Intensity-based protein identification by machine learning from a library of tandem mass spectra
Seminar paper
Yan Zhao

Helsinki November 1, 2010
UNIVERSITY OF HELSINKI
Department of Computer Science
This paper is to introduce decision tree model, a machine learning approach, in protein identification. By constructing a library of tandem mass spectra with high-quality matches, intensities are modeled in order to select attributes and corresponding values for a decision tree. Train the model with non-redundant data to build a match tree. Test the performance of the decision tree and prove that this method is effective both in precision and sensitivity in peptide identification. Further prove that it is also a potential approach in protein identification.
# Contents

1 Introduction ................................................................. 1

2 Basics in machine learning ........................................... 1

   2.1 Machine learning .......................................................... 2
   2.2 How to learn ................................................................. 2
   2.3 Classification ............................................................... 2
   2.4 Representation of a tree ............................................... 2
   2.5 Entropy ........................................................................ 3

3 Peptide identification .................................................. 4

   3.1 Tandem mass spectrometry (MS/MS) ................................ 4
   3.2 Intensity modeling ......................................................... 5
   3.3 Construct decision tree ................................................. 7

4 Conclusion ....................................................................... 8

References ........................................................................ 9
1 Introduction

The rapid development in biology has raised new acquirements for bioinformatics. Facing with the large amount of biological data, people are thinking that there should be more effective approaches to store and to extract the useful information. Year after year, scientists are making great efforts to design new tools. Machine learning techniques develop very fast under this background. The field of machine learning is concerned with the question of how to construct computer programs that automatically improve with experience [1]. In several biological domains, machine learning plays an important role, especially genomics and proteomics. In this seminar paper, I will focus on a machine learning approach – decision tree and its application in protein identification.

An introductory section will be given mainly about the basic ideas in machine learning. Following parts will be my study report of the article “Intensity-based protein identification by machine learning from a library of tandem mass spectra” by Joshua E Elias and other four authors.

2 Basics in machine learning

As everyone knows, the ability of learning is the most important factor which makes human beings more advanced than other kinds of mammal. It is learning that helps people gain experience, create new tools to conquer natural challenges and evolute.

Learning provides us with intelligence while our intelligence feeds back to learning. Ever since the invention of computer, we have been benefiting from computer science almost in every aspect of our daily life. From my stand of point, it is not because computers are more intelligent than humans, but they are fast and accurate, unimaginable speed and accuracy. Computers are merely doing what an algorithm tells them to. Here comes the question, since humans could program computers to execute commands, whether we could program them to learn – to improve automatically with what they have done. There has been many fields where machine learning plays an important role. Computers learn from medical records, giving therapeutic suggestions on which treatments are more effective for a new disease. Vero system gains experience to direct citizens to declare personal income in a simple manner saving our time on reading the items. HSL selects routes with the best road condition and least time.
2.1 How to learn

The learning problem is actually a progress to improve with experience at some task. We ask computers to collect information in form of data and transform data into knowledge which is really what human could understand. System is supposed to give better choice to users through its experience in route selection. To make computer learn is to design advanced algorithms which could direct effectively.

2.2 Classification

Classification is a supervised machine learning procedure in which individual items are placed into groups based on quantitative information on one or more characteristics inherent in the items and based on a training set of previously labeled items. There are several algorithms could give classifiers to achieve data training, for example Linear classifiers, quadratic classifiers, decision tree, neural networks and so on.

In this seminar paper, I am focusing on decision tree, a tree-like graph which is used as a kind of classifier to identify proteins.

2.3 Decision tree

Decision tree learning is a method for approximating discrete–value target functions, in which the learned function is represented by a decision tree. Learned trees can also be re-represented as sets of if-then rules to improve human readability [1]. It is a process using a model which transform data information into structural graphs according to certain rules. This tree-like model will lead the data flow from one root node to leaf nodes which stand for different decisions about the task.

2.4 Representation of a tree

In order to explain how to construct a decision tree, let us use an example in which we are considering the task – decision for PlayTennis. On Saturday morning, one is thinking about going to play tennis. How would the weather factors influence the decision? We transform this task into this tree (see Figure 2.1 [1]) and ask it for help to make decision.
Figure 2.1
A decision tree for the concept PlayTennis. An example is classified by sorting it through the tree to appropriate leaf node, then returning the classification associated with this leaf (in this case, Yes or No). This tree classifies Saturday morning according to whether or not they are suitable for playing tennis.

➢ Each internal node tests an attribute.
➢ Each branch corresponds to attribute value.
➢ Each leaf node assigns a classification.

The constructing of a decision tree is to classify instances by sorting them down the tree from root to leaf. Each node in the tree specifies a test of some attribute of the instance, and each branch descending from that node corresponds to one of the possible values for the attribute. An instance is classified by starting at the root node of the tree, testing the attribute specified by this node, then move down the tree branch corresponding to the value of the attribute. Repeating this process will build subtrees.

2.5 Entropy

There is a common concept in information theory, called entropy. Entropy is used to characterize the purity of an arbitrary collection of examples in order to define information gain precisely. Given a collection S, containing positive and negative examples of some target concept, the entropy of S relative to this boolean classification is

$$\text{Entropy}(S) \equiv \sum_{i=1}^{c} - p_i \log_2 p_i$$

where $p_\oplus$ is the proportion of positive examples in S and $p_\ominus$ is the proportion of negative examples in S. In all calculation involving entropy, $0 \log 0$ is defined 0.

If a target attribute can take on c different values, then the entropy of S is

$$\text{Entropy}(S) \equiv \sum_{i=1}^{c} - p_i \log_2 p_i$$

where $p_i$ is the proportion of S belonging to class i. The logarithm is base 2 because entropy is a measure of the expected encoding length measured in bits.
Information gain is defined to measure the effectiveness of an attribute in classifying the training data when entropy is given. This is to measure the expected reduction in entropy caused by partitioning the examples according to this attribute. We use $Gain(S,A)$ to denote the information gain of an attribute $A$, related to a collection of examples $S$, which is calculated as

$$Gain(S,A) \equiv \text{Entropy}(S) - \sum_{(v \in \text{values}(A))} \left( \frac{|S_v|}{|S|} \right) \times \text{Entropy}(S_v)$$

where $\text{values}(A)$ is the set of all possible values for attribute $A$, and $S_v$ is the subset of $S$ for which attribute $A$ has the value $v$. With $Gain(S,A)$, we can evaluate the relevance of attributes and determine which is a better attribute for classifying the training examples.

3 Peptide identification

In this chapter, I will present the application of decision tree in the article “Intensity-based protein identification by machine learning from a library of tandem mass spectra” [3] which was published online on 18th January 2004 on nature biotechnology.

The authors of this article implement a probabilistic decision tree to model the probability of observing a fragment ion intensity, conditioned on 63 peptide and fragment attributes, using peptide-spectrum matches (PSMs) as training data. They use reduction in Shannon entropy of intensity to select attributes and corresponding values for decision points. And finally construct a match tree for identifying peptides.

3.1 Tandem mass spectrometry (MS/MS) [4]

Mass spectrometers are commonly combined with separation devices. MS/MS is an analogous technique which combines two mass spectrometers(see Figure3.1).

![Figure 3.1](image-url)

The first Ms is used to select a single mass. The second MS is used to separate the fragment ions.
The “MS/MS” spectrum will be resulted after the analysis, which consists only of product ions from the selected precursor. Chemical background and other mixture components are absent. Peptide-spectrum matches (PSMs) are the results gained by analysing the mass spectrum with peptide database. If we run a MS/MS process for certain kinds of peptide, a mass spectrum will be generated. By discovering the fragment-peptide pairs, we could search the database to find out whether they exist. [5]

### 3.2 Intensity modeling

Mass spectrum is a kind of biological data which contains peptide fragment information mapping certain sequence of amino acids and certain type of proteins. In this research work, authors try to employ intensity information to reflect the probabilities of observing a predicted single fragment ion. There is an example which illustrates what a intensity-based spectrum looks like by generating a lod, spectrum of all predicted fragments from peptide SALSGHLETILGLK. A log-odds ratio (lod) approach is used to represent $i^{th}$ fragment referring to the ratio of likelihood ($p$) of the observed fragment intensity under the alternative hypotheses of a match or mismatch candidate peptide. ( see function below ).

\[
\text{lod}_i = \log_{10}\left( \frac{p(\text{observed intensity} | \text{attributes, match})}{p(\text{observed intensity} | \text{attributes, mismatch})} \right)
\]

In order to obtain the intensity probabilities, researchers need to model the intensities with both matched and mismatched peptides. Approximately 1,000,000 tandem mass spectra (TMS) were acquired and searched against appropriate protein sequence databases with the SEQUEST algorithm without enzymatic restriction.

To model the intensity information, about 140,000 fully tryptic spectral matches receiving XCorr scores $\geq 2.0$ and ΔCn scores $\geq 0.10$ were selected as high confidence PSMs. Of these, 27,266 were randomly selected as unique. Each unique peptide in this set was fragmented in silico to generate 783,994 b- and y-type ions to which observed intensities were mapped from acquired spectra. The intensity is modeled by $H(I)$,

\[
H(I) = - \sum p(I_i) \ln p(I_i)
\]
where $H$ is entropy, $j$ is a given intensity bin index and $I_j$ is the intensity bin $j$. The change in entropy is calculated as,

$$\Delta H = \frac{N}{M} (H_{\text{parent}} - p(\text{right child})H_{\text{right child}} - p(\text{left child})H_{\text{left child}})$$

$N$ is the number of data points at the parent node and $M$ is the number of data points at the root node. The attribute and bin yielding the greatest reduction in entropy are selected as the next node for growing the decision tree. Branches are terminated when the creation of additional child nodes do not reduce entropy to improve the model. With these predicted fragment intensities, the probability of observing any single fragment intensity could be calculated conditioning on matched or mismatched spectra. As instructed by the function, two probabilities are combined to get a log-odds ratio which is a part of the complete lod, spectra. The procedure could be illustrated by Figure 3.2.

![Figure 3.2](image-url)

Steps towards generating a particular lod score. "U" denotes unobserved fragment intensities.

The following figure is the lod, spectra for the example peptide. (Figure 3.3)
Each predicted fragment m/z is assigned a lod score, plotted on the lower axis below the MS/MS spectrum. Positive values indicate that the fragment is more likely to be derived from a correctly matched PSM than a mismatched PSM. Negative values indicate that the fragment is more likely to be derived from a mismatched PSM than a matched. XCorr stands for SEQUEST cross-correlation score, LOD is for Log(p(intensity|match)/p(intensity|mismatch)) and ΔCn means normalized difference between first- and second-ranked XCorr scores. The letters b and y stands for b-type ion and y-type ion separately.

### 3.3 Construct decision tree

Once obtained the intensity-based spectra, the attributes and corresponding values acting as decision points which best segregate the resulting subgroup of fragment ions are calculated. The following table lists part of the attributes in the decision tree model.

**Table 3.1** The abbreviations and descriptions for attributes are list in the following table.

<table>
<thead>
<tr>
<th>Tree abbreviation</th>
<th>Attribute description</th>
<th>Tree</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISTC</td>
<td>Distance (number of residues) from fragmentation site to C terminus</td>
<td>Match, mismatch</td>
</tr>
<tr>
<td>DISTN</td>
<td>Distance (number of residues) from fragmentation site to N terminus</td>
<td>Match, mismatch</td>
</tr>
<tr>
<td>FRACF_HKR</td>
<td>Fraction of histidines, lysines, arginines in fragment ion</td>
<td>Match</td>
</tr>
<tr>
<td>GBC_X</td>
<td>Gas phase basicity of residue X positions C-terminal to fragmentation site</td>
<td>Mismatch</td>
</tr>
<tr>
<td>GBF</td>
<td>Average gas phase basicity of fragment ion</td>
<td>Mismatch</td>
</tr>
<tr>
<td>GBN_X</td>
<td>Gas phase basicity of residue X positions N-terminal to fragmentation site</td>
<td>Match</td>
</tr>
<tr>
<td>HLCN_X</td>
<td>Helicity of residue X positions N-terminal to fragmentation site</td>
<td>Match</td>
</tr>
<tr>
<td>HYDN_X</td>
<td>Hydrophobicity of residue X positions N-terminal to fragmentation site</td>
<td>Match</td>
</tr>
<tr>
<td>DON</td>
<td>b- or y-type ion</td>
<td>Match</td>
</tr>
<tr>
<td>LENF</td>
<td>Length of fragment ion</td>
<td>Mismatch</td>
</tr>
<tr>
<td>LENP</td>
<td>Length of peptide</td>
<td>Match</td>
</tr>
<tr>
<td>M_Z</td>
<td>Fragment m/z</td>
<td>Match, mismatch</td>
</tr>
<tr>
<td>NTERM</td>
<td>Identity of residue at peptide’s N terminus</td>
<td>Match</td>
</tr>
<tr>
<td>NUM_HKR</td>
<td>Number of histidines, lysines, arginines in peptide</td>
<td>Match</td>
</tr>
<tr>
<td>PMASC</td>
<td>Fragment mass – precursor mass</td>
<td>Match, mismatch</td>
</tr>
<tr>
<td>PMZ</td>
<td>Precursor ion m/z</td>
<td>Match</td>
</tr>
<tr>
<td>PMZD</td>
<td>Fragment m/z – precursor m/z</td>
<td>Match, mismatch</td>
</tr>
<tr>
<td>POS</td>
<td>Fractional location of fragmentation site along peptide</td>
<td>Match, mismatch</td>
</tr>
<tr>
<td>RESC_X</td>
<td>Identity of residue X positions C-terminal to fragmentation site</td>
<td>Match</td>
</tr>
<tr>
<td>RESN_X</td>
<td>Identity of residue X positions N-terminal to fragmentation site</td>
<td>Match</td>
</tr>
</tbody>
</table>

All previous accomplishments will result the decision tree for peptide identification, trained by 27,000 match spectra, the decision tree branches were terminated according to the Bayesian information criterion to avoid over-fitting. Finally the match tree is graphically represented in Figure 3.4.

---

1. **SEQUEST** is a TMS data analysis program used for protein identification. SEQUEST identifies collections of tandem mass spectra to peptide sequences that have been generated from databases of protein sequences.

2. In mass spectrometry, an N-terminal fragment of a polypeptide produced by cleavage of the bond between an α-carbon and its neighbouring α-carboxy carbon. A **b-type ion** is an α-N-terminal acylium ion which results from cleavage of an amide bond, and **y-type ions** are C-terminal iminium ions from such a cleavage.
8

Figure 3.4
Probabilistic match decision tree automatically learned from fragment intensity data. Text within internal nodes indicates the attribute that provides the greatest reduction in intensity distribution entropy for fragments reaching that point in the tree. Internal nodes or leaf nodes each have a corresponding probability distribution of intensities. The six colors shown in the key correspond to the six intensity bins. Saturated internal node colors indicate probability distribution that strongly favor a single intensity bin (lower entropy). Lower saturation indicates a weaker tendency to favor a particular intensity bin (higher entropy). The relationship between color, saturation and intensity is depicted by the thumbnail graphs of probability (p) versus observed intensity (I) above (low entropy) and below (high entropy). Arrowhead area is proportional to the fraction of fragments emanating from the source node. Arrows pointing to the right indicate fragments satisfying the condition indicated by the source node; arrows pointing left indicate fragments that do not. The tree is used to assign each individual input fragment to representative leaf node. Numbers within the leaf nodes indicate the number of training set fragments used to generate the intensity probability distribution represented by the node. This tree was generated from fragment ions of the top SEQUEST-ranked high-confidence peptide-spectral match.

4 Conclusion

The decision tree not only rediscovered qualitative phenomena known to experienced mass spectrometrists, also identified undescribed rules, for example, the proline's inhibitory effect on fragmentation may extend to the second C-terminal peptide bond (RESC_2=P), which lends credence to machine learning approach. In comparison with published SEQUEST criteria and other analytical tools, the intensity-based method reduces peptide identification error by 50-96% without any loss in sensitivity. To focus on protein identification, researchers prove that higher precision on the peptide level translates to higher protein precision, which means their decision tree model also performing well in protein identification. However, the correlation is not that perfect: the protein identification error rate is two to three times greater than peptide identification error rate because of the fewer protein mapped from correctly matched peptides.
References


