### Summary

- We are committed to an application-driven development of theory. This means that—
  - we apply theory to problems that arise in applications;
  - we advance the theory inspired by practical solutions.
- Here are some current research projects:
  - Information theory based gene design for heterogeneous gene expression
  - Credit rating using multiclass machine learning algorithms
  - Application of Conditional Random Fields (CRFs) to solve labeling problems
  - CRF-based structure prediction of proteins
  - Circuit complexity of boolean functions
  - Data compression and phylogeny reconstruction
  - Assuring Software Quality by Means of Simulation Methods (with SWE group)
- Software-Development for E-Learning

The courses offered in the master program prepare the participants for those research projects. The topics covered are—
- algorithmic learning and probabilistic data structures;
- algorithmic methods and theoretical concepts in applied computer science;
- computational complexity;
- cryptography and coding theory;
- information theory;
- randomized algorithms.

### 1. Lectures and Seminars — Fall Semester

- **Group seminar**
  - Mo 16-18 (slightly irregular base, controlled by StudIP mailing list)
  - presentation of bachelor/master/PhD projects
  - discussion of current ideas, paper seminar
- **Master level**
  - St. Waack: Seminar on Algorithmic Methods and Theoretical Concepts. Mo 14-16
  - St. Waack: Lecture/Exercise on Algorithmic Learning and Probabilistic Data Models. We/Th 14-16
  - C. Damm: Lecture/Exercise on Randomized Algorithms. We/Th 10-12
- **Bachelor level**
  - St. Waack: Vorlesung/Übung Informatik III
  - C. Damm: Vorlesung/Übung Informatik I
  - C. Damm/B. Ashirmatov: Practical Course Algorithms for Programming Contests

Take care and watch out for possible changes.

### 2. Lectures and Seminars — Spring Semester

- **Group seminar**
  - Mo 16-18 (slightly irregular base, controlled by StudIP mailing list)
  - presentation of bachelor/master/PhD projects
  - discussion of current ideas, paper seminar
- **Master level**
  - St. Waack: Lecture/Exercise on Computational Complexity. We/Th 14-16
  - St. Waack: Lecture/Exercise on Information Theory for Computer Scientists. Tu/Fr 10-12.
  - St. Waack/M. Welter: Conditional Random Fields (CRFs) – a data structure in Bioinformatics and Software Engineering. Th 8-10/14-16
  - C. Damm/L. Wiese: Lecture/Exercise on Cryptography. Tue 12-14/Th 10-12
- **Bachelor level**
  - C. Damm: Vorlesung/Übung Grundlagen der theoretischen Informatik
  - C. Damm: Vorlesung/Übung Theoretische Informatik
  - C. Damm/B. Ashirmatov: Practical course Advanced Algorithms for programming Contests
- **PhD level**
  - C. Damm: Discussion rounds on good scientific practice (guests welcome)

Take care and watch out for possible changes.

### 3. Labeling Problems

We consider random data elements \((X, Y)\), where \(X\) and \(Y\) are sequences of discrete random elements of length \(l\) and \(m\) in which turn take values in the observation alphabet \(\mathcal{A} = \{1, 2, \ldots, a\}\) and the label set \(\mathcal{B} = \{1, 2, \ldots, b\}\). An instance \(x\) of \(X\) is called observation or input, whereas an instances \(y\) of \(Y\) is named output or labeling. An element of the index set \(V = \{1, 2, \ldots, m\}\) of the random vector \(Y\) is a position or site. In the context of image processing sites are called pixel.

Here we only model the conditional probability mass function \((cpmf) p(y | x)\) of \(Y\) given \(X\) defined by \(p(y | x) = P(Y = y | X = x)\).

For an instance \((x, y)\) of \((X, Y)\), the output \(y\) is called labeling of the sites of \(V\) given observation \(x\).

The labeling problem is informally specified as follows. Devise an algorithm that outputs an “appropriate” labeling \(y \in \mathcal{B}^V\) of \(V\) taking \(x \in \mathcal{A}^V\) as input. On possibility to do so is to implement the MAP induction principle defined as follows

\[
y^* = \arg\max_{y \in \mathcal{B}^V} p(y | x).
\]

### 4. Conditional Random Fields (CRFs)

We additional assume that there is a neighborhood relation on the set of sites \(V\) in terms of a site graph \(G = (V, E)\) having maximal clique size \(s\). Let \((X, Y)\) be a random data as before.

If for every \(c \in \{1, 2, \ldots, s\}\) there are real-valued features \(\psi^c(y, x)\) \((I, J)\) of a clique of size \(|I| = c\) such that the conditional probabilities \(p(y | x)\) satisfy the following equations.

\[
p(y | x) = \frac{1}{Z(x)} \exp \left( \sum_{c=1}^{s} \sum_{|I|=c} \psi^c(y, x) \right),
\]

where \(Z(x)\) is the canonical normalization factor, then we call the pair \((X, Y)\) a conditional random fields. Alternatively, we speak of a feature-based exponential model.

We study conditional random fields to solve labeling problems. All that matters here is to determine the features and to make the MAP prediction tractable. The latter in turn heavily depends on the site graph.

### 5. CRF-based Research Projects

**Usability of CRFs.** Depending on the underlying site graph and the features used, it is often the case that some of the algorithmic support available for Hidden Markov Models is not there in the case of conditional random fields. Moreover, training for CRFs is still a challenge in the general case. Our way to improve usability of CRFs is to make use of Markov-Chain Monte-Carlo Methods.

**Prediction of protein structures.** Here take CRFs as a probabilistic model for detecting rigid-body movements in protein structures. The vertices of the side graph are the amino acids of the protein, where two amino acids are joined by an edge if and only if they are neighboring ones in the 3D structure of the protein.

**Software Quality Assurance.** CRFs are used to assess the quality of a software project. The vertices of the site graph are software entities, where to entities are connected by an edge if they depend on each other, since an excellent software entity does not perform well if it requires services of error-prone other software entities.

### 6. Data Compression Applications

- To estimate similarity, compress one object using the other.
  - Toy example: Describe zebra and elephant using horse as side-information.
    - A zebra is a striped horse.
    - An elephant is a huge horse with large ears, long nose, …
  - zebra and horse are close to each other, elephant and horse are far from each other.
- Universal Application: Compression based phylogenies
  1. estimate object distances using data compression with side-information
  2. group objects hierarchically, ranked by closeness (neighbor joining)
  3. translate inclusion hierarchy into a phylogenetic tree
- Some research problems:
  - explain the observed correlation of compression distance to other distance measures
  - design algorithmic improvements based on observed correlation