DM812
METAHEURISTICS

Lecture 10

Machine Learning and Estimation of Distribution Algorithms

Marco Chiarandini

Department of Mathematics and Computer Science
University of Southern Denmark, Odense, Denmark
<marco@imada.sdu.dk>

Outline

Outline

Machine Learning

Machine Learning: algorithms to make computers learn from experience and modify their activities.

Sub-field of artificial intelligence and, hence, of computer science

But related also with statistics (statistical learning, learning from data) computational complexity, information theory, biology, cognitive sciences, philosophy

Components of a machine learning model: environment, learning element, performance element

Statistical learning methods: least squares, maximum likelihood, decision trees, Bayesian networks, artificial neural networks, evolutionary learning, reinforcement learning

Supervised learning: predict the value of an outcome measure based on a number of a number of input measures

- Probability space $\mathcal{P}$, input variables $\mathbf{x} \in \mathbf{R}^p$ and output variable $y \in \mathbf{R}$
- Learning machine implementing a function $F(\mathbf{x}, \theta), \theta \in W$
- Find $F(\mathbf{x}, \theta)$ such that, for example, $\min E\left[(F(\mathbf{x}, \theta) - y)^2\right]$

Unsupervised learning: find association patterns among a set of input without any output being supplied

Reinforcement Learning: learn an input-output mapping through continued interaction with the environment rather than learning from a teacher.

Learning Tasks and Domains

- Classification: recognition of classes
- Regression, Interpolation and Density Estimation: functional description
- Learning sequence of actions: robots, chess
- Data mining: learning patterns and regularities from data

Two phases: training and testing (or learning and generalizing)

Issues: which algorithm for which task? how many training samples

Learning Algorithms

Overview

- Decision trees
- Inductive Logic Programming
- Bayesian Learning
- Reinforcement Learning
- Neural Networks
- Evolutionary Learning

Information Theory

Information theory, founded by Shannon, is a branch of probability theory and statistics whose aim is quantifying information

The information entropy H measures the information contained in random variables:

- X discrete random variable
- $\mathbb{X}$, is the set of all messages x that X could be,
- $p(x) = \Pr(X = x)$ probability of X given x
- $I(x)$ self-information, the entropy contribution of an individual message
- bit based on binary logarithm is the unit of information

$$H(X) = \mathbb{E}_X[I(x)] = -\sum_{x \in \mathbb{X}} p(X) \log p(X).$$

The entropy is maximized when all the messages in the message space are equiprobable $p(x) = 1/n$ in which case $H(X) = \log n$

$$H(X) = \mathbb{E}_X[I(x)] = -\sum_{x \in \mathbb{X}} p(X) \log p(X).$$

Example: Random variable with two outcomes

$$H_{\text{b}}(p) = -p \log p - (1-p) \log(1-p)$$

A fair coin flip (2 equally likely outcomes) will have less entropy than a roll of a die (6 equally likely outcomes).

Outline

Model Based Metaheuristics

Key idea Solutions generated using a parameterized probabilistic model updated using previously seen solutions.

1. Candidate solutions are constructed using some parameterized probabilistic model, that is, a parameterized probability distribution over the solution space.

2. The candidate solutions are used to modify the model in a way that is deemed to bias future sampling towards low cost solutions.

Stochastic Gradient Method

- $\{\mathcal{P}(s,\theta) \mid \theta \in \Theta\}$ family of probability functions defined on $s \in \mathcal{S}$
- $\Theta \subset \mathbf{R}^m$ m-dimensional parameter space
- $\mathcal{P}$ continuous and differentiable

The original problem may be replaced with the following continuous one

$$\text{argmin}_{\theta \in \Theta} E_\theta [f(s)]$$

Gradient Method:

Step 1: start from some initial guess θ_0

Step 2: at stage t, calculate the gradient $\nabla E_{\theta_t}[f(s)]$ and update θ_{t+1} to be $\theta_t + \alpha_t \nabla E_{\theta_t}[f(s)]$ where α_t is a step-size parameter.

Computing $\nabla E_{\theta_t}[f(s)]$ requires summation over all the search space.

Stochastic Gradient Method:

Step 2: $\theta_{t+1} = \theta_t + \alpha_t \sum_{s \in S_t} f(s) \nabla \ln \mathcal{P}(s, \theta_t)$

Estimation of Distribution

Estimation of Distribution Algorithms (EDAs)

Key idea: estimate a probability distribution over the search space and then use it to sample new solutions

Attempt to learn the structure of a problem and thus avoid the problem of breaking good building blocks of EAs

EDAs originate from evolutionary computation by replacing recombination and mutation with

- Construct a parameterized probabilistic model from selected promising solutions
- New solutions are generated according to the constructed model

Needed:

- A probabilistic model
- An update rule for the model's parameter and/or structure

Estimation of Distribution Algorithm (EDA):

```
generate an initial population $sp$
while termination criterion is not satisfied: do
    select $sc$ from $sp$
    estimate the probability distribution $p_i(x_i)$ of solution component $i$
      from the highest quality solutions of $sc$
    generate a new $sp$ by sampling according to $p_i(x_i)$
```

Probabilistic Models

No Interaction

- Simplest form of EDA: univariate marginal distribution algorithm [Mühlenbein and Paaß, 1996]. Assumes variables are independent.
- Works asymptotically the same as uniform crossover.
- Probabilities are chosen as weighted frequencies over the population
- Solutions are replaced by their probability vector
  In binary representations, all probabilities are initialized to 0.5
- A mutation operator can be applied to the probability
- Classical selection procedures
- Incremental learning with binary strings:
  $p_{t+1,i}(x_i) = (1-\rho)p_{t,i}(x_i) + \rho r_{t,i}(x_i)$ with $x_i \in S_{best}$

Probabilistic Models (cont.)

Pairwise Interaction

- Chain distribution of neighboring variables
  conditional probabilities constructed using sample frequencies
  solutions replaced by a vector of all pairwise probabilities
- Chains (= ordering of variables) found by a greedy algorithm
- Dependency trees
- Forest

Probabilistic Models (cont.)

Multivariate

- Independent clusters based on minimum description length found by a greedy algorithm that merges groups with least increase in the metric
- Bayesian optimization algorithm: learns a Bayesian network that models good solutions
  metric to select among models (Bayesian-Dirichlet looks only at accuracy)
  greedy algorithm to search the network
- Increased computational time spent by learning the models
- Prior information can be included
- Should work for decomposable problems

Bayesian Optimization

[Pelikan and Goldberg (GECCO-1999)]

The Bayesian Optimization Algorithm (BOA)

Step 1: set $t = 0$ randomly generate initial population $P(0)$

Step 2: select a set of promising strings $S(t)$ from $P(t)$

Step 3: construct the network B using a chosen metric and constraints

Step 4: generate a set of new strings $O(t)$ according to the joint distribution encoded by B

Step 5: create a new population $P(t+1)$ by replacing some strings from $P(t)$ with $O(t)$
set $t = t + 1$

Step 6: if the termination criteria are not met, go to (2)

Bayesian Networks

- BN encodes relationships between the variables contained in the modeled data, representing the structure of the problem.
- Each node corresponds to one variable X_i (ith position in the string)
- Directed edges represent relationships between variables
- Restriction to acyclic graphs with max in-degree k

- $X = (X_1, \ldots, X_n)$ vectors of variables
- Π_{X_i} set of parents of X_i in the network

$$p(X) = \prod_{i=1}^{n} p(X_i|\Pi_{X_i})$$

Bayesian Networks (cont.)

Example of measures of the quality of networks
(related to the concept of model selection in statistics)

- Minimum Description Length: (implements sort of Occam's razor)
  "length of the shortest two-part code for a string s that uses less than α bits. It consists of the number of bits needed to encode the model m_p that defines a distribution and the negative log likelihood of s under this distribution. "

  $$\beta_s(\alpha) = \min_{m_p} \left[-\log p(s|m_p) + K(m_p) : p(s|m_p) > 0, K(m_p) \leq \alpha \right]$$

  $K(s)$ Kolomogorov complexity: shortest program that can output s on the Universal Turing Machine

- Bayesian Dirichlet metric: combines the prior knowledge about the problem and the statistical data from a given data set.

- Many others: eg, Akaike Information Criterion, ...

Bayesian Networks (cont.)

Searching for a Network that maximize the value of a scoring metric

$k = 0$ Trivial. An empty network is the best one (and the only one possible).

$k = 1$ polynomial algorithm by reduction to a special case of the maximal branching problem (Edmonds, 1967).

$k > 1$ the problem is NP-hard (Heckerman et al., 1994).
Simple local search based methods can be used with edge additions, edge reversals, and edge removals (preserving acyclicity).
Simple greedy algorithm with edge additions starting from an empty network.

Bayesian Networks (cont.)

Generating new solutions
New solutions are generated using the joint distribution encoded by the network.

Step 1: compute the conditional probabilities of each possible instance of each variable given all possible instances of its parents in a given data set

Step 2: use the conditional probabilities to generate each new instance. At each iteration, the nodes whose parents are already fixed are generated using the corresponding conditional probabilities.
Repeat until the values of all variables are generated.

Since the network is acyclic, the algorithm is well defined.

The time complexity of generating an instance of all variables is bounded by $O(kn)$ where n is the number of variables.