Introduction

Introduce the topics that will be discussed in this thesis.

- In this chapter we will introduce foundations such as Kolmogorov-complexity, MDL, entropy, etc.
- Furthermore we will give a brief description of the problem, with some examples
- The introduction is concluded with a section on related work
- In the chapter 'Theoretical Framework' we discuss the formal part of VOUW, with all of its definitions and a more complete description of the problem
- In the chapter 'A Search Algorithm' we will talk about the actual search algorithm and extensions thereof
- The (short) chapter 'Practical Implementation' will be about actual code and QVouw, the GUI
- 'Experiments' will try to empirically show some of the concepts introduced in the previous two chapters
- And finally 'Discussion' and 'Conclusion' will conclude the thesis

General Problem Description

VOUW is a framework/set of definitions, a practical algorithm an academic implementation of these concepts. We use explanatory data mining to discover the structure of a dataset. To goals is not only to describe the data as concise as possible, but also how we derive the vocabulary to make that description. This concept can be extended from analysis to similarity and clustering: want we not only be able to tell how similar two datasets are, but also why they are (dis)similar.

Brief Example

I would like to illustrate the problem with a small example, such that the reader's interest is (hopefully) piqued. This example could be a small fabricated matrix or (ideally) a toy example that bears some real-world relevance.

Grids, Images and Matrices

Introduction

The problem is defined on a specific type of input:

- Matrix-like data, but not in the linear algebra sense of the word. The closest description would be an image, because it also is a rigid, grid-like dataset where rows and columns have a fixed ordering.
- VOUW is, however, not an image mining algorithm and therefore we use the term 'matrix'.
- We use 2D input only. The algorithm and definitions could be expanded for multi-dimensional matrices.
- Individual data points must be discrete.
- Matrix can be sparse, but low-density matrices yield no useful patterns and thus the algorithm has limited usability on sparse matrices.

Patterns

In order to gain better understanding of the input data, we try to express it in terms of a set of *patterns*. In this context a pattern is:

- A submatrix of the original matrix
- A structure that occurs more than once, either exact or with some degree of 'noise tolerance'

A complete, loss-less description does not only contain patterns, but also tells us where they are located in the original matrix. We call the placement of a pattern on the original matrix an *instance* of a pattern. A description of the original matrix is valid only if each element belongs to exactly one instance. There are, however, many valid descriptions possible. The third chapter discusses a search algorithm that tries to approach the optimal.

The Problem and Its Classes

Apart from a formal definition, it is useful to semantically define what kinds of problems we would like to be able to solve. Therefore a division in five different classes was made. [1]

[1]

Class 1a Exact duplicates in a sparse matrix.

Class 1b Exact duplicates in (differently distributed) noise.

Class 1c Exact duplicates in noise with equal distribution.

Class 2 Approximate duplicates

Class 3 Transformed duplicates