On musical score recognition using probabilistic reasoning

VUILLEUMIERSTUECKELBERG, Marc Christian, DOERMANN, David

Abstract

We present a probabilistic framework for document analysis and recognition and illustrate it on the problem of musical score recognition. Our system uses an explicit descriptive model of the document class to find the most likely interpretation of a scanned document image. In contrast to the traditional pipeline architecture, we carry out all stages of the analysis with a single inference engine, allowing for an end-to-end propagation of the uncertainty. The global modeling structure is similar to a stochastic attribute grammar, and local parameters are estimated using hidden Markov models.

Reference


Available at:
http://archive-ouverte.unige.ch/unige:121385

Disclaimer: layout of this document may differ from the published version.
On Musical Score Recognition using Probabilistic Reasoning

Marc Vuilleumier Stückelberg  
CUI - University of Geneva  
CH-1211 Geneva 4, Switzerland  
mvuilleu@cui.unige.ch

David Doermann  
LAMP - University of Maryland  
College Park, MD 20742  
doermann@cfar.umd.edu

Abstract

We present a probabilistic framework for document analysis and recognition and illustrate it on the problem of musical score recognition. Our system uses an explicit descriptive model of the document class to find the most likely interpretation of a scanned document image. In contrast to the traditional pipeline architecture, we carry out all stages of the analysis with a single inference engine, allowing for an end-to-end propagation of the uncertainty. The global modeling structure is similar to a stochastic attribute grammar, and local parameters are estimated using hidden Markov models.

1. Introduction

Musical score recognition (MSR) is the task of interpreting the content of the bitmap image of a musical score and reformulating it with a high-level symbolic structure. Most existing MSR systems combine pattern recognition algorithms and a symbolic parser in well separated, sequential stages, without any feedback [1]. Items found in the bitmap image are classified using a pattern recognition module, and then rule-based inferences [2]-[3] or graph-based inferences [4] are used to recover a structured interpretation. Before, between and after these two main modules, ad-hoc modules are used to preprocess the image, build a list of connected components, locate staves and symbols, and organize the recognition process.

The first successful attempt to integrate pattern recognition with top-down reasoning was limited to the subtask of note recognition [5]. Some years later, the document image decoding (DID) approach [6] was successfully applied to a simplified model of the music notation[7], eventually integrating all stages of MSR into a single optimization process. Patterned after the use of hidden Markov models in speech recognition, DID produces an estimate of the original message coded in the document according to a maximum a posteriori decision criterion. Given a Markov model of a music image source and a music symbol set, DID uses a generalized Viterbi decoder to recover the most likely sequence of symbols that might have generated the bitmap image. Among many interesting aspects, it is segmentation-free, it allows for the integration of a noise model within the optimization process, it maximizes a well-defined objective function, and it allows for a complete separation of the document model from the recognition algorithms.

Despite its unique qualities, DID has not emerged as a preeminent technique for graphics recognition, possibly because the size of the underlying Markov model grows prohibitively fast when handling exceptional cases. Moreover, although heuristics have been found to reduce the computational cost of the optimal decoding procedure to a very acceptable level for line-based text decoding, it is unlikely that such powerful heuristics exist for complex graphics recognition problems such as MSR.

This paper presents another approach to probabilistic document decoding that addresses some of the limitations of DID, while compromising few of the goals of DID. We look for the maximum a posteriori of an objective function representing the likelihood of the document interpretation, and use a task-specific model of the document to decode. The model is completely separated from the recognition engine. As in DID, our approach is principally grounded on stochastic attribute grammars and hidden Markov models, and is segmentation-free.

To facilitate both the knowledge modeling and the recognition task, we moved from a generative (source) model to a descriptive recognition model. This is a major departure from previous work, although our recognition model is still loosely equivalent to a source model when considered as a set of constraints. The objects in our model are defined using arbitrary parameters rather than using the strict sidebearing model. Moreover, we do not limit observations to predefined templates, and we separate the measures (that can be performed on the document) from the prior model (that predicts the value of such measures). In contrast to the non-overlapping symbol constraint of DID, our model allows for the decoding of an image by successive refinements and re-analysis of the same pattern
at different stages of the process, as long as the different underlying models are approximately independent. Finally, we do not directly compute the global optimal solution at once, but we search for it in the subspace defined by the combinations of all locally optimal solutions.

The rest of this paper is organized as follows. Section 2 states our formulation of the document recognition task, along with our assumptions and simplifications. Sections 3 and 4 describe the structure of our document model and the recognition engine respectively, with examples taken from the MSR application. Section 5 provides a concrete overview of the resulting recognition process experienced with the prototype implementation of our system. More details, results and discussions can be found in [8].

2. Problem formulation

We view the task of document understanding as the search for the optimal description of a bitmap image, where a description is defined as an instantiation of an expert-defined model describing a class of documents, and is said to be optimal if it has the maximal a posteriori likelihood given the bitmap image. The posterior likelihood of a description is defined as the product of the posterior likelihood over all objects that are part of the description, and the posterior likelihood of an object is defined as the product of the posterior likelihood of all its attributes given the measures and models that link the attributes to the bitmap image.

Formally, we define a descriptive recognition model similar to a stochastic context-free attribute grammar \( G = (N.T.R,S) \), where \( N \) and \( T \) are the non-terminal and terminal symbols representing compound and atomic document objects, \( R \) is a set of rules (object definitions) and \( S \in N \) is the start symbol (the top-level object, for instance a Score Page for MSR). In a stochastic grammar, each rule of \( R \) has an associated probability \( P(Lhs => Rhs) \) so that the sum of the probabilities of all rules for a given left-hand side is one. In the same way, we allow variants in the definition of the objects to recognize, with preset prior probabilities defined so that the probabilities of all configurations sum to one. In addition, attribute grammars associate a collection of attributes with each rule, using functions that compute the values of the attributes on one side given the values of the attributes on the other side. We view the attributes on the left-hand side as the parameters that specify the instantiation of the object, and the attributes on the right-hand side as the corresponding parameters of the object components.

Obviously, an attribute grammar rule has the power of representing an arbitrarily large number of attribute-less rules, as the attribute can take on different values to handle different cases. If the attribute values are not uniquely defined by their functional definition when a rule is to be matched, all cases have to be tested. We handle bundles of similar cases simultaneously by working on parameter distributions rather than on distinct values. For this purpose, as an extension to traditional attribute grammars, we introduce priors on the distributions of attribute values. This is justified by the fact that when defining a parametric model of an object to recognize, one always has a good idea of the kind of distribution that each parameter will follow (typically Gamma distribution for lengths and normal distributions for positions). By using a parametric model of such a distribution, we can very efficiently process a continuum of cases for a computational cost comparable to the handling of a single attribute-less rule. For instance, if a rule involves an affine transform on the attribute, we perform the transform analytically on the parameters of the distribution.

In a traditional stochastic grammar, terminal symbols are directly observable and non-terminal symbols themselves are not observable at all. The presence or absence of a terminal symbol in the observable data triggers or blocks the application of any rule which includes the symbol in its right-hand side. We relax this rule by suppressing directly observable symbols (which have no convincing equivalent in the context of a segmentation-free recognition process) and introducing measures and models for all symbols of the grammar. We associate with each object a collection of measurements that can be performed on the bitmap, and a collection of models that predict the outcome of these measures as a function of the object parameters. If we use the appropriate kind of model (for instance hidden Markov models), we can actually rate the likelihood of any object and its parameters given measures done on the image. See Section 5 for examples.

Parametric modeling of the attribute distributions and model-based likelihood rating can be very naturally combined in the form of a Bayesian inference. Given a prior attribute distribution \( p(a) \) and a likelihood measure \( P(M \mid a, \mu) \) where \( M \) is a model and \( \mu \) is a measure done on the image, we can directly compute the posterior distribution

\[
p(a \mid M, \mu) = P(M \mid a, \mu) p(a)
\]

or estimate its parameters from a Monte Carlo sampling. Apart from the explicit prior assumption, the resulting distribution is equivalent to the set of rules that would have been obtained if the attribute rule would have been expanded and the likelihood of each evaluated separately.

To summarize, our approach is to find the document description \( D \) that maximizes the global likelihood, or equivalently the log likelihood:

\[
P(D) = \sum_{M \in D} \sum_{a, \mu} \log(P(M \mid a, \mu) p(a))
\]
This formula is misleadingly simple, as it does not reflect the fact that most of the priors \( p(a) \) are attributes inherited from another rule (object), and are therefore themselves posterior estimates. However, this does not affect the validity of the above formula as long as the likelihood estimators can reasonably be assumed to be independent, that is, as long as the document model does not contain significantly correlated measure models.

More details on the formulation of the document model are given in the next section.

3. Document modeling

The document model is a task-oriented model that defines objects and how they are to be recognized. For instance in the context of MSR, it defines all entities that are necessary to describe the content of a musical score, from the score page as a whole to the staves to each individual symbol on each staff. The model is parametric, as defined in the previous section, and descriptive rather than generative. Therefore, although the model could theoretically be used to generate musical score images, this would involve solving a hard, possibly underconstrained satisfaction problem, and may produce a meaningless score. The advantage of such a model is that it eases the recognition task, allows for the inclusion of high-level recognition heuristics (for expert-directed search) and also eases the knowledge engineering process as exceptional cases can be handled by over-general rules that will not hurt the recognition process.

A document model is defined as a set of object definitions, and each object is made of a collection of attributes\(^1\) that have to be given a value. An attribute is defined either as a parameter of the object, a region, a measure, a model, an inference, a component object or a relation to another object. Attributes might be expressed as a function of other attributes of the same object, thus defining a directed acyclic dependency graph. The attributes with no prerequisites are the intrinsic parameters defining the object, and receive their prior value distribution from their instantiation context. All other attributes are inferred from the previous one in the order specified by the dependency graph.

For instance, a staff system might be defined by the following collection of attributes:
- the intrinsic parameters defining its bounding box;
- the function defining the corresponding image region;
- a horizontal projection profile measure on the region;
- a hidden Markov model of the projection profile;
- an inferred distribution of cut points to separate the staves, according to the hidden Markov model;
- the expected staff components, with prior parameter distributions defined by the previous inference.

These definitions are expressed simply as mathematical formulas and stored in a text file. They make an objective, explicit recognition model that can be discussed and compared independently of the details of the recognition engine that process these definitions. Actual samples taken from the MSR model can be found in [8].

4. Recognition engine

The recognition engine is a program that takes as input a collection of mathematical formulas making a document model (as described in previous section) and finds the description with the maximum a posteriori likelihood for a given bitmap image (or collection of bitmap images).

Our recognition engine searches globally for the most likely description in the subspace defined by combinations of locally most likely object descriptions. The inner search for the optimal object description is carried by working on parametric attribute value distributions, always focusing on the most likely value while keeping a precise estimate of the uncertainty. We perform the Bayesian inference described in Section 2 by Monte Carlo sampling, and estimate the resulting posterior by a mixture distribution. The number of modes in the distribution is automatically determined by the EM-based estimation procedure, patterned after AutoClass II [9]. Each mode is then further processed separately, the most likely first, in a greedy fashion.

The outer search for the most likely combination of object descriptions is handled by an active chart parser (also known as Earley parser), a commonly used tool for working with context-free grammars. Functional constraints on the attributes are verified on the most likely parse trees only, to take advantage of the polynomial-space parse-forest representation as long as possible [10].

5. Examples

Although we do not yet have a complete model for MSR at the time this paper is written, we describe in this Section some fragments as examples to illustrate our approach.

The root object of our model (equivalent to the start symbol of a grammar) is the scanned page, representing the bitmap image to parse. The first suite of refinements involves vertical and horizontal cuts to locate the score page, the staff systems, the staff system measures, the staves, the measures and the symbol groups. All these cuts are very similar, based on the measure of one-dimensional moments of the image. We exemplify the very first one,

---

\(^1\)Unless specified otherwise, the word attribute is used in this section in its object-oriented acceptation, with no direct correspondence with the attribute of a grammar rule as described in the previous section.
namely finding the boundaries of a score page within a scanned page. A score page has four intrinsic parameters, its four margins. Their prior distributions are Gamma distributions with mean and standard deviation set to 1/10th of the width or height of the page. The corresponding region on the image is defined as:
\[
\text{reg}(x,y) := P(lm < x < width-rm) \cdot P(tm < y < height-bm)
\]

The posterior distribution of each pair of margins is estimated by evaluating the likelihood of a hidden Markov model (HMM) on the corresponding projection profile on the prior region, with a state constraint at the margin position. The illustration below shows a scanned page with the prior (on the left) and posterior (on the right) regions for the score page highlighted in black. The gray level reflects the uncertainty. The HMM in the middle is the one used for this task (See [8] for the exact set of parameters).

In the above example the goal is only to remove the possible black bands on the border of the document. We use the same technique to split the page to the level of symbol groups on the basis of the projection profile and center of mass profile.

For each located symbol group, further decoding must be carried out to identify what the group is made of. It might be a single isolated symbol such as a clef or a time signature, but also a compound object such as a chord or a complex group of notes. We classify the symbol group on the basis of its dimension and profile. The illustration below shows three different groups, with a possible HMMs for modeling their vertical projection profile. Our actual model combines horizontal and vertical projection profiles, as well as horizontal centered moment of inertia. The HMM parameters are estimated from a training set using the segmental k-means algorithm.

If the object is a note group, we have to locate its graphic primitives, and attach to them the appropriate musical semantic such as quarter note, note stem, sharp. We use the Hough transform (HT) and a low-band filter to locate the main graphic primitives, i.e., straight lines and note heads. They are then grouped according to music notation rules, and noise-sensitive graphics such as dots (for dotted notes) are searched for, only at the place where they make sense according to the context. The figure below shows how the HT can be used to locate the position and orientation of lines in a note group. The HT gives a precise estimate of the position and angle of every possible straight line in the image, regardless of any fragmentation.

References


---

[2] This formula is included verbatim in our MSR model.