$

Weight update rule for a sigmoid unit:

$$w_{jk} \leftarrow w_{jk} + \Delta w_{jk}$$

where

$$\Delta w_{jk} = \eta \sum_{i=1}^{m} (d_i - h(x_i))\ x_{ijk}$$

10 Minimum Description Length Principle

Occam's razor: prefer the shortest hypothesis

MDL: prefer the hypothesis h that minimizes

$$h_{MDL} = \operatorname*{argmin}_{h \in H} L_{C_1}(h) + L_{C_2}(D|h)$$

where $L_C(x)$ is the description length of x under encoding C

Example 3 *H = decision trees, D = training data labels*

- *$L_{C_1}(h)$ is # bits to describe tree h*

- *$L_{C_2}(D|h)$ is # bits to describe D given h*

  - *Note $L_{C_2}(D|h) = 0$ if examples classified perfectly by h. Need only describe exceptions*

- *Hence h_{MDL} trades off tree size for training errors*

$$\begin{aligned} h_{MAP} &= \arg\max_{h \in H} P(D|h)P(h) \\ &= \arg\max_{h \in H} \log_2 P(D|h) + \log_2 P(h) \\ &= \arg\min_{h \in H} -\log_2 P(D|h) - \log_2 P(h) \end{aligned} \tag{6}$$

Fact (interesting) from information theory:

The optimal (shortest expected coding length) code for an event with probability p is $-\log_2 p$ bits.

So interpret (1):

1. $-\log_2 P(h)$ is length of h under optimal code

2. $-\log_2 P(D|h)$ is length of D given h under optimal code

$\rightarrow$ prefer the hypothesis that minimizes

$$length(h) + length(misclassifications)$$

11 Most Probable Classification of New Instances

So far we've sought the most probable *hypothesis* given the data D (i.e., h_{MAP})

Given new instance x, what is its most probable *classification*?

- $h_{MAP}(x)$ is not the most probable classification!

Consider:

- Three possible hypotheses:

$$P(h_1|D) = .4,\ P(h_2|D) = .3,\ P(h_3|D) = .3$$

- Given new instance x,

$$h_1(x) = +,\ h_2(x) = -,\ h_3(x) = -$$

- What's most probable classification of x?

12 Bayes Optimal Classifier

Bayes optimal classification:

$$\arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j|h_i)P(h_i|D)$$

Example:

$$\begin{array}{lll}
P(h_1|D) = .4, & P(-|h_1) = 0, & P(+|h_1) = 1 \\
P(h_2|D) = .3, & P(-|h_2) = 1, & P(+|h_2) = 0 \\
P(h_3|D) = .3, & P(-|h_3) = 1, & P(+|h_3) = 0
\end{array}$$

therefore

$$\begin{aligned}
\sum_{h_i \in H} P(+|h_i)P(h_i|D) &= .4 \\
\sum_{h_i \in H} P(-|h_i)P(h_i|D) &= .6
\end{aligned}$$

and

$$\arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j|h_i)P(h_i|D) = -$$

13 Gibbs Classifier

Bayes optimal classifier provides best result, but can be expensive if many hypotheses.

Gibbs algorithm

1. Choose one hypothesis at random, according to $P(h|D)$

2. Use this to classify new instance

Surprising fact: Assume target concepts are drawn at random from H according to priors on H. Then:

$$E[error_{Gibbs}] \leq 2E[error_{BayesOptimal}]$$

Suppose correct, uniform prior distribution over H, then

- Pick any hypothesis from VS, with uniform probability

- Its expected error no worse than twice Bayes optimal

14 Naive Bayes Classifier

Along with decision trees, neural networks, nearest nbr, one of the most practical learning methods.

When to use

- Moderate or large training set available
- Attributes that describe instances are conditionally independent given classification

Successful applications:

- Diagnosis
- Classifying text documents

Assume target function $f : X \rightarrow V$, where each instance x described by attributes $\langle a_1, a_2 \ldots a_n \rangle$.

Most probable value of $f(x)$ is:

$$
\begin{aligned}
v_{MAP} &= \operatorname*{argmax}_{v_j \in V} P(v_j | a_1, a_2 \ldots a_n) \\
v_{MAP} &= \operatorname*{argmax}_{v_j \in V} \frac{P(a_1, a_2 \ldots a_n | v_j) P(v_j)}{P(a_1, a_2 \ldots a_n)} \\
&= \operatorname*{argmax}_{v_j \in V} P(a_1, a_2 \ldots a_n | v_j) P(v_j)
\end{aligned}
$$

Naive Bayes assumption:

$$P(a_1, a_2 \ldots a_n | v_j) = \prod_i P(a_i | v_j)$$

which gives

$$\textbf{Naive Bayes classifier: } v_{NB} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j) \prod_i P(a_i | v_j)$$

15 Naive Bayes Algorithm

Naive_Bayes_Learn($examples$)

For each target value v_j

$\hat{P}(v_j) \leftarrow$ estimate $P(v_j)$

For each attribute value a_i of each attribute a
$\hat{P}(a_i|v_j) \leftarrow$ estimate $P(a_i|v_j)$

Classify_New_Instance(x)

$$v_{NB} = \underset{v_j \in V}{\operatorname{argmax}} \hat{P}(v_j) \prod_{a_i \in x} \hat{P}(a_i|v_j)$$

16 Naive Bayes: Example

Consider *PlayTennis* again, and new instance

$$\langle Outlk = sun, Temp = cool, Humid = high, Wind = strong \rangle$$

Want to compute:

$$v_{NB} = \operatorname*{argmax}_{v_j \in V} P(v_j) \prod_i P(a_i|v_j)$$

$$P(y)\ P(sun|y)\ P(cool|y)\ P(high|y)\ P(strong|y) = .005$$

$$P(n)\ P(sun|n)\ P(cool|n)\ P(high|n)\ P(strong|n) = .021$$

$$\rightarrow v_{NB} = n$$

17 Naive Bayes: Subtleties

1. Conditional independence assumption is often violated

$$P(a_1, a_2 \ldots a_n|v_j) = \prod_i P(a_i|v_j)$$

- ...but it works surprisingly well anyway. Note that **we don't need estimated posteriors $\hat{P}(v_j|x)$ to be correct; need only that**

$$\operatorname*{argmax}_{\mathbf{v_j} \in \mathbf{V}} \hat{\mathbf{P}}(\mathbf{v_j}) \prod_{\mathbf{i}} \hat{\mathbf{P}}(\mathbf{a_i}|\mathbf{v_j}) = \operatorname*{argmax}_{\mathbf{v_j} \in \mathbf{V}} \mathbf{P}(\mathbf{v_j})\mathbf{P}(\mathbf{a_1} \ldots, \mathbf{a_n}|\mathbf{v_j})$$

- see [Domingos & Pazzani, 1996] for analysis

- Naive Bayes posteriors often unrealistically close to 1 or 0

2. what if none of the training instances with target value v_j have attribute value a_i? Then

$$\hat{P}(a_i|v_j) = 0, \text{ and...}$$

$$\hat{P}(v_j) \prod_i \hat{P}(a_i|v_j) = 0$$

Typical solution is Bayesian estimate for $\hat{P}(a_i|v_j)$

$$\hat{P}(a_i|v_j) \leftarrow \frac{n_c + mp}{n + m}$$

where

- n is number of training examples for which $v = v_j$,
- n_c number of examples for which $v = v_j$ and $a = a_i$
- p is prior estimate for $\hat{P}(a_i|v_j)$
- m is weight given to prior (i.e. number of "virtual" examples)

18 Learning to Classify Text

Why?

- Learn which news articles are of interest
- Learn to classify web pages by topic

Naive Bayes is among most effective algorithms
What attributes shall we use to represent text documents??
Target concept $Interesting?: Document \rightarrow \{+, -\}$

1. Represent each document by vector of words
   - one attribute per word position in document
2. Learning: Use training examples to estimate
   - $P(+)$
   - $P(-)$
   - $P(doc|+)$
   - $P(doc|-)$

Naive Bayes conditional independence assumption

$$P(doc|v_j) = \prod_{i=1}^{length(doc)} P(a_i = w_k|v_j)$$

where $P(a_i = w_k|v_j)$ is probability that word in position i is w_k, given v_j

one more assumption: $P(a_i = w_k|v_j) = P(a_m = w_k|v_j), \forall i, m$

LEARN_NAIVE_BAYES_TEXT($Examples, V$)

1. collect all words and other tokens that occur in Examples

- $Vocabulary \leftarrow$ all distinct words and other tokens in $Examples$

2. calculate the required $P(v_j)$ and $P(w_k|v_j)$ probability terms

- For each target value v_j in V do
  - $docs_j \leftarrow$ subset of $Examples$ for which the target value is v_j
  - $P(v_j) \leftarrow \frac{|docs_j|}{|Examples|}$
  - $Text_j \leftarrow$ a single document created by concatenating all members of $docs_j$
  - $n \leftarrow$ total number of words in $Text_j$ (counting duplicate words multiple times)
  - for each word w_k in $Vocabulary$
    * $n_k \leftarrow$ number of times word w_k occurs in $Text_j$
    * $P(w_k|v_j) \leftarrow \frac{n_k+1}{n+|Vocabulary|}$

CLASSIFY_NAIVE_BAYES_TEXT(Doc)

- $positions \leftarrow$ all word positions in Doc that contain tokens found in $Vocabulary$
- Return v_{NB}, where

$$v_{NB} = \operatorname*{argmax}_{v_j \in V} P(v_j) \prod_{i \in positions} P(a_i|v_j)$$

19 Twenty NewsGroups

Given 1000 training documents from each group

Learn to classify new documents according to which newsgroup it came from

comp.graphics	misc.forsale
comp.os.ms-windows.misc	rec.autos
comp.sys.ibm.pc.hardware	rec.motorcycles
comp.sys.mac.hardware	rec.sport.baseball
comp.windows.x	rec.sport.hockey
alt.atheism	sci.space
soc.religion.christian	sci.crypt
talk.religion.misc	sci.electronics
talk.politics.mideast	sci.med
talk.politics.misc	
talk.politics.guns	

Naive Bayes: 89% classification accuracy

20 Bayesian Belief Networks (also called Bayes Nets)

Interesting because:

- Naive Bayes assumption of conditional independence too restrictive
- But it's intractable without some such assumptions...
- Bayesian Belief networks describe conditional independence among *subsets* of variables

→ allows combining prior knowledge about (in)dependencies among variables with observed training data

20.1 Conditional Independence

Definition: X is *conditionally independent* of Y given Z if the probability distribution governing X is independent of the value of Y given the value of Z.

That is, if

$(\forall x_i, y_j, z_k)\ P(X = x_i | Y = y_j, Z = z_k) = P(X = x_i | Z = z_k)$

or, more compactly, we write

$$P(X|Y, Z) = P(X|Z)$$

Example 4 *Thunder is conditionally independent of Rain, given Lightning*

$$P(Thunder|Rain, Lightning) = P(Thunder|Lightning)$$

Naive Bayes uses conditional independence to justify the following equation which very useful computationally:

$$\begin{aligned} P(X, Y|Z) &= P(X|Y, Z)P(Y|Z) \\ &= P(X|Z)P(Y|Z) \end{aligned}$$

Network represents a set of conditional independence assertions:

- Each node is asserted to be conditionally independent of its nondescendants, given its immediate predecessors.
- Directed acyclic graph

Represents joint probability distribution over all variables

- e.g., $P(Storm, BusTourGroup, \ldots, ForestFire)$
- in general,

$$P(y_1, \ldots, y_n) = \prod_{i=1}^{n} P(y_i | Parents(Y_i))$$

  where $Parents(Y_i)$ denotes immediate predecessors of Y_i in graph
- so, joint distribution is fully defined by graph, plus the $P(y_i | Parents(Y_i))$

20.2 Inference in Bayesian Networks

How can one infer the (probabilities of) values of one or more network variables, given observed values of others?

- Bayes net contains all information needed for this inference
- If only one variable with unknown value, easy to infer it
- In general case, problem is NP hard

In practice, can succeed in many cases

- Exact inference methods work well for some network structures
- Monte Carlo methods "simulate" the network randomly to calculate approximate solutions

20.3 Learning of Bayesian Networks

Several variants of this learning task

- Network structure might be *known* or *unknown*
- Training examples might provide values of *all* network variables, or just *some*

If structure known and observe all variables

- Then it's easy as training a Naive Bayes classifier

Suppose structure known, variables partially observable

e.g., observe *ForestFire, Storm, BusTourGroup, Thunder*, but not *Lightning, Campfire*...

- Similar to training neural network with hidden units (not covered yet)
- In fact, can learn network conditional probability tables using gradient ascent (not covered yet)!
- Converge to network h that (locally) maximizes $P(D|h)$

21 Gradient Ascent for Bayes Nets

Let w_{ijk} denote one entry in the conditional probability table for variable Y_i in the network

$$w_{ijk} = P(Y_i = y_{ij} | Parents(Y_i) = \text{the list } u_{ik} \text{ of values})$$

e.g., if $Y_i = Campfire$, then u_{ik} might be $\langle Storm = T, BusTourGroup = F \rangle$

- **Idea in Gradient Ascent**: Need to maximize a quantity. Use gradient - the vector of its partial derivatives with respect to its variables and adjust these in the direction of the gradient.
- The Gradient Ascent Training for Bayesian networks was given by Russell (most of you would know this from the AI course) (we will follow this in class: book pages 188-189) to obtain the following formula for gradient ascent:
  1. update all w_{ijk} using training data D
  $$w_{ijk} \leftarrow w_{ijk} + \eta \sum_{d \in D} \frac{P_h(y_{ij}, u_{ik}|d)}{w_{ijk}}$$
  2. then, renormalize the w_{ijk} to assure
     - $\sum_j w_{ijk} = 1$
     - $0 \leq w_{ijk} \leq 1$

- Russell shows that this converges to a **locally** optimal solution: locally maximum likelihood hypothesis for the conditional probabilities of the Bayesian network.

21.1 Learning the Structure of the Bayesian Network

Very difficult (some research is being done):

- greedy search among network structures on a cost function that trades off structure complexity for accuracy;
- Constraint-based approaches.