An Empirical Comparison of the Performance of PPM Variants on a Prediction Task with Monophonic Music

Marcus Pearce*
* Department of Computing, City University, Northampton Square, London EC1V OHB
m.t.pearce@city.ac.uk

Geraint Wiggins†
† Department of Computing, City University, Northampton Square, London EC1V OHB
geraint@city.ac.uk

Abstract

N-gram models have been employed for a number of musical tasks including the development of practical applications providing computational support for creative individuals as well as theoretical studies of creative processes. Our goal in this research is to evaluate, in an application independent manner, some recent techniques for improving the performance on monophonic music of a subclass of such models based on the Prediction by Partial Match (PPM) algorithm. These techniques include the use of escape method C, interpolated smoothing and unbounded orders. We have applied these techniques incrementally to eight melodic datasets using cross entropy computed by 10-fold cross-validation on each dataset as our performance metric. The results demonstrate statistically significant performance improvements afforded by the use of all three techniques. We discuss these findings in terms of previous research carried out in the field of data compression and with natural language and music corpora and present some directions for future research. It is our hope that these improvements may be applied usefully to specific musical tasks.

1 Introduction

N-gram models have been employed for a number of research tasks in music including the development of practical applications providing computational support for creative individuals as well as theoretical studies of creative processes. In the former category, we cite models for computer-assisted composition (Ames, 1989; Assayag et al., 1999; Hall & Smith, 1996), automatic improvisation (Lartillot et al., 2001) and music information retrieval (Pickens et al., 2002); in the latter category, *n*-gram models have been used for stylistic analysis of music (Conklin & Witten, 1995; Dubnov et al., 1998; Ponsford et al., 1999) and cognitive modelling of music perception (Ferrand et al., 2002; Reis, 1999).

Our goal here is to investigate the performance of a subset of such models on a range of monophonic music data in an application independent manner. We are concerned, in particular, with the application to music data of a particular technique for combining the predictions of *n*-gram models called *Prediction by Partial Match* (PPM – Cleary & Witten, 1984) which forms the central component in some of the best performing data compression algorithms currently available (Bunton, 1997). PPM has previously been applied to natural language data (Chen & Goodman, 1999) and to music data (Conklin & Witten, 1995). Since its introduction, a great deal of research has focused on improving the compression performance of PPM models and our specific aim is to evaluate the performance of these improved models on a range of monophonic music. It is our hope that these improvements, evaluated here in an application independent manner, may then be applied usefully to some of the specific musical tasks cited above.

The paper is organised as follows. In §2, we introduce *n*-gram modelling in general and the PPM scheme in particular, as well as the information-theoretic performance metrics we shall use. Much of the background for this research is drawn from the fields of statistical language modelling (Manning & Schütze, 1999) and text compression (Bell et al., 1990). We hope to demonstrate that practical techniques and methodologies from these fields can be usefully applied in the modelling of music. As noted above, *n*-gram models have been applied to a number of musical tasks and in §3, we discuss research in the musical domain which uses related models and methodologies. The data and experimental methodology employed are described in §4. The results of our experiments are presented in §5, discussed in §6 and, in §7, we conclude by presenting a number of useful directions for future research.

2 Background

2.1 *N*-gram Models

For the purpose of describing this research we shall characterise the acquisition of knowledge about melodic music as a sequence learning problem (Dietterich & Michalski, 1986). The objects of interest are sequences of events where each event consists of a finite set of attributes and each attribute may assume a value drawn from some fi-

$$p(e_i|e_{i-n+1}^{i-1}) = \begin{cases} \alpha(e_i|e_{i-n+1}^{i-1}) & \text{if } c(e_i|e_{i-n+1}^{i-1}) > 0 \\ \gamma(e_{i-n+1}^{i-1})p(e_i|e_{i-n+2}^{i-1}) & \text{if } c(e_i|e_{i-n+1}^{i-1}) = 0 \end{cases} \quad (1)$$

nite alphabet ξ. Here our events are musical notes as notated on a score and we restrict ourselves to the attribute of chromatic pitch. We shall use the notation $e_i^j \in \xi^*$ to denote a sequence of events $e_i \ldots e_j$ where $i, j \in \mathbb{N}^+$ and ξ^* denotes the set of all sequences composed of members of ξ including the empty sequence ε. The goal of sequence learning is to derive from example sequences a model which estimates the probability function $p(e_i|e_1^{i-1})$.

If we make the assumption that the probability of the next event depends only on the previous $n-1$ events, for some $n \in \mathbb{N}^+$:

$$p(e_i|e_1^{i-1}) \approx p(e_i|e_{i-n+1}^{i-1})$$

then we have an $(n-1)^{th}$ order Markov model or *n*-gram model. An *n*-gram is a sequence e_{i-n+1}^{i} consisting of a *context* e_{i-n+1}^{i-1} and a single-event *prediction* e_i. Since the use of a global order bound imposes assumptions about the nature of the data, the selection of an appropriate *n* is an issue when designing and building *n*-gram models. If the order is too high, the model will overfit the training data and fail to capture enough statistical regularity; low order models, on the other hand, suffer from being too general and failing to represent enough of the structure present in the data. The appropriate order for any particular corpus of data can only be determined experimentally.

An *n*-gram *parameter* is the probability of the prediction occurring immediately after the context. The parameters are typically estimated on some corpus of example sequences. Of several different means of estimating *n*-gram parameters, the simplest is *maximum likelihood* (ML) estimation which estimates the parameters as:

$$p(e_i|e_{i-n+1}^{i-1}) = \frac{c(e_i|e_{i-n+1}^{i-1})}{\sum_{e \in \xi} c(e|e_{i-n+1}^{i-1})}$$

where $c(e_{i-n+1}^{i})$ denotes the frequency count for *n*-gram e_{i-n+1}^{i}.

Due to data sparseness, problems arise when using fixed order ML models due to the occurrence of as yet unseen *n*-grams. In particular, if a novel *n*-gram context is encountered or a novel symbol occurs in an existing context (the zero-frequency problem – see Witten & Bell, 1991), the ML estimate will be zero. In these situations, the estimated probability of a novel *n*-gram will be too low and consequently the estimated probability of *n*-grams with non-zero counts will be too high. In addition, the information theoretic performance measures that we shall use (see §2.2) require that every symbol is predicted with non-zero probability.

In statistical language modelling, a set of techniques known collectively as *smoothing* are commonly used to address these problems. The central idea of smoothing is to adjust the ML estimates in order to generate probabilities for as yet unencountered *n*-grams. This is typically achieved by combining the distributions generated by an *h*-gram model with some fixed *global order bound h* with distributions less sparsely estimated from lower order *n*-grams. Most existing smoothing techniques can be expressed using the framework described in Equation 1 (Kneser & Ney, 1995).

If an *n*-gram e_{i-n+1}^{i} occurs with a non-zero count then the estimate $\alpha(e_i|e_{i-n+1}^{i-1})$ is used; otherwise, we recursively *backoff* to a scaled version of the $(n-2)^{th}$ order distribution $p(e_i|e_{i-n+2}^{i-1})$ where the scaling factor $\gamma(e_i|e_{i-n+1}^{i-1})$ is chosen to ensure that each conditional distribution sums to one: $\sum_{e \in \xi} p(e|e_{i-n+1}^{i-1}) = 1$. Recursion is typically terminated with the zeroth order model or by taking a uniform distribution over ξ. Different smoothing algorithms vary in the methods used for computing $\alpha(e_i|e_{i-n+1}^{i-1})$ and $\gamma(e_i|e_{i-n+1}^{i-1})$.

An alternative to backoff smoothing is *interpolated* smoothing in which the probability of an *n*-gram is always estimated by recursively computing a weighted combination of the $(n-1)^{th}$ order distribution with the $(n-2)^{th}$ order distribution as described in Equation 2.

$$p(e_i|e_{i-n+1}^{i-1}) = \alpha(e_i|e_{i-n+1}^{i-1}) + \gamma(e_{i-n+1}^{i-1})p(e_i|e_{i-n+2}^{i-1}) \quad (2)$$

Detailed empirical comparisons of the performance of different smoothing techniques have been conducted on natural language corpora (Chen & Goodman, 1999; Martin et al., 1999). One of the results of this work is the finding that, in general, interpolated smoothing techniques outperform their backoff counterparts. Chen & Goodman (1999) found that this performance advantage is restricted, in large part, to *n*-grams with low counts and suggest that the improved performance of interpolated algorithms is due to the fact that low order distributions provide valuable frequency information about such *n*-grams.

2.2 Performance Metrics

It is common in the field of statistical language modelling to use information theoretic measures to evaluate statistical models of language. Given a discrete random variable $\mathcal{X}$ distributed over an alphabet ξ according to a probability distribution *P* such that the individual probabilities are independent and sum to one, the entropy $H(P)$ is defined as:

$$H(P) = -\sum_{e \in \xi} p(e) \log_2 p(e) \quad (3)$$

Shannon's fundamental coding theorem (Shannon, 1948) states that entropy provides a lower bound on the average number of binary bits per symbol required to encode an outcome of $\mathcal{X}$. The corresponding upper bound occurs in the case where each symbol in the alphabet has an equal probability of occurring: $\forall e \in \xi, p(e) = \frac{1}{|\xi|}$.

$$H_{max}(\xi) = \log_2 |\xi| \tag{4}$$

Entropy has an alternative interpretation in terms of the degree of uncertainty that is involved in selecting a symbol from an alphabet: greater entropy implies greater uncertainty.

In practice, we rarely know the true probability distribution of the stochastic process and use a model to approximate the probabilities in Equation 3. *Cross entropy* is a quantity which represents the divergence between the entropy calculated from these estimated probabilities and the source entropy. Given a model which assigns a probability of $p_m(e_1^j)$ to a sequence of outcomes of $\mathcal{X}$, e_1^j, we can calculate the cross entropy $H_m(e_1^j)$ of model m with respect to event sequence e_1^j simply by having very large sequences of outcomes available. In particular, if we make some assumptions about the stochastic process which generated the sequence, the cross entropy $H_m(e_1^j)$ may be calculated as:[1]

$$H_m(e_1^j) \quad = \quad -\frac{1}{j}\sum_{i=1}^{j} \log_2 p_m(e_i|e_1^{i-1}) \tag{5}$$

Since $H_m(e_1^j)$ provides an estimate of the number of binary bits required on average to encode a symbol in e_1^j in the most efficient manner and there exist techniques, such as arithmetic coding (Witten et al., 1987), which can produce near optimal codes, cross entropy provides a direct performance metric in the realm of data compression. However, cross entropy has a wider use in the evaluation of statistical models. Since it provides us with a measure of how uncertain a model is, on average, when predicting a sequence of events, it can be used to compare the performance of different models on some corpus of data. In statistical language modelling, cross entropy measures are commonly used: "For a number of natural language processing tasks, such as speech recognition, machine translation, handwriting recognition, stenotype transcription and spelling correction, language models for which the cross entropy is lower lead directly to better performance." (Brown et al., 1992, p. 39).

[1] In particular, we assume that the process is *stationary* and *ergodic*. A stochastic process is stationary if the probability distribution governing the emission of symbols is stationary over time (i.e., independent of the position in the sequence) and ergodic if sufficiently long sequences of events generated by it can be used to make inferences about its typical behaviour.

2.3 The PPM Algorithm

2.3.1 Overview

Prediction by Partial Match (Cleary & Witten, 1984) is a data compression scheme the central component of which is an algorithm for performing backoff smoothing of n-gram distributions. Variants of the PPM scheme have set the standard in lossless data compression since its introduction (Bunton, 1997). We shall describe several of these variants in terms of Equations 1 and 2 where recursion is terminated with a model which predicts each event $e \in \xi$ with equal probability mass, $\frac{1}{|\xi|}$. This model is usually referred to as the $order - 1$ model and allows for the prediction of events which have yet to be encountered.

2.3.2 The Zero-frequency Problem and Escaping

We shall now describe how the probability estimates $\alpha(e_i|e_{i-n+1}^{i-1})$ and $\gamma(e_i|e_{i-n+1}^{i-1})$ in Equations 1 and 2 are computed in PPM models. The problem is usually characterised by asking how we estimate $\gamma(e_i|e_{i-n+1}^{i-1})$ – the amount of probability mass to assign to events which are novel in the current context e_{i-n+1}^{i-1}. $\alpha(e_i|e_{i-n+1}^{i-1})$ is then set such that the distributions sum to one. As noted by Witten & Bell (1991), there is no sound theoretical basis for choosing these *escape probabilities* in the absence of *a priori* knowledge about the data being modelled. As a result, although several schemes exist, their relative performance on any particular task can only be determined experimentally. In the following discussion, $t(e_i^j)$ denotes the total number of symbol types that have occurred in context e_i^j.

Method B (Cleary & Witten, 1984) classifies a symbol occurring in a given context as novel unless it has already occurred *twice* in that context. This is achieved by subtracting one from all the counts and has the effect of filtering out anomalies. In addition, the escape count increases as more types are observed.

$$\gamma(e_i|e_{i-n+1}^{i-1}) = \frac{t(e_{i-n+1}^{i-1})}{\sum_{e\in\xi} c(e_{i-n+1}^{i-1})}$$
$$\alpha(e_i|e_{i-n+1}^{i-1}) = \frac{c(e_i|e_{i-n+1}^{i-1}) \; - \; 1}{\sum_{e\in\xi} c(e_{i-n+1}^{i-1})}$$

Method C (Moffat, 1990) retains from method B the effect that the escape count increases as more types are observed but symbols are predicted immediately.

$$\gamma(e_i|e_{i-n+1}^{i-1}) = \frac{t(e_{i-n+1}^{i-1})}{\sum_{e\in\xi} c(e_{i-n+1}^{i-1}) + t(e_{i-n+1}^{i-1})}$$
$$\alpha(e_i|e_{i-n+1}^{i-1}) = \frac{c(e_i|e_{i-n+1}^{i-1})}{\sum_{e\in\xi} c(e|e_{i-n+1}^{i-1}) + t(e_{i-n+1}^{i-1})}$$

One particular smoothing technique called *Witten-Bell smoothing*, often used in statistical language modelling, is based on escape method C (Manning & Schütze, 1999).

These escape methods have been subjected to empirical evaluation in data compression experiments which demonstrate that method C typically yields better performance than method B (Bunton, 1997; Moffat et al., 1994).

2.3.3 Interpolated Smoothing

We have discussed the difference between backoff and interpolated smoothing in §2.1 and shown how they can be described within the same framework. While the original PPM algorithm uses a backoff strategy (called *blending*), Bunton (1997) has experimented with using interpolated smoothing within PPM. Bunton notes that for blending (and other backoff methods), the estimates for novel events are slightly inflated while the estimates for events which are not novel are slightly deflated. Replacing blending with interpolated smoothing remedies this and yields significant and consistent improvements in compression performance (Bunton, 1997).

2.3.4 Unbounded Length Contexts

One of the goals of *universal* modelling is to make minimal assumptions about the nature of the stochastic processes responsible for generating observed data. As we discussed in §2.1, *n*-gram models make assumptions about these processes to the effect that the probability of an event depends only on the previous $n-1$ events. Cleary & Teahan (1997) describe an extension to PPM, called PPM*, which eliminates the need to impose an arbitrary order bound. The policy used to select a maximum order context can be freely varied depending on the situation.

A context e_i^j is said to be *deterministic* when it makes exactly one prediction: $t(e_i^j) = 1$. Cleary & Teahan (1995) have found that for such contexts the observed frequency of novel events is much lower than expected based on a uniform prior distribution. As a consequence, the entropy of the distributions estimated in deterministic contexts will tend to be lower than in non-deterministic contexts. Since the event will have occurred at least as many times in the lowest order matching deterministic context as any of the other matching deterministic contexts, it will produce the lowest-entropy probability distribution (Bunton, 1997). Cleary & Teahan (1997) exploit this in PPM* by selecting the shortest deterministic matching context if one exists or otherwise selecting the longest matching context. Unfortunately, the original PPM* implementation provided modest improvement in compression performance over the original order bounded PPM. When combined with interpolated smoothing, however, PPM* does outperform the corresponding order bounded PPM models in data compression experiments (Bunton, 1997).

2.3.5 Implementation Issues

Since PPM* does not impose an order bound, all subsequences of the input sequence must be stored which makes for increased demands on computational resources. Suffix-tree representations provide a space-efficient means of achieving this end (Bunton, 1997; Larsson, 1996). We have implemented our PPM models as suffix trees using the online construction algorithm described by Ukkonen (1995). The application of this algorithm to the construction of PPM models was first described by Larsson (1996) and the construction developed independently by Bunton (1997) is similar to the Ukkonen-Larsson algorithm in many respects. In addition to being online, these algorithms have linear time and space complexity and, as demonstrated by Bunton (1997), the resulting models have optimal space requirements (in contrast to the original PPM* implementation). The existence of path compressed nodes in suffix trees complicates the storage of frequency counts and their use in prediction. We have followed the strategies for initialising and incrementing the counts employed by Bunton (1997) to address these complications.

2.4 Long- and Short-term Models

In data compression, a model which is typically empty initially is constructed incrementally as more of the input data is seen. However, experiments with PPM using an initial model that has been derived from a training text demonstrate that pre-training the model, both with related and with unrelated texts, significantly improves compression performance (Teahan & Cleary, 1996). A complementary approach is often used in the literature on statistical language modelling where improved performance is obtained by augmenting *n*-gram models derived from the entire training corpus with *cache* models which are constructed dynamically from a portion of the recently processed text (Kuhn & De Mori, 1990).

Conklin (1990) has employed similar ideas with music data by using both a *long-term model* (LTM) and a *short-term model* (STM). While the LTM parameters are estimated on the entire training corpus, the STM is constructed online for each composition in the test set and is discarded after the relevant composition has been processed. The predictions of both models are combined to provide an overall probability estimate for the current event. The motivation for doing so is to take advantage of recently occurring *n*-grams whose structure and statistics may be specific to the individual composition being predicted.

A simple way of achieving the combination of predictions from the LTM and STM is to use a weighted average of the individual predictions (Conklin, 1990). Let $e \in \xi$ be the current symbol to be predicted, M be a set $\{ltm, stm\}$ containing the LTM and STM and $p_m(e)$ be the probability assigned to symbol e by model $m \in M$. The weighted mean of the two predictions is:

$$p(e) = \frac{\sum_{m \in M} w_m p_m(e)}{\sum_{m \in M} w_m} \quad (6)$$

Conklin describes a method for calculating the weights, w_{ltm} and w_{stm} based on the entropies of the distributions generated by the LTM and STM such that greater entropy (and hence uncertainty) is associated with a lower weight. Let P_m be the probability distribution generated by model m. The *relative entropy* of a model is:

$$H_{relative}(m) = \begin{cases} H(P_m)/H_{max}(\xi) & \text{if } H_{max}(\xi) > 0 \\ 1 & \text{otherwise} \end{cases}$$

where H and H_{max} are as defined in Equations 3 and 4 respectively. The weight $w_m$of model m is computed as $w_m = H_{relative}(m)^{-b}$ where $b \in \mathbb{N}$ is a parameter giving an exponential bias towards models with lower relative entropy. The combined use of long- and short-term models yields better prediction performance than either the LTM or STM used individually (Conklin, 1990). Finally, Conklin & Witten (1995, p. 61) have used a different scheme based on the Dempster-Schaffer theory of evidence for combining the predictions of long- and short-term models "with some success" but do not provide any details of the scheme or the performance improvements it yielded.

3 Related Work

N-gram models have been used for music related tasks since the 1950s when they were investigated as tools for composition and analysis (see e.g., Brooks Jr. et al., 1957; Hiller & Isaacson, 1959; Pinkerton, 1956). Since extensive reviews of this early research exist (Ames, 1989; Hiller, 1970), we shall focus here on more recent approaches.

Ponsford et al. (1999), for example, have applied trigrams and tetragrams (without smoothing) to the modelling of harmonic movement in a corpus of 84 seventeenth century *sarabandes*. The aim was to find out how adequate a simple *n*-gram model would be for the description and generation of harmonic movement in the style. Higher order structure was represented in the corpus through the annotation of events delimiting bars, phrases and entire pieces. A number of pieces were generated from the models and subjected to an informal stylistic analysis. The generated harmonies were "characteristic of the training corpus in terms of harmony transitions, the way in which pieces, phrases and bars begin and end, modulation between keys and the relation between harmony change and metre" (Ponsford et al., 1999, p. 169). The generation of features such as enharmony, which was not present in the corpus, and weak final cadences was attributed mainly to the use of low order models.

Conklin & Witten (1995, see also Conklin 1990) developed PPM models of the soprano lines of 100 of the chorales harmonised by J.S. Bach. The escape method used was B and both long- and short-term models were employed. The global order bounds of the LTM and STM were set at 3 and 2 respectively and the predictions combined using a Dempster-Schaffer scheme. One of the central features of this work was the representation of multiple attributes, or *viewpoints*, of a melodic sequence. Conklin & Witten (1995) describe a number of *multiple viewpoint systems* consisting of several PPM models trained on different viewpoints whose predictions were combined in the same manner as described in §2.4. Several evaluation techniques were employed. First, split-sample validation with a training set of 95 compositions and a test set of five compositions was used to compare the performance (in terms of cross entropy) of different multiple viewpoint systems. Conklin & Witten were able to derive multiple viewpoint systems whose entropy was significantly lower than that of a single viewpoint system modelling chromatic pitch. The second means of evaluation was a *generate-and-test* approach from which Conklin & Witten concluded that the generated compositions seemed to be "reasonable". Finally, Witten et al. (1994) conducted an empirical study of the sequential chromatic pitch predictions made by human listeners on the same test set of compositions. The entropy profiles derived from the experimental results for each composition were strikingly similar in form to those generated by the model.

Hall & Smith (1996) have extended the approach used by Conklin & Witten (1995) to a corpus of 58 twelve-bar blues compositions. The aim was to develop a compositional tool that would automatically generate a melody when supplied with a twelve-bar blues harmonic structure. In order to model pitch, zero, first and second order models were derived from 48 compositions in the corpus. Separate first and second order models were derived for each individual chord occurring in the corpus. Rhythm was represented using an alphabet of short rhythmic patterns (e.g., two semiquavers followed by a quaver) and zero, first and second order models were derived from the training set over this alphabet. The model was evaluated by asking 198 human subjects to judge which of a pair of compositions (of which one was human- and the other machine-composed) was machine-generated. The data consisted of the ten remaining compositions in the corpus and ten compositions randomly selected from the model's output all of which were played to the subjects over a standard harmonic background. Statistical analysis of the results demonstrated that the subjects were unable to distinguish reliably between the human and machine generated compositions.

Reis (1999) has extended the work of Conklin & Witten (1995) in a different direction through the incorporation of psychological constraints in *n*-gram models. In particular, he argues that storing all *n*-grams (with order less the the global bound) which occur in the data is highly inefficient and unlikely to accurately depict the manner in which humans represent melodies. Reis describes a

ID	Description
0	Folk songs and Ballads from Nova Scotia, Canada
1	Chorale soprano melodies harmonised by J.S. Bach
2	Alsatian folk songs from the Essen Folk Song Collection
3	Yugoslavian folk songs from the Essen Folk Song Collection
4	Swiss folk songs from the Essen Folk Song Collection
5	Austrian folk songs from the Essen Folk Song Collection
6	German folk songs from the Essen Folk Song Collection (`kinder` dataset)
7	Chinese folk songs from the Essen Folk Song Collection (`shanxi` dataset)

Table 1: Melodic datasets used in this research.

model which segments the data according to perceptual cues such as contour changes or unusually large pitch or duration intervals. The order of the *n*-grams stored by the model is then determined by the sequence of events back to the previous segmentation point. If a novel *n*-gram is encountered during prediction, the distribution delivered by the variable order model is smoothed with a uniform distribution over the alphabet. The model also incorporates perceptually guided predictions for more than one step ahead. The performance of the model was evaluated on the chorale dataset used by Conklin & Witten (1995) and folk melodies from the Essen Folk Song Collection (Schaffrath, 1995) using entropy as the performance metric with a split sample experimental design. Although the results demonstrated that the model failed to outperform that of Conklin & Witten (1995), the work is useful since it addresses the question of which segmentation and modelling strategies work best when model-size is limited.

4 Experimental Methodology

4.1 Model Parameters

A PPM model has been implemented in Common Lisp such that each of the variant features described in §2.3 are accepted as parameters to the top-level call. We shall use the following shorthand to refer to each of the model parameters: the model type is indicated by 'LTM' and 'STM' for the long- and short-term models respectively; the escape method is indicated explicitly by 'B' or 'C'; the order bound is indicated by an integer or '*' if unbounded; and interpolated smoothing is indicated by an 'I' (blending is the default). Thus, for example, a PPM long-term model with escape method C, unbounded order and interpolated smoothing is denoted by 'LTMC*I'. When combined with a short-term model with the same parameters, the model would be denoted by 'LTMC*I-STMC*I' (for readability the two models are separated by a hyphen).

4.2 Data

The aim of this research was to assess the performance of PPM variants over a range of different musical styles. The datasets used were all obtained in the `**kern` format (Huron, 1997) from the *Centre for Computer Assisted Research in the Humanities* (CCARH) at Stanford University, California (see `http://www.ccarh.org`) and the *Music Cognition Laboratory* at Ohio State University (see `http://kern.humdrum.net`). During preprocessing, tied notes were collapsed together and the chromatic pitch of each event was converted into a MIDI note number where 60 represents middle C. Each composition therefore consists of a sequence of integers each of which represents a chromatic pitch.

The datasets themselves contain purely melodic music. The first is a collection of 152 folk songs and ballads from Nova Scotia, Canada collected between 1928 and 1932 by Helen Creighton. The dataset is freely available from the *Music Cognition Laboratory* at Ohio State University. The second dataset contains 185 of the chorale soprano melodies harmonised by J.S Bach (BWV 253 to BWV 438) and is freely available from CCARH. The remaining datasets come from the Essen Folk Song Collection (EFSC – Schaffrath, 1992, 1994). The collection comprises 6,252 (mostly) European folk melodies collected and encoded under the supervision of Helmut Schaffrath at the University of Essen in Germany between 1982 and 1994. A dataset containing all the compositions in the collection is published and distributed by *CCRAH* (Schaffrath, 1995) and an additional dataset of 2580 Chinese folk melodies is available on request. The six datasets from the EFSC used in this research contained respectively 91 Alsatian folk melodies, 119 Yugoslavian folk melodies, 93 Swiss folk melodies, 104 Austrian folk melodies, 213 German folk melodies (from dataset `kinder`) and 237 Chinese folk melodies (from dataset `shanxi`).

Each dataset is assigned a natural ID as shown in Table 1 and will be referred to henceforth by this ID. More detailed information about each dataset, including the number of compositions and events contained in the dataset and the number of chromatic pitches from which the dataset is composed, can be found in Table 2.

ID	No. Compositions	No. Events	Mean Events/Composition	Alphabet Size
0	152	8553	56.270	25
1	185	9227	49.876	21
2	91	4496	49.407	32
3	119	2691	22.613	25
4	93	4586	49.312	34
5	104	5306	51.019	35
6	213	8393	39.403	27
7	237	11056	46.650	41

Table 2: Detailed information about the datasets used in this research.

4.3 Performance Evaluation

Many methods have been used to evaluate the performance of statistical models of music, some of which have been described in §3. We have followed the resampling approach using entropy as a performance metric for two reasons: first, entropy has an unambiguous interpretation in terms of model uncertainty on unseen data (see §2.2); and second, entropy bears a direct relationship with performance in compression and indirectly correlates with the performance of *n*-gram models on practical natural language tasks and is widely used in both these fields. These factors support its use in an application independent evaluation such as this.

Conklin & Witten (1995) used a *split-sample* (or *held-out*) experimental paradigm in which the data is divided randomly into two disjoint sets, a training set and a test set; the *n*-gram parameters are then estimated on the training set and the cross entropy of the test set given the resulting model is computed using Equation 5. Conklin & Witten used a training set of 95 melodies and a test set of 5 melodies. Although commonly used, split-sample validation suffers from two disadvantages: first, it reduces the amount of data available for both training and testing; and second, with small datasets it provides a biased estimate of the true entropy of the corpus. A simple way of addressing these limitations is to use *k-fold cross-validation* in which the data is divided into k subsets of approximately equal size. The model is trained k times each time leaving out a different subset to be used for testing and an average of the k cross entropy values thus obtained is then computed. In machine learning research, differences in model performance as assessed by resampling techniques, such as cross-validation, are often analysed for significance using statistical tests such as the t-test (Dietterich, 1998; Mitchell, 1997).

5 Results

To illustrate the performance improvements obtained with the PPM variants discussed in this paper, we have successively applied escape method C, unbounded orders and interpolated smoothing to our emulation of the model used by Conklin & Witten (1995) which is described in our framework as LTMB3-STMB2 (see §2). The results are shown in Table 3 where each result was obtained using ten-fold cross-validation. Paired t-tests confirmed the significance of the improvements afforded by incrementally applying escape method C [$t = 32.53, df = 79, p < 0.001$], unbounded orders [$t = 7.34, df = 79, p < 0.001$] and interpolated smoothing [$t = 24.57, df = 79, p < 0.001$].[2]

6 Discussion

Before discussing the results presented in §5, some words on the methodology employed are in order. Our goal was to demonstrate that a number of techniques improve the prediction performance of PPM models on monophonic music data. We have approached this task by using cross entropy of the models as our performance metric and applying ten-fold cross validatory resampling on eight monophonic datasets. Since we have been concerned with optimising average performance over all eight datasets, the best performing model will not necessarily correspond to the best performing model on any single dataset. However, the results increase our confidence that this model will perform well on a given dataset without requiring further empirical investigation of that dataset.

Furthermore, since we have applied the variant techniques incrementally, there is no guarantee that the resulting model reflects the global optimum in the space of possible LTM and STM parameterisations. Once again, we note that our aim was to demonstrate how some variant techniques can improve the performance of PPM models and consequently our interest is in the relative, rather than absolute, performance of the PPM variants.

In our experiments, escape method C yielded consistent improvements in model performance over method B. Although, as noted in §2.3.2, there is no principled a priori means of selecting the escape method, these results are consistent with those obtained in data compression experiments (Bunton, 1997; Moffat et al., 1994; Witten & Bell, 1991). The use of unbounded orders, as described in §2.3.4, improves the average performance of PPM models

[2] These tests were performed over all 10 resampling sets of each dataset ($n = 80$) which are not shown in Table 3.

Dataset	LTMB3-STMB2	LTMC3-STMC2	LTMC*-STMC*	LTMC*I-STMC*I
0	2.901	2.604	2.585	2.465
1	2.665	2.470	2.451	2.327
2	3.080	2.759	2.705	2.602
3	2.960	2.686	2.643	2.558
4	3.007	2.671	2.556	2.483
5	3.284	2.852	2.728	2.661
6	2.536	2.249	2.167	2.085
7	3.060	2.762	2.770	2.642
Average	2.937	2.632	2.576	2.478

Table 3: Performance improvements to our emulation of the model used by Conklin & Witten (1995).

though to a lesser extent and with less consistency across the datasets than applying escape method C. This finding agrees with those obtained in experiments on data compression (Bunton, 1997) and is likely to be due to the fact that the optimal order bound varies between datasets. As noted by Bunton (1997, p. 90), order bound experiments "provide more information about the nature of the test data, rather than the universality of the tested algorithms." Finally, the use of interpolated smoothing consistently improves model performance regardless of the dataset. This is in agreement with results obtained in experiments in data compression (Bunton, 1997) and on natural language corpora (Chen & Goodman, 1999; Martin et al., 1999). The reason appears to derive from the fact that backoff smoothing (of which blending is an example) consistently underestimates the probabilities of non-novel events (Bunton, 1997) for which the low order distributions provide valuable frequency information.

7 Conclusions

By way of conclusion, we would like to present some directions that we feel would be profitable to explore in future research. The first set of suggestions concern model development. First, an empirical comparison of the performance of the Demster-Schaffer scheme used by Conklin & Witten (1995) to combine the predictions of the LTM and STM with other techniques (including the weighted mean employed here) would be useful for future model developers. Second, Bunton (1997) describes an information-theoretic state selection mechanism which replaces the original state selection used in PPM* (see §2.3.4) and which consistently improves performance in data compression experiments. It remains to be seen whether this mechanism can be fruitfully applied with music data. Finally, the extension of the methodology used in this research to comparisons between different modelling approaches could yield interesting results. It would be useful, for example, to compare the performance of the PPM variants analysed here with that of models using other smoothing techniques commonly used in statistical language modelling, such as Katz backoff (Katz, 1987) and Kneser-Ney smoothing (Kneser & Ney, 1995), and models based on the Lempel-Ziv dictionary compression algorithm as used by Dubnov, Assayag and their colleagues (Assayag et al., 1999; Dubnov et al., 1998; Lartillot et al., 2001), the prediction suffix automata used by Lartillot et al. (2001) and Triviño-Rodriguez & Morales-Bueno (2001) and the neural network models described by Mozer (1994).

Our second set of suggestions concern the data used. It should be emphasised that we have restricted our attention to a single attribute of musical sequences: chromatic pitch. None of the conclusions reached in this research can be guaranteed to hold for other attribute domains and representations; we shall need similarly detailed experiments to assess whether the performance improvements recorded here remain valid with these new representations. Therefore, an important consideration is the extension of the approach to other attributes of musical events and more sophisticated representations of musical works. Conklin & Witten (1995), for example, describe several means of deriving more abstract representations of the musical surface as well as developing methods for combining the predictions of n-gram models of these representations. In a similar vein, we consider it important to extend the approach to homophonic and polyphonic music. The issue of representing such music for training statistical models is discussed by, for example, Assayag et al. (1999), Conklin (2002), Pickens et al. (2002) and Ponsford et al. (1999). Since the results obtained here are in broad agreement with those obtained in data compression and statistical language modelling experiments, we expect the performance improvements to hold some degree of generality and to carry over to these more sophisticated representations of music.

Our final suggestions are methodological. The first concerns the fact that many of the directions cited above concern comparisons between different models. Standard corpora exist for comparing model performance in both the data compression and statistical language modelling communities: e.g., the Calgary corpus (Bell et al., 1990) and LOB corpus (Johansson et al., 1986) respectively. Such standardisation facilitates the objective and empirical comparison of different models and would be highly beneficial to the music processing community. Another

methodological issue concerns the validity of entropy as a measure of performance; in order to address this question we need detailed empirical studies of the relationship between entropy measures and model performance on a range of musical tasks such as those outlined in §1. In the meantime, we hope that the techniques described in this paper can be profitably applied to practical musical tasks and that the consequent reduction in cross entropy will translate into actual performance improvement on these tasks.

Acknowledgements

We would like to thank Darrell Conklin and Kerry Robinson as well as two anonymous reviewers for their comments on previous drafts of this paper. Marcus Pearce is supported by EPSRC via studentship number 00303840.

References

Ames, C. (1989). The Markov process as a compositional model: a survey and tutorial. *Leonardo*, *22*(2), 175–187.

Assayag, G., Dubnov, S., & Delerue, O. (1999). Guessing the composer's mind: applying universal prediction to musical style. In *Proceedings of the 1999 International Computer Music Conference*, (pp. 496–499). San Francisco: ICMA.

Bell, T. C., Cleary, J. G., & Witten, I. H. (1990). *Text Compression*. Englewood Cliffs, NJ: Prentice Hall.

Brooks Jr., F. P., Hopkins, A. L., Neumann, P. G., & Wright, W. V. (1957). An experiment in musical composition. *IRE Transactions on Electronic Computers*, *EC-6*(1), 175–182.

Brown, P. F., Della Pietra, S. A., Della Pietra, V. J., Lai, J. C., & Mercer, R. L. (1992). An estimate of an upper bound on the entropy of English. *Computational Linguistics*, *18*(1), 32–40.

Bunton, S. (1997). Semantically motivated improvements for PPM variants. *The Computer Journal*, *40*(2/3), 76–93.

Chen, S. F. & Goodman, J. (1999). An empirical study of smoothing techniques for language modelling. *Computer Speech and Language*, *13*(4), 359–394.

Cleary, J. G. & Teahan, W. J. (1995). Some experiments on the zero-frequency problem. In Storer, J. A. & Cohn, M. (Eds.), *Proceedings of the IEEE Data Compression Conference*. Washington, DC: IEEE Computer Society Press.

Cleary, J. G. & Teahan, W. J. (1997). Unbounded length contexts for PPM. *The Computer Journal*, *40*(2/3), 67–75.

Cleary, J. G. & Witten, I. H. (1984). Data compression using adaptive coding and partial string matching. *IEEE Transactions on Communications*, *32*(4), 396–402.

Conklin, D. (1990). Prediction and entropy of music. Master's thesis, Department of Computer Science, University of Calgary. Available as Technical Report 1990–390–14.

Conklin, D. (2002). Representation and discovery of vertical patterns in music. In Anagnostopoulou, C., Ferrand, M., & Smaill, A. (Eds.), *Proceedings of the Second International Conference of Music and Artificial Intelligence*, (pp. 32–42).

Conklin, D. & Witten, I. H. (1995). Multiple viewpoint systems for music prediction. *Journal of New Music Research*, *24*(1), 51–73.

Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, *10*(7), 1895–1924.

Dietterich, T. G. & Michalski, R. S. (1986). Learning to predict sequences. In R. S. Michalski, J. Carbonell, & T. M. Mitchell (Eds.), *Machine Learning: an Artificial Intelligence Approach*, volume II (pp. 63–106). San Mateo, CA: Morgan Kauffman.

Dubnov, S., Assayag, G., & El-Yaniv, R. (1998). Universal classification applied to musical sequences. In *Proceedings of the 1998 International Computer Music Conference*, (pp. 332–340). San Francisco: ICMA.

Ferrand, M., Nelson, P., & Wiggins, G. (2002). A probabilistic model for melody segmentation. In *Electronic Proceedings of the 2nd International Conference on Music and Artificial Intelligence (ICMAI'2002)*, University of Edinburgh, Scotland.

Hall, M. & Smith, L. (1996). A computer model of blues music and its evaluation. *Journal of the Acoustical Society of America*, *100*(2), 1163–1167.

Hiller, L. (1970). Music composed with computers – a historical survey. In H. B. Lincoln (Ed.), *The Computer and Music* chapter 4, (pp. 42–96). Cornell, USA: Cornell University Press.

Hiller, L. & Isaacson, L. (1959). *Experimental Music*. New York: McGraw–Hill.

Huron, D. (1997). *Humdrum* and *Kern*: selective feature encoding. In E. Selfridge-Field (Ed.), *Beyond MIDI: The Handbook of Musical Codes* (pp. 375–401). Cambridge, MA: MIT Press.

Johansson, S., Atwell, E., Garside, R., & Leech, G. (1986). The tagged lob corpus. ICAME, The Norwegian Computing Centre for the Humanities, Bergen University, Norway.

Katz, S. M. (1987). Estimation of probabilities from sparse data for the language model component of a speech recogniser. *IEEE Transactions on Acoustics, Speech and Signal Processing*, *35*(3), 400–401.

Kneser, R. & Ney, H. (1995). Improved backing-off for for m-gram language modelling. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 1, (pp. 181–184)., Detroit, MI.

Kuhn, R. & De Mori, R. (1990). A cache-based natural language model for speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *12*(6), 570–583.

Larsson, N. J. (1996). Extended application of suffix trees to data compression. In Storer, J. A. & Cohn, M. (Eds.), *Proceedings of the IEEE Data Compression Conference*, (pp. 190–199). Washington, DC: IEEE Computer Society Press.

Lartillot, O., Dubnov, S., Assayag, G., & Bejerano, G. (2001). Automatic modelling of musical style. In *Proceedings of the 2001 International Computer Music Conference*, (pp. 447–454). San Francisco: ICMA.

Manning, C. D. & Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press.

Martin, S., Hamacher, C., Liermann, J., Wessel, F., & Ney, H. (1999). Assessment of smoothing methods and complex stochastic language modeling. In *Proceedings of the Sixth European Conference on Speech Communication and Technology*, (pp. 1939–1942)., Budapest, Hungary.

Mitchell, T. M. (1997). *Machine Learning*. New York: McGraw Hill.

Moffat, A. (1990). Implementing the PPM data compression scheme. *IEEE Transactions on Communications*, *38*(11), 1917–1921.

Moffat, A., Sharman, N., Witten, I. H., & Bell, T. C. (1994). An empirical evaluation of coding methods for multi-symbol alphabets. *Information Processing & Management*, *30*(6), 791–804.

Mozer, M. C. (1994). Neural network music composition by prediction: Exploring the benefits of psychoacoustic constraints and multi–scale processing. *Connection Science*, *6*(2–3), 247–280.

Pickens, J., Bello, J. P., Crawford, T., Dovey, M., Monti, G., & Sandler, M. B. (2002). Polyphonic score retrieval using polyphonic audio queries: A harmonic modeling approach. In *Proceedings of the third International Symposium on Information Retrieval*, (pp. 140–149). IRCAM: Paris, France.

Pinkerton, R. C. (1956). Information theory and melody. *Scientific American*, *194*(2), 77–86.

Ponsford, D., Wiggins, G. A., & Mellish, C. (1999). Statistical learning of harmonic movement. *Journal of New Music Research*, *28*(2), 150–177.

Reis, B. Y. (1999). Simulating music learning: on–line, perceptually guided pattern induction of context models for multiple–horizon prediction of melodies. In *Proceedings of the AISB'99 Symposium on Musical Creativity*, (pp. 58–63). Brighton, UK: SSAISB.

Schaffrath, H. (1992). The ESAC databases and MAPPET software. *Computing in Musicology*, *8*, 66.

Schaffrath, H. (1994). The ESAC electronic songbooks. *Computing in Musicology*, *9*, 78.

Schaffrath, H. (1995). The Essen folksong collection. In D. Huron (Ed.), *Database containing 6,255 folksong transcriptions in the Kern format and a 34-page research guide*. Menlo Park, CA: CCARH.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, *27*(3), 379–423 and 623–656.

Teahan, W. J. & Cleary, J. G. (1996). The entropy of English using PPM-based models. In Storer, J. A. & Cohn, M. (Eds.), *Proceedings of the IEEE Data Compression Conference*. Washington, DC: IEEE Computer Society Press.

Triviño-Rodriguez, J. L. & Morales-Bueno, R. (2001). Using multi-attribute prediction suffix graphs to predict and generate music. *Computer Music Journal*, *25*(3), 62–79.

Ukkonen, E. (1995). On-line construction of suffix trees. *Algorithmica*, *14*(3), 249–260.

Witten, I. H. & Bell, T. C. (1991). The zero-frequency problem: estimating the probabilities of novel events in adaptive text compression. *IEEE Transactions on Information Theory*, *37*(4), 1085–1094.

Witten, I. H., Manzara, L. C., & Conklin, D. (1994). Comparing human and computational models of music prediction. *Computer Music Journal*, *18*(1), 70–80.

Witten, I. H., Neal, R. M., & Cleary, J. G. (1987). Arithmetic coding for data compression. *Communications of the ACM*, *30*(6), 520–541.