

Parsing Incremental para Acompanhamento em Tempo Real

Giordano Cabral¹, Jean-Pierre Briot¹, François Pachet²

¹Laboratoire d'Informatique de Paris 6 – Université Pierre et Marie Curie
8 Rue du Capitaine Scott 75018 Paris – France

²Sony Computer Science Lab Paris
6 Rue Amyot 75005 Paris – France

{Giordano.CABRAL, Jean-Pierre.BRIOT}@lip6.fr, pachet@csl.sony.fr

Abstract. *The incremental parsing (IP) algorithm has been successfully used in real-time applications, due to its efficiency and power of modeling musical style. Musical systems using IP can continue phrases played by a musician in a consistent way. This paper proposes some alterations in the original algorithm in order to allow the development of systems able to respect adaptation and continuity in musical accompaniment.*

Resumo. *O algoritmo de parsing incremental tem sido usado para o desenvolvimento de aplicações musicais em tempo real devido à sua eficiência e capacidade de modelar estilos musicais. Esta capacidade permite a tais sistemas continuarem frases tocadas por um músico, coerentemente ao estilo destas. Este artigo propõe uma alteração no algoritmo de modo a permitir o desenvolvimento de sistemas que além de guardar uma coerência estilística, se adaptem ao que está sendo tocado pelos outros músicos, características básicas do acompanhamento.*

1. Introdução e Problemática

A análise estatística de um corpus de material musical indica possibilidades de recombinação que se adequam às restrições e redundâncias típicas de um determinado modelo. Foi comprovado que a construção de cadeias de Markov [Dubnov et al. 1998] a partir de uma base de exemplos é um tipo de aprendizagem bastante eficaz para modelar o comportamento dinâmico de melodias. Este paradigma foi utilizado tanto na classificação de estilos musicais [Lartillot et al. 2001] quanto em sistemas interativos [Pachet 2002]. Estes últimos são capazes, por exemplo, de continuar frases tocadas por um músico, coerentemente ao estilo destas. Para implementar eficientemente as cadeias de Markov, é utilizado o algoritmo de *parsing* incremental (PI) [Ziv and Lempel 1977], projetado inicialmente como método de compressão.

O presente trabalho toma como ponto de partida o *Continuator*, o estado da arte de tais sistemas. O projeto explorou diversos modos de interação possíveis, como o de questão-resposta, mas mostrou-se particularmente deficiente no modo de acompanhamento. De fato, a dificuldade encontra-se em ser capaz ao mesmo tempo de gerar frases musicais coerentes a um determinado estilo, e de adaptar-se ao que está sendo tocado pelos outros músicos. A Figura 1 ilustra esse problema no caso de um

sistema destinado a acompanhar uma melodia cantada. Os acordes escolhidos para realizar o acompanhamento devem manter uma continuidade, mas também se encaixarem à melodia que está sendo tocada. No caso de um sistema em tempo real, uma dificuldade a mais é encontrada: não se sabe à priori o que vai ser cantado. Neste caso, o algoritmo deve ter a capacidade, além de adaptação e continuidade, de predição.

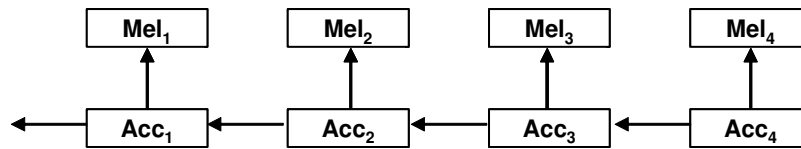


Figura 1 – relação de dependência no acompanhamento. Os acordes dependem não só dos acordes anteriores mas também da melodia que está sendo tocada.

Este trabalho propõe uma alteração no algoritmo de PI de forma que ele seja capaz de também levar em consideração a adaptação aos demais instrumentos. Com isso, acreditamos que o método será apropriado para ser utilizado em sistemas de acompanhamento musical. A próxima seção descreve o PI tradicional. A seção 3 apresenta os diversos compromissos possíveis entre adaptação, continuidade e predição apresentados por músicos no momento do acompanhamento. A seção 4 descreve as alterações possíveis no *parsing* incremental.

2. Parsing Incremental

Este algoritmo provém da fase de análise das técnicas de compressão Lempel-Ziv [Ziv and Lempel 1977]. Desde que a música seja interpretada como uma seqüência de notas, a técnica é aplicável. O algoritmo original é dividido em duas partes: primeiro, ele lê a seqüência de entrada e gera um modelo que captura a redundância. Em seguida, ele gera o código comprimido desta seqüência com relação ao modelo. Em um sistema interativo, a segunda parte é substituída por uma simulação estocástica do modelo a fim de produzir novas seqüências.

O PI lê a seqüência incrementalmente. A cada ciclo, ele escolhe o padrão mais curto que ainda não existe no dicionário de sub-padrões, e o adiciona. Uma representação otimizada é uma árvore de sufixos onde os contextos são invertidos (i.e. da folha à raiz), e as continuações são armazenadas como índices associados a cada nó. Para pesquisar uma seqüência, basta navegar na árvore de sufixos. A Figura 2 ilustra o funcionamento do algoritmo.

Suponhamos que queiramos construir o modelo relativo às seqüências [C,A,F,E] seguida por [C,C,F,G]. Primeiro, o algoritmo lê a subseqüência [C,A,F], cuja continuação tem índice 4. O ramo [C,A,F], onde cada elemento possui o valor 4 como continuação. Em seguida, é lida a subseqüência [C,A], cuja continuação tem índice 3. Finalmente a subseqüência [C] é lida, criando o último ramo (a árvore completa é a Figura 2a). Em seguida, uma nova seqüência, [C,C,F,G], é observada. O mesmo processo é feito. O ramo [C,C,F], por exemplo, é anexado ao ramo [C,A,F], que tem a mesma raiz. A árvore final é mostrada na Figura 2b. Este mecanismo faz com que apenas uma nova informação seja adicionada. Para procurar uma continuação, basta navegar na árvore, procurando a maior subseqüência. Por exemplo, para a seqüência [E,A,F], a resposta é a continuação de índice 4 (E), referente à subseqüência [A,F].

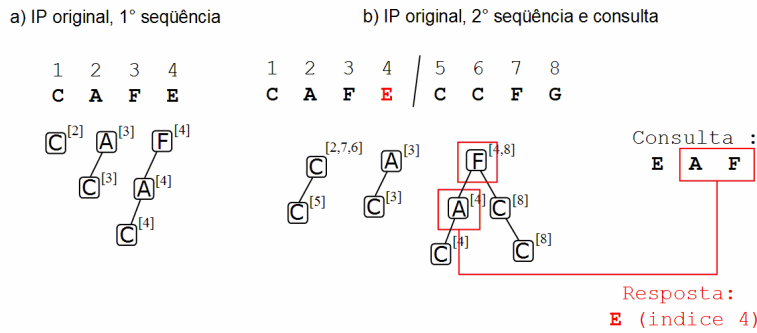


Figura 2 – árvores de sufixos geradas pelo IP para as seqüências [C,A,F,E] e [C,C,F,G], e consulta à mesma. Em vermelho, a resposta.

3. Acompanhamento Musical

É extremamente difícil criar um sistema de acompanhamento musical interativo, em tempo real, sobretudo se a música não é conhecida à priori. Nós pretendemos modelar e comparar 3 comportamentos diferentes do acompanhador que nós julgamos particularmente pertinentes, combinando de maneiras diferentes predição, adaptação e continuidade.

1. Predição/Recuperação – de acordo com a melodia que foi tocada, prevê-se uma continuação. Em seguida, procura-se um acompanhamento para esta continuação. A continuidade é mantida à medida que a melodia segue o caminho previsto.
2. Recuperação/Predição – de acordo com a melodia que foi tocada, procura-se um acompanhamento. Em seguida, procura-se uma continuação para este acompanhamento. A continuidade do acompanhamento é naturalmente respeitada, mas a adaptação à melodia não é mais garantida.
3. Conjunta – a melodia e o acompanhamento são vistos como uma única entidade, onde a predição, a adaptação e a continuidade são pensadas simultaneamente.

4. Parsing Incremental Alterado

Para cada um dos comportamentos acima, nós previmos um algoritmo alterado. Para o comportamento 1, nossa proposição é alterar a estrutura das árvores de sufixos (Figura 3). Ao invés de armazenar as possíveis continuações, seriam armazenados os possíveis acompanhamentos. Para o comportamento 2, nossa proposição é criar duas árvores de sufixos separadas, uma para a melodia, outra para o acompanhamento (Figura 4). Cada folha da árvore referente à melodia apontaria para um nó na árvore relativa aos acordes. Para o comportamento 3, nossa proposição é mixar os estados. Cada elemento da seqüência não seria mais uma única nota (da melodia), ou um único acorde (do acompanhamento), mas uma tupla <nota, acorde> (Figura 5).

Suponhamos que as seqüências [C,A,F,E] e [C,C,F,G] são acompanhadas pelos acordes [Am,%F,E7] e [Am,F,E7,%], onde “%” indica que o acorde é mantido. Neste momento é observada a seqüência [E,A,F], para a qual será procurado um acompanhamento.

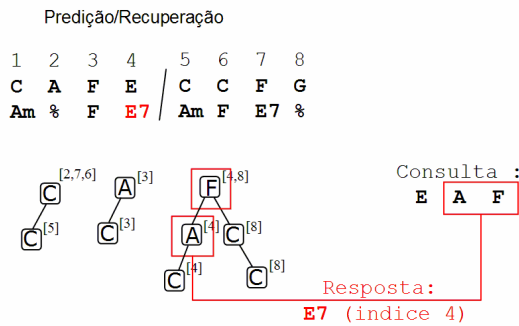


Figura 3 – exemplo de funcionamento com o algoritmo de predição/recuperação.

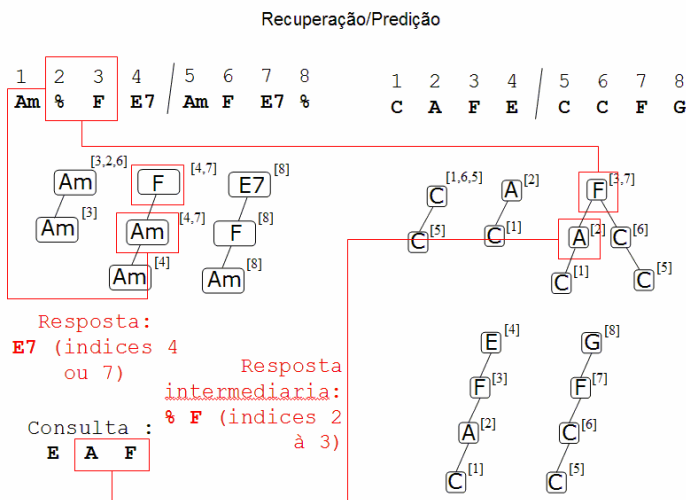


Figure 4 – exemplo de funcionamento com o algoritmo de recuperação/predição.

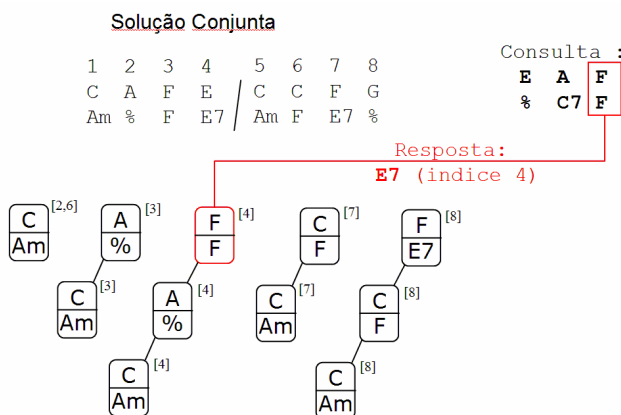


Figura 5 – exemplo de funcionamento com o algoritmo conjunto.

O caso 1 é ilustrado na Figura 3. A árvore é semelhante à anterior, apenas os índices referem-se desta vez aos acordes do acopanhamento, e não à continuação. A árvore é navegada a partir da última nota (F), encontrando-se então a maior subsequência. Neste caso: [A, F].

No caso 2, a maior subsequência de notas encontrada retorna uma subsequência de acordes (no exemplo, os acordes entre os índices 2 a 3). É procurada então a continuação desta seqüência de acordes (índices 4 ou 7), como visto na Figura 4.

No caso 3, os acordes que foram tocados também são considerados no momento da consulta. Digamos que os acordes previamente selecionados haviam sido [% ,C7,F], a consulta torna-se então [E|% ,A|C7,F|F] (Figura 5). Neste caso, a maior subsequência encontrada não é mais de tamanho 2 (como o [A,F] anterior), mas de tamanho 1 ([F|F]), o que mostra que o algoritmo tem maior dificuldade em encontrar soluções.

Tais algoritmos estão em vias de implementação. Esperamos em breve efetuar experimentos com músicos e amadores, de forma a avaliar quais os pontos fortes e fracos de cada um, investigar em que pontos o sistema realmente emula uma aptidão do músico, e jogar um pouco de luz sobre o tema.

Nós gostaríamos de agradecer à equipe do Sony Computer Science Lab em Paris pelo apoio.

Esta pesquisa é patrocinada por CAPES/COFECUB, Brasil/França.

References

O. Lartillot, S. Dubnov, G. Assayag and G. Bejerano. (2001) Automatic Modeling of Musical Style. In International Computer Music Conference, La Havana.

S. Dubnov, G. Assayag, and R. El-Yaniv. (1998) "Universal Classification Applied to Musical Sequences". In Proceedings of International Computer Music Conference, pp. 332-340.

Pachet. P. (2002) "The Continuator: Musical Interaction with Style". In ICMA, editor, Proceedings of ICMC, pp. 211-218.

Ziv J., Lempel A. (1977) "A Universal Algorithm for Sequential Data Compression", IEEE Transactions on Information Theory, Vol. 23, No. 3, pp. 337-343.