

Automatic Evaluation of Music Students from MIDI Data

Jessica Díaz-Estrada

Fac. de Ing. Eléctrica, Div. de Estudios de Postgrado
Universidad Michoacana de San Nicolás de Hidalgo.
Morelia, Mich. México
Email: jdiaz@dep.fie.umich.mx

Antonio Camarena-Ibarrola

Fac. de Ing. Eléctrica, Div. de Estudios de Postgrado
Universidad Michoacana de San Nicolás de Hidalgo.
Morelia, Mich. México
Email: camarena@umich.mx

Resumen—A virtual music teacher just as a human music teacher needs to evaluate his students to decide if they should go to the next lesson or not. In this work the student is evaluated from symbolic data, specifically from the information recorded in MIDI files which are normally produced by modern instruments such as music keyboards. Our approach for evaluating a music student consists of converting the MIDI file produced by the student to a long string with the sequence of music notes and durations. This string is compared to another long string that corresponds to a “correct” version of the same music which has been generated by a trained musician. The more similar the string generated by the student to the string generated by the trained musician is the higher the grade. In our experiments 13 music students were evaluated and their grades compared to those assigned by a music teacher. According to our tests the Transposition invariant InDel distance proved to be the most adequate. The Branch and Bound algorithm was used to deal with the fact that this distance requires more time to be computed. The grades of the students are quite similar to those assigned by the teacher so the system can reliably and impartially be used to evaluate music students.

I. INTRODUCCIÓN

Una herramienta capaz de automatizar la evaluación de estudiantes de música sería de gran utilidad para las escuelas de música proporcionando resultados confiables e imparciales. La evaluación automática sería un importante módulo para poder implementar un maestro virtual de música, un software que se pueda instalar en cualquier computadora y que habilitaría la masificación de la enseñanza de la música. Percival [1] presentó un estudio de trabajos recientes de enseñanza asistida por computadora de instrumentos musicales y afirma que la música es el área que más necesita la ayuda de la enseñanza asistida por computadora pues se requiere de mucha práctica diaria y de motivar a los alumnos a la práctica a través de juegos y multimedia, afirma también que se debe de proporcionar un análisis objetivo del rendimiento de los estudiantes. Muchos proyectos existentes intentan reemplazar a los maestros humanos, proporcionando lecciones durante la práctica diaria.

I-A. Trabajo previo

Cano *et al* utilizó Modelos Ocultos de Markov (HMM) para evaluar la similitud entre dos interpretaciones musicales [2], Raphael [3] usó HMM para segmentar señales musicales. En 2003, Turetsky y Ellis [4], proponen un sistema

de alineamiento, mediante el cálculo de una matriz de costos entre el espectrograma de la señal original y el de una señal sintética obtenida a partir de un archivo MIDI. En 2005 Dixon [5] desarrolló un sistema de seguimiento de interpretaciones musicales en vivo utilizando Doblado Dinámico de Tiempo (DTW por sus siglas en inglés). En 2006, Cont [6] desarrolló un sistema de alineamiento de audio a partitura de tiempo real para instrumentos polifónicos utilizando NMF (del inglés Non-negative Matrix Factorization) y un HMM jerárquico. En 2015 Camarena y Morales [7] desarrollaron un método para la evaluación automática de los estudiantes de música, el estudiante interpreta una pieza de música específica frente de un micrófono conectado a una computadora que realiza la evaluación. El estudiante selecciona la pieza que va a interpretar entre una lista de piezas musicales conocidas por el sistema llamadas referencia base, la señal de audio es convertida en una secuencia de vectores croma, posteriormente se alinean la interpretación del estudiante y la referencia base para determinar que tan parecidas son, usan técnicas de alineamiento de series de tiempo como DTW.

I-B. Representación simbólica moderna de la música

Una pieza musical se puede representar mediante el uso de datos simbólicos como los que se encuentran en los archivos MIDI [8] o en las partituras de una pieza musical. También se puede por supuesto trabajar con señales de audio. Por otro lado, podemos clasificar las piezas musicales como monofónicas o polifónicas. La música monofónica es la de textura más sencilla y consiste en una sola línea melódica sin acompañamiento alguno mientras que la polifónica es un tipo de textura musical en la que suenan simultáneamente múltiples voces melódicas. El presente trabajo de evaluación de estudiantes de música se centra en el procesamiento de datos simbólicos tanto monofónicos como polifónicos como se muestra en la Tabla I.

II. EXTRACCIÓN DE DATOS SIMBÓLICOS

La información contenida en un archivo MIDI esta organizada de acuerdo a la Tabla II.

Cada mensaje está producido por una acción realizada en el teclado musical. Ejemplo: un mensaje *Note on* cuando se pulsa la tecla musical, un mensaje *Note off* cuando se libera la tecla

Tabla I
CLASIFICACIÓN DE LA REPRESENTACIÓN COMPUTACIONAL DE LOS
DATOS MUSICALES.

| | | Pieza Musical Interpretada por el Estudiante | | | |
|--|------------|--|--|-------------|-------------|
| | | Simbólica | | Acústica | |
| | | Mono-fónica | Poli-fónica | Mono-fónica | Poli-fónica |
| Pieza Musical Referencia Base Correcta | Sim-bólica | Monofónica | El presente trabajo se centra en este problema | D | |
| | | Polifónica | | | |
| | Acús-tica | Monofónica | B | C | |
| | | Polifónica | | | |

Tabla II
ESTRUCTURA DEL ARCHIVO MIDI.

| Información de cabecera del archivo | |
|-------------------------------------|---|
| Número de pistas | |
| | Pista 1: Contiene número de eventos |
| | Mensaje 1: Nota, Velocidad, Nú.de Ticks |
| | Mensaje 2: Nota, Velocidad, Nú.de Ticks |
| | Mensaje n: Nota, Velocidad, Nú.de Ticks |
| | Pista 2: Contiene número de eventos |
| | Mensaje 1: Nota, Velocidad, Nú.de Ticks |
| | Mensaje 2: Nota, Velocidad, Nú.de Ticks |
| | Mensaje n: Nota, Velocidad, Nú.de Ticks |
| | Pista n: Contiene número de eventos |
| | Mensaje 1: Nota, Velocidad, Nú.de Ticks |
| | Mensaje 2: Nota, Velocidad, Nú.de Ticks |
| | Mensaje n: Nota, Velocidad, Nú.de Ticks |

[9]. Los mensajes contenidos en el MIDI almacenan información como son los parámetros: Nota, Velocidad, Número de Ticks. La *nota* y la *velocidad* en un archivo MIDI son cuantificados en valores entre 0 a 127 y el *número de ticks* indica la duración real que una tecla estuvo presionada.

III. TÉCNICAS DE COMPARACIÓN

Nosotros hacemos uso de técnicas de comparación del área de procesamiento de texto [10] implementadas mediante programación dinámica.

III-A. Distancia de Levenshtein

La distancia de Levenshtein [11] llamada también distancia de edición indica el número mínimo de operaciones de edición (*inserciones, eliminaciones, y sustituciones*) que transforman una cadena de caracteres en otra. La recurrencia (1) calcula la distancia de Levenshtein entre las cadenas A y B de longitudes N y M respectivamente.

$$L_{i,0} = i, \quad \forall 0 \leq i \leq N$$

$$L_{0,j} = j, \quad \forall 0 \leq j \leq M$$

$$L_{i,j} = \begin{cases} L_{i-1,j-1} & \text{Si } A_i = B_j, \\ 1 + \min(L_{i-1,j-1}, L_{i-1,j}, L_{i,j-1}) & \text{Si } A_i \neq B_j \end{cases} \quad (1)$$

III-B. Distancia de la Subsecuencia Comun Mas Larga

La distancia LCS conocida también como distancia Indel [12] es similar a distancia de Levenshtein excepto que no permite la operación de *sustitución* [13]. La distancia LCS entre dos cadenas es mas pequeña mientras mas larga sea alguna secuencia de símbolos presente en ambas cadenas lo cual nos parece adecuado para señales de voz o de música. Por ejemplo la distancia de Levenshtein indica que las cadenas *campamento* y *cooompuuutaadoora* están a la misma distancia de la cadena *computadora*. En cambio la distancia LCS si es capaz de ver la diferencia, vea la Tabla III. Para el cálculo de la distancia LCS usamos la recurrencia (2).

$$C_{i,0} = i, \quad \forall 0 \leq i \leq N$$

$$C_{0,j} = j, \quad \forall 0 \leq j \leq M$$

$$C_{i,j} = \begin{cases} C_{i-1,j-1} & \text{Si } A_i = B_j, \\ 1 + \min(C_{i-1,j}, C_{i,j-1}) & \text{Si } A_i \neq B_j \end{cases} \quad (2)$$

III-C. Distancia TWLCS

Para calcular una medida de similitud entre dos cadenas de caracteres tomando en cuenta los errores que implican las distorsiones rítmicas Grabowski *et al* propuso la medida de similitud TWLCS (Time Warped Longest Common Subsequence) [14]. Nosotros realizamos una modificación para convertir dicha medida de similitud en una de distancia y así poderla utilizar en nuestro sistema, utilizamos la recurrencia (3) para calcular dicha distancia. Cabe mencionar que esta modificación no altera la complejidad en tiempo que implica el cálculo de la similitud que es $O(MN)$. Observe en la Tabla III la distancia TWLCS entre las cadena *computadora* y la cadenas *cooompuuutaadoora* que es una versión alargada de la misma es cero.

$$T_{i,0} = i, \quad \forall 0 \leq i \leq N$$

$$T_{0,j} = j, \quad \forall 0 \leq j \leq M$$

$$T_{i,j} = \begin{cases} \min(T_{i,j-1}, T_{i-1,j}, T_{i-1,j-1}) & \text{Si } A_i = B_j \\ 1 + \min(T_{i,j-1}, T_{i-1,j}) & \text{Si } A_i \neq B_j \end{cases} \quad (3)$$

Tabla III
EJEMPLO COMPARACIÓN DE DISTANCIAS

| A | B | Lev | LCS | TWLCS |
|-------------|-------------------|-----|-----|-------|
| computadora | cooompuuutaadoora | 8 | 8 | 0 |
| computadora | campamento | 8 | 11 | 11 |

III-D. Distancia Invariante a Transposición LCS

Una característica particular en el análisis de la música es la transposición de notas musicales. La misma melodía es percibida incluso si el tono es modificado. La transposición musical consiste en que una pieza se traslada de una

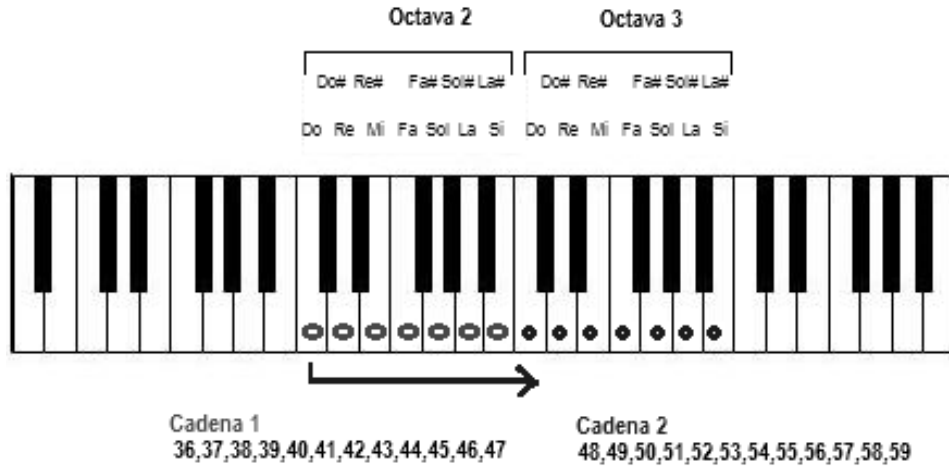


Figura 1. Notas desplazadas en una Interpretación

tonalidad a otra. Suele ocurrir que una melodía está correctamente ejecutada, pero ha sido interpretada con un desplazamiento de notas. La distancia invariante a transposición [15] busca hacer coincidir dos cadenas cuando todos los caracteres de una de ellas pueden estar desplazados un cierto t desconocido. En la Figura 1 se muestra la cadena 1 correspondiente a las teclas con círculos azules pero que son interpretadas en la siguiente octava correspondiente a teclas con círculos rojos. La cadena 1 es la secuencia (36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47) y la cadena 2 es la secuencia (48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59) y el desplazamiento es $t=12$.

Sea Σ un alfabeto finito de símbolos. Por simplicidad se considera $\Sigma = 0, \dots, \sigma$. Se define a la subsecuencia común mas larga invariante a transposición (LCTS por sus siglas en inglés) entre las cadenas A y B de longitudes N y M respectivamente como la menor distancia LCS entre $A + t$ y B donde $-\sigma \leq t \leq \sigma$, como lo indica la Ecuación 4. Observe que $A + t = (a_1 + t)(a_2 + t) \dots (a_N + t)$.

$$LCTS(A, B) = \min_{t \in [-\sigma, \sigma] \mathbb{Z}} LCS(A + t, B) \quad (4)$$

III-E. Algoritmo Branch and Bound

Determinar la distancia LCTS de manera directa implica un gran número de operaciones y una carga considerable para la computadora, para calcular esta distancia de manera mas rápida se puede utilizar el Algoritmo Branch and Bound [13], este algoritmo encuentra el valor máximo $LCS^t(A, B)$ ($t \in X$), para ello construye un árbol binario cuyos nodos tienen la forma $[\tau, \tau']$ que representa el rango de transposiciones $X = \tau \dots \tau'$. La raíz es $[-\sigma, \sigma]$, cada nodo interno $[\tau, \tau']$ tiene dos hijos $[\tau, \theta]$ y $[\theta + 1, \tau']$. Donde $\theta \leftarrow \lfloor \frac{(\tau + \tau')}{2} \rfloor$. Para cada nodo $[\tau, \tau']$ del árbol se calcula $LCS^{[\tau, \tau']}(A, B)$, el resultado es la cota superior de $LCS^t(A, B)$ para cualquier $\tau \leq t \leq \tau'$. Para calcular $LCS^{[\tau, \tau']}(A, B)$ se usa la recurrencia (5). El Algoritmo 1 determina $LCTS(A, B)$ mediante Brach and Bound. La función $SetLCS(A, B, \tau, \tau')$ calcula

$LCS^{[\tau, \tau']}(A, B)$ como se describe en la recurrencia 5. El algoritmo mantiene una cola de prioridad con elementos de la forma $([\tau, \tau'], val)$ (val es el valor de $LCTS$), la función $ExtractMax()$ se utiliza para extraer en orden los elementos de dicha cola de prioridad. La función $Insert()$ inserta elementos en dicha cola de prioridad. Observe que al final del Algoritmo 1, se usa la fórmula para convertir la longitud común mas larga a distancia.

Algorithm 1 Algoritmo Branch and Bound LCTS

procedure BRANCH AND BOUND LCTS(A, B, σ)

Init(Q)

$[\tau, \tau'] \leftarrow [-\sigma, \sigma]$

while $\tau \neq \tau'$ **do**

$\theta \leftarrow \lfloor \frac{(\tau + \tau')}{2} \rfloor$

$Insert(Q, ([\tau, \theta], SetLCS(A, B, \tau, \theta)))$

$Insert(Q, ([\theta + 1, \tau'], SetLCS(A, B, \theta + 1, \tau')))$

$([\tau, \tau'], lcts) \leftarrow ExtractMax(Q)$

Convertir *distancia* $\leftarrow A.length + B.length - (2 * lcts)$

return *distancia*

$$C_{i,0}^{[\tau, \tau']} = i \quad , \forall 0 \leq i \leq N$$

$$C_{0,j}^{[\tau, \tau']} = j \quad , \forall 0 \leq j \leq M$$

$$C_{i,j}^{[\tau, \tau']} \begin{cases} C_{i-1,j-1}^{[\tau, \tau']} & \text{Si } \tau \leq b_j - a_i \leq \tau' \\ 1 + \min(C_{i-1,j}^{[\tau, \tau']}, C_{i,j-1}^{[\tau, \tau']}) & \text{Sino} \end{cases} \quad (5)$$

En la Figura 2 se muestra el árbol binario que genera el algoritmo Branch and Bound para calcular LCTS entre las cadenas:

11, 30, 21, 12, 13, 44, 33, 34, 39, 21, 6, 1, 36, 2, 47, 45, 23, 40, 24, 17

8, 24, 46, 44, 16, 39, 12, 18, 10, 41, 23, 42, 17, 3, 44, 20, 41, 35, 7, 37

Suponiendo que el algoritmo recibe el rango de transposiciones inicial de $[-50, 50]$. La longitud (LCTS) que obtiene el algoritmo para este ejemplo es de **8**, al convertir dicha longitud

a distancia como en el Algoriimo 1 obtenemos un resultado de 24.

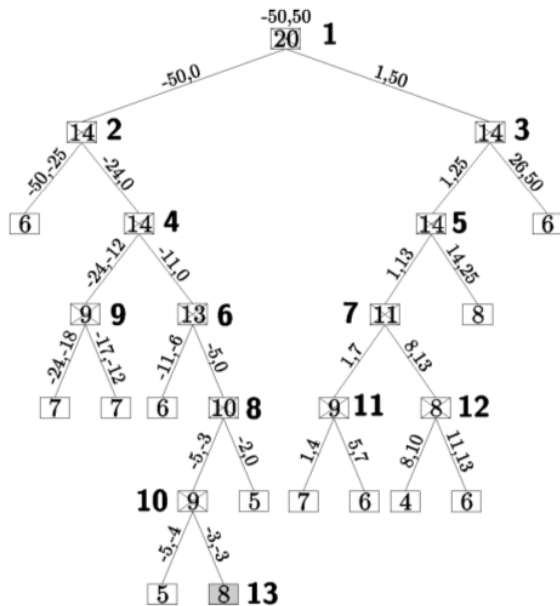


Figura 2. Ejemplo de Árbol Binario con Solución. [13]

IV. DESCRIPCIÓN DEL SISTEMA DE EVALUACIÓN AUTOMÁTICA

Para evaluar a un estudiante de música, el alumno toca una melodía con un teclado musical conectado a una computadora a través del puerto MIDI. El alumno selecciona en la computadora la melodía que va a tocar, luego realiza su ejecución y cuando termina el sistema le muestra su calificación. El sistema tiene dos módulos, el módulo de extracción de datos simbólicos del archivo de audio y el módulo de evaluación.

IV-A. Módulo Extracción de Datos Simbólicos

Este módulo extrae la información del archivo MIDI generado por el estudiante, de este archivo extraemos el tono, la velocidad, y el número de ticks por cada tecla que el estudiante pulsó en el teclado. Para extraer los datos del archivo MIDI utilizamos el paquete de clases de java *javax.sound.midi*

IV-A1. Obtención de las Notas Musicales: Para determinar la nota musical correspondiente a la tecla pulsada en el teclado leemos en el mensaje MIDI el valor del tono que en el archivo MIDI es un número entero entre 0 a 127, utilizamos la Tabla IV para averiguar la nota musical y la octava correspondiente.

IV-A2. Determinación de las duraciones de las Notas Musicales: Las duraciones de las teclas presionadas por el estudiante se leen del archivo MIDI en el parámetro *número de ticks*, para ello es necesario leer los mensajes *Note on* y *Note Off* correspondiente a una misma nota para saber el número de ticks al inicio de la nota y el número de ticks de final de nota

Tabla IV
VALORES DE LOS TONOS CORRESPONDIENTES A CADA NOTA [9].

| 8va | Do | Do# | Re | Re# | Mi | Fa | Fa# | Sol | Sol# | La | La# | Si |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|-----|-----|-----|
| 0 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 1 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| 2 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 |
| 3 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 |
| 4 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 |
| 5 | 60 | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 |
| 6 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 | 81 | 82 | 83 |
| 7 | 84 | 85 | 86 | 87 | 88 | 89 | 90 | 91 | 92 | 93 | 94 | 95 |
| 8 | 96 | 97 | 98 | 99 | 100 | 101 | 102 | 103 | 104 | 105 | 106 | 107 |
| 9 | 108 | 109 | 110 | 111 | 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 |
| 10 | 120 | 121 | 122 | 123 | 124 | 125 | 126 | 127 | | | | |

respectivamente. Con la Ecuación 6 obtenemos la duración dR de la nota musical en segundos.

$$dR = \frac{I - F}{r} \quad (6)$$

donde I es el inicio de nota en ticks, F tiene el momento final de nota en ticks y r es la resolución en ticks por segundo.

IV-A3. Conversión de duraciones reales a duraciones estándar de notas musicales: Es conveniente realizar una conversión de la duración en segundos de las notas a duraciones musicales al estilo de las partituras. Recordando que en la música, hay notas que tienen duración de 1 tiempo (nota blanca), de 1/2 (nota negra), de 1/4 etc.. La duración dM en cuartos de nota la obtenemos mediante (7) donde dR es la duración de la nota en segundos y te es el *tempo* en microsegundos por cuarto de nota el cual se lee en el encabezado del archivo MIDI. En la Tabla V se muestran los valores que utilizamos en las cadenas que generan los estudiantes al tocar correspondientes a las duraciones en cuartos de nota.

$$dM = \left\lceil \frac{dR * 1000000}{te} \right\rceil \quad (7)$$

Tabla V
VALORES EN TIEMPOS MUSICALES ESTILO PARTITURAS.

| Valor asignado a la cadena | Significado en tiempos musicales |
|----------------------------|----------------------------------|
| 5 | Más de 1 tiempo |
| 4 | 4/4 de nota |
| 3 | 3/4 de nota |
| 2 | 2/4 de nota |
| 1 | 1/4 de nota |

IV-A4. Cuantización de Velocidades de las Notas Musicales: La velocidad correspondiente a cada nota musical, es reflejada como *volumen* ante el oído humano [9]. El parámetro de velocidad de la nota interpretada en un archivo MIDI se cuantifica con valores de 0 a 127 ordenados de menor a mayor intensidad, este valor depende de la fuerza con la que se pulse la tecla musical. Aun si la interpretación es realizada por la misma persona con la misma intención es muy poco probable que se presionen las teclas exactamente con la misma intensidad. Por esa razón, decidimos dividir la velocidad en 6 rangos mostrados en la Tabla VI, en la columna 3 se describe el nombre de cada rango.

Tabla VI
CUANTIZACIÓN DEL PARÁMETRO DE VELOCIDAD.

| Rango de velocidad en el mensaje MIDI | Valor asignado a la cadena | Nombre | Ejecución |
|---------------------------------------|----------------------------|-------------|--------------|
| 0 | 0 No presión tecla musical | - | - |
| 1 - 22 | 1 | pianísimo | Muy suave |
| 23 - 43 | 2 | piano | Suave |
| 44 - 64 | 3 | mezzo piano | No tan suave |
| 65 - 85 | 4 | mezzo forte | Poco fuerte |
| 86 - 106 | 5 | forte | Fuerte |
| 107 - 127 | 6 | fortísimo | Muy fuerte |

IV-B. Módulo Evaluación

Para evaluar a un estudiante comparamos la cadena de caracteres extraída de la interpretación del estudiante con otra cadena considerada correcta. Para establecer la similitud de las cadenas utilizamos las técnicas de comparación expuestas anteriormente, estas comparan las cadenas y dan una medida de distancia entre ellas. Con el fin de convertir las distancias a calificaciones de 0 a 10 usamos la Ecuación (8) donde *distancia* es obtenida mediante la técnica de comparación y *x* es un parámetro de valor 0.1 para melodía monofónicas y de 0.044 para polifónicas, estos valores fueron obtenidos empíricamente.

$$\text{calificacion} = 10 - (\text{distancia} * x) \quad (8)$$

V. EXPERIMENTOS Y RESULTADOS

Nuestro conjunto experimental consta de 26 actuaciones musicales grabadas en un *Estudio de Grabación Musical* en Morelia Michoacán. Contando con 13 estudiantes de piano tocando una melodía monofónica y una melodía polifónica cada uno. Adicionalmente un profesor de música interpretó las melodías consideradas como correctas (referencia base), el profesor también proporcionó una evaluación de cada estudiante. En las tablas VII y VIII se comparan las calificaciones de los estudiantes obtenidas con nuestro sistema con las calificaciones del profesor para la pieza musical monofónica y para la pieza polifónica respectivamente. Las columnas etiquetadas C_L , C_{LCS} , C_{ITLCS} y C_{TWLCS} muestran las calificaciones arrojadas por el sistema utilizando la distancia de Levenshtein, InDel, InDel invariante a transposición y TWLCS respectivamente. La columna C_P muestra las calificaciones asignadas por el profesor para cada uno de los 13 estudiantes, la última columna muestra la diferencia entre la calificación del profesor y la mas cercana de las evaluaciones automáticas las cuales se muestran con negrita en las tablas.

Considerando que las calificaciones asignadas por el profesor son correctas y para decidir cual técnica de alineamiento o medida de distancia entre cadenas es mas adecuada, evaluamos al sistema evaluador de tres maneras; la primera consistió en calcular la distancia de Manhattan entre cada columna de la Tabla VII correspondiente a cada técnica de alineamiento (Columnas 2 a 5) y la columna 6 correspondiente a la calificación del profesor. A esta forma la podemos denominar *Norma L1 de la diferencia*; La segunda manera fue como

Tabla VII
COMPARACIÓN DE EVALUACIONES AUTOMÁTICAS Y LAS ASIGNADAS POR EL PROFESOR PARA MÚSICA MONOFÓNICA

| Eval | C_L | C_{LCS} | C_{ITLCS} | C_{TWLCS} | C_P | Min. Dif. |
|------|-----------|------------|-------------|-------------|-------|-----------|
| 1 | 9.3 | 8.6 | 8.6 | 8.6 | 8 | 0.6 |
| 2 | 8.1 | 6.4 | 6.4 | 6.6 | 6 | 0.4 |
| 3 | 8.6 | 7.2 | 7.2 | 7.2 | 7 | 0.2 |
| 4 | 8.3 | 6.6 | 6.6 | 6.6 | 7 | 0.4 |
| 5 | 7.4 | 5.6 | 4.6 | 5.7 | 5 | 0.4 |
| 6 | 0 | 0 | 9.2 | 0 | 8 | 1.2 |
| 7 | 2 | 0.6 | 9.6 | 0.6 | 9 | 0.6 |
| 8 | 8.8 | 8.7 | 6.3 | 8.7 | 7 | 0.7 |
| 9 | 8.6 | 8 | 7.2 | 8 | 8 | 0 |
| 10 | 2 | 0.6 | 9.6 | 0.6 | 9 | 0.6 |
| 11 | 6.5 | 5.6 | 5.6 | 5.9 | 5 | 0.6 |
| 12 | 6 | 3.1 | 0.9 | 3.2 | 2 | 1.1 |
| 13 | 10 | 10 | 10 | 10 | 10 | 0 |

Tabla VIII
COMPARACIÓN DE EVALUACIONES AUTOMÁTICAS Y LAS ASIGNADAS POR EL PROFESOR PARA MÚSICA POLIFÓNICA

| Eval | C_L | C_{LCS} | C_{ITLCS} | C_{TWLCS} | C_P | Min. Dif. |
|------|-----------|-------------|-------------|-------------|-------|-----------|
| 1 | 9.30 | 8.99 | 8.64 | 8.99 | 8 | 0.64 |
| 2 | 8.11 | 7.76 | 5.91 | 7.76 | 7 | 0.76 |
| 3 | 9.25 | 8.99 | 8.55 | 8.99 | 8 | 0.55 |
| 4 | 9.69 | 9.38 | 9.38 | 9.38 | 9 | 0.38 |
| 5 | 7.80 | 6.66 | 5.25 | 6.66 | 6 | 0.66 |
| 6 | 10 | 10 | 10 | 10 | 10 | 0 |
| 7 | 4.59 | 2.34 | 0 | 2.34 | 3 | 0.66 |
| 8 | 7.58 | 6.92 | 5.07 | 6.92 | 6 | 0.92 |
| 9 | 8.68 | 8.46 | 6.17 | 8.46 | 7 | 0.83 |
| 10 | 0 | 0 | 9.74 | 0 | 9 | 0.74 |
| 11 | 2.34 | 0 | 9.91 | 0 | 10 | 0.09 |
| 12 | 8.90 | 7.98 | 7.45 | 7.98 | 8 | 0.02 |
| 13 | 6.04 | 3.80 | 0 | 2.34 | 4 | 0.2 |

la primera pero usando la distancia Euclidiana o *Norma L2 de la diferencia*; La tercera manera consistió en ver a las calificaciones no como vectores sino como series de tiempo, así entonces calculamos el coeficiente de correlación entre las series de calificaciones arrojadas por el sistema para cada técnica y las series de calificaciones entregadas por el profesor. El resultado de esta comparación se muestra en la Tabla IX para las pruebas con la pieza musical monofónica. En la Tabla X se muestran los resultados de la comparación con la pieza musical polifónica. Las tres medidas de evaluación del evaluador nos permiten ver que la distancia LCS invariante a transposición es la técnica de alineamiento mas adecuada tanto para música monofónica como para música polifónica.

Tabla IX
COMPARACIÓN ENTRE LAS CALIFICACIONES ARROJADAS POR EL SISTEMA Y LAS ASIGNADAS POR EL PROFESOR DE ACUERDO A LAS PRUEBAS CON LA PIEZA MUSICAL MONOFÓNICA

| | Inv_Trans_LCS | LCS | TWLCS | Levenshtein |
|-------------|---------------|-------|-------|-------------|
| Norma L1 | 7.6 | 30.4 | 31.1 | 38.6 |
| Norma L2 | 2.3 | 14.51 | 14.54 | 16.14 |
| Correlación | 0.988 | 0.561 | 0.557 | 0.567 |

VI. CONCLUSIONES Y TRABAJOS FUTUROS

La distancia invariante a transposición LCS obtuvo calificaciones muy apegadas a las asignadas por el profesor de música y si usamos la metodología Branch and Bound la comparación

Tabla X

COMPARACIÓN ENTRE LAS CALIFICACIONES ARROJADAS POR EL SISTEMA Y LAS ASIGNADAS POR EL PROFESOR DE ACUERDO A LAS PRUEBAS CON LA PIEZA MUSICAL POLIFÓNICA

| | Inv_Trans_LCS | LCS | TWLCS | Levenshtein |
|-------------|---------------|--------|--------|-------------|
| Norma L1 | 13.544 | 26.036 | 27.488 | 30.592 |
| Norma L2 | 5.47 | 13.69 | 13.79 | 15.67 |
| Correlación | 0.954 | 0.637 | 0.641 | 0.668 |

se realiza de forma rápida. El promedio de las diferencias entre las evaluaciones y la mejor de las evaluaciones automáticas es de apenas medio punto en la escala de 0 a 10 tanto para las pruebas con la pieza monofónica como en las pruebas con la pieza polifónica. Como trabajo futuro nos planteamos utilizar Modelos Ocultos de Markov discretos para implementar otro sistema de evaluación automática y así comparar el desempeño de tal sistema con el sistema expuesto en este trabajo.

REFERENCIAS

- [1] G. Percival, Y. Wang, and G. Tzanetakis, "Effective use of multimedia for computer-assisted musical instrument tutoring," in *Proceedings of the International Workshop on Educational Multimedia and Multimedia Education*, ser. Emme '07. New York, NY, USA: ACM, 2007, pp. 67–76. [Online]. Available: <http://doi.acm.org/10.1145/1290144.1290156>
- [2] P. Cano, A. Loscos, and J. Bonada, "Score-performance matching using hmms," *Proceedings ICMC99*, pp. 441–444, 1999.
- [3] C. Raphael, "Automatic segmentation of acoustic musical signals using hidden markov models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 21, no. 4, pp. 360–370, 1999.
- [4] R. J. Turetsky and D. P. Ellis, "Ground-truth transcriptions of real music from force-aligned midi syntheses," in *Fourth International Symposium on Music Information Retrieval (ISMIR)*, 2003.
- [5] S. Dixon, "Live tracking of musical performances using on-line time warping," *Proc of the 8th Int Conf on Digital Audio Effects (DAFx'05)*, september 2005.
- [6] A. Cont, "Realtime audio to score alignment for polyphonic music instruments, using sparse non-negative constraints and hierarchical hmms," in *IEEE International Conference on Acoustics Speech and Signal Processing Proceedings (ICASSP)*, vol. 5. IEEE, 2006, pp. 245–248.
- [7] A. Camarena-Ibarrola and S. Morales-Pintor, "Automatic evaluation of music students," in *Mexican International Conference on Artificial Intelligence*. Springer, 2015, pp. 220–231.
- [8] J. Chilitos Valenzuela, *Audio Digital, Conceptos Básicos y Aplicaciones*. Miller Freeman Books, 1996.
- [9] A. Molina and J. R. Hidalgo, *Conversor WAVE a MIDI en tiempo real para guitarras eléctricas*. Reverte, 2006.
- [10] G. Navarro and M. Raffinot, *Flexible Pattern Matching in Strings. Practical On-Line Search for Texts and Biological Sequences*. Cambridge University Press, 2002, vol. 17.
- [11] Z. Bar-Yossef, T. Jayram, R. Krauthgamer, and R. Kumar, "Approximating edit distance efficiently," in *Foundations of Computer Science, 2004. Proceedings. 45th Annual IEEE Symposium on*. IEEE, 2004, pp. 550–559.
- [12] G. Navarro, "A guided tour to approximate string matching," *ACM Computing Surveys*, vol. 33, no. 1, pp. 31–88, 2001.
- [13] K. Lemström, G. Navarro, and Y. Pinzon, "Practical algorithms for transposition-invariant string-matching," *Journal of Technology*, no. 2, pp. 267–292, 2004.
- [14] A. Guo and H. Siegelmann, "Time warped longest common subsequence algorithm for music retrieval," in *5th International Conference on Music Information Retrieval (ISMIR)*, 2004.
- [15] S. Grabowski and G. Navarro, "O(mn log σ) time transposition invariant lcs computation," *Technical Report TR/DCC-2004-6*, pp. 1–29, 2004.