Learning Stochastic Edit Distance from Structured Data: Application in Music Retrieval

Jose Oncina[1], Marc Sebban[2], José Manuel Iñesta[1], Amaury Habrard[3], David Rizo[1] and Cristian Olivares[1]

[1]Universidad de Alicante – *Spain*
[2]Université Jean Monnet Saint-Étienne – *France*
[3]Université de Provence – *France*

January 29, 2008

Outline

1. Tree Edit Distance
2. Stochastic Extension
3. Music Database
4. Experiments
5. Future work

Outline

Outline

1. Tree Edit Distance
2. Stochastic Extension
3. Music Database
4. Experiments
5. Future work

Outline

Outline

Outline

1. Tree Edit Distance

2. Stochastic Extension

3. Music Database

4. Experiments

5. Future work

Tree Edit Distance

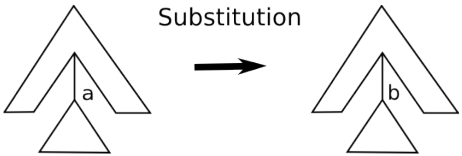
- The Tree Edit Distance is a generalization of the (String) Edit distance where the edit operations take place over trees.
- The tree edit operations are:
  - Substitution: Change the label of a tree node ($a \rightarrow b$)
  - Deletion: The children will become children of their father ($b \rightarrow \epsilon$)
  - Insertion: Some of the sibling will become children of the inserted node ($\epsilon \rightarrow b$)

Tree Edit Distance

- The Tree Edit Distance is a generalization of the (String) Edit distance where the edit operations take place over trees.
- The tree edit operations are:
  - Substitution: Change the label of a tree node ($a \longrightarrow b$)
  - Deletion: The children will become children of their father ($b \longrightarrow \epsilon$)
  - Insertion: Some of the sibling will become children of the inserted node ($\epsilon \longrightarrow b$)

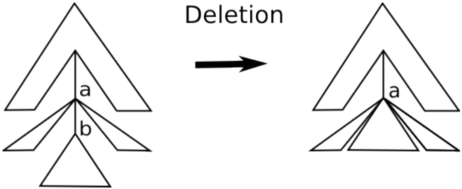
Tree Edit Distance

- The Tree Edit Distance is a generalization of the (String) Edit distance where the edit operations take place over trees.
- The tree edit operations are:
  - Substitution: Change the label of a tree node ($a \longrightarrow b$)
  - Deletion: The children will become children of their father ($b \longrightarrow \epsilon$)
  - Insertion: Some of the sibling will become children of the inserted node ($\epsilon \longrightarrow b$)



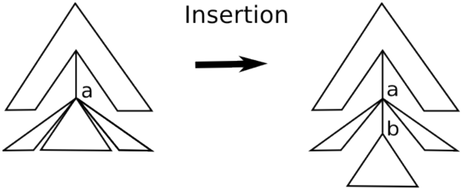
Tree Edit Distance

- The Tree Edit Distance is a generalization of the (String) Edit distance where the edit operations take place over trees.
- The tree edit operations are:
  - Substitution: Change the label of a tree node ($a \longrightarrow b$)
  - Deletion: The children will become children of their father ($b \longrightarrow \epsilon$)
  - Insertion: Some of the sibling will become children of the inserted node ($\epsilon \longrightarrow b$)



Tree Edit Distance

- The Tree Edit Distance is a generalization of the (String) Edit distance where the edit operations take place over trees.
- The tree edit operations are:
  - Substitution: Change the label of a tree node ($a \longrightarrow b$)
  - Deletion: The children will become children of their father ($b \longrightarrow \epsilon$)
  - Insertion: Some of the sibling will become children of the inserted node ($\epsilon \longrightarrow b$)



- Let S (script) be a sequence $s_1, \ldots, s_n$ of edit operations that transforms a tree into another
- Let γ be a weight function that assigns to each edit operation $(a \rightarrow b)$ a nonnegative real number $\gamma(a \rightarrow b)$.
- Let $\gamma(S) = \sum_{s_i \in S} \gamma(s_i)$

Definition:

The tree edit distance is a function such that:

$$\delta(t_1, t_2) = \min\{\gamma(S) | S \text{ is a script that transforms } t_1 \text{ into } t_2\}$$

- This can be computed [Zhang and Shasha, 89] in: $O(|t_1||t_2| \min(depth(t_1), leaves(t_1)) \min(depth(t_2), leaves(t_2)))$

The Edit Distance

- Let S (script) be a sequence $s_1, \ldots, s_n$ of edit operations that transforms a tree into another
- Let γ be a weight function that assigns to each edit operation $(a \rightarrow b)$ a nonnegative real number $\gamma(a \rightarrow b)$.
- Let $\gamma(S) = \sum_{s_i \in S} \gamma(s_i)$

Definition:

The tree edit distance is a function such that:

$$\delta(t_1, t_2) = \min\{\gamma(S) | S \text{ is a script that transforms } t_1 \text{ into } t_2\}$$

- This can be computed [Zhang and Shasha, 89] in: $O(|t_1||t_2| \min(depth(t_1), leaves(t_1)) \min(depth(t_2), leaves(t_2)))$

The Edit Distance

- Let S (script) be a sequence $s_1, \ldots, s_n$ of edit operations that transforms a tree into another
- Let γ be a weight function that assigns to each edit operation $(a \rightarrow b)$ a nonnegative real number $\gamma(a \rightarrow b)$.
- Let $\gamma(S) = \sum_{s_i \in S} \gamma(s_i)$

Definition:

The tree edit distance is a function such that:

$$\delta(t_1, t_2) = \min\{\gamma(S) | S \text{ is a script that transforms } t_1 \text{ into } t_2\}$$

- This can be computed [Zhang and Shasha, 89] in: $O(|t_1||t_2| \min(depth(t_1), leaves(t_1)) \min(depth(t_2), leaves(t_2)))$

The Edit Distance

- Let S (script) be a sequence $s_1, \ldots, s_n$ of edit operations that transforms a tree into another
- Let γ be a weight function that assigns to each edit operation $(a \rightarrow b)$ a nonnegative real number $\gamma(a \rightarrow b)$.
- Let $\gamma(S) = \sum_{s_i \in S} \gamma(s_i)$

Definition:

The tree edit distance is a function such that:

$$\delta(t_1, t_2) = \min\{\gamma(S) | S \text{ is a script that transforms } t_1 \text{ into } t_2\}$$

- This can be computed [Zhang and Shasha, 89] in: $O(|t_1||t_2| \min(depth(t_1), leaves(t_1)) \min(depth(t_2), leaves(t_2)))$

The Edit Distance

- Let S (script) be a sequence $s_1, \ldots, s_n$ of edit operations that transforms a tree into another
- Let γ be a weight function that assigns to each edit operation $(a \rightarrow b)$ a nonnegative real number $\gamma(a \rightarrow b)$.
- Let $\gamma(S) = \sum_{s_i \in S} \gamma(s_i)$

Definition:

The tree edit distance is a function such that:

$$\delta(t_1, t_2) = \min\{\gamma(S) | S \text{ is a script that transforms } t_1 \text{ into } t_2\}$$

- This can be computed [Zhang and Shasha, 89] in: $O(|t_1||t_2| \min(depth(t_1), leaves(t_1)) \min(depth(t_2), leaves(t_2)))$

Outline

The Stochastic Edit Distance

- The weights are probabilities
  - $\gamma(a \rightarrow b)$: substitution probability
  - $\gamma(\epsilon \rightarrow b)$: insertion probability
  - $\gamma(a \rightarrow \epsilon)$: deletion probability
  - γ: ending probability
- Now $\gamma(S) = \prod_{s_i \in S} \gamma(s_i) \cdot \gamma$

Definition:

The stochastic tree edit distance (δ) is a function such that:

$$p(t_1, t_2) = \sum_{S \in S(t_1 \rightarrow t_2)} \gamma(S) \qquad \delta(t_1, t_2) = -\log p(t_1, t_2)$$

where $S(t_1 \rightarrow t_2) = \{S | S \text{ is a script that transforms } t_1 \text{ into } t_2\}$

The Stochastic Edit Distance

- The weights are probabilities
  - $\gamma(a \rightarrow b)$: substitution probability
  - $\gamma(\epsilon \rightarrow b)$: insertion probability
  - $\gamma(a \rightarrow \epsilon)$: deletion probability
  - γ: ending probability
- Now $\gamma(S) = \prod_{s_i \in S} \gamma(s_i) \cdot \gamma$

Definition:

The stochastic tree edit distance (δ) is a function such that:

$$p(t_1, t_2) = \sum_{S \in S(t_1 \rightarrow t_2)} \gamma(S) \qquad \delta(t_1, t_2) = -\log p(t_1, t_2)$$

where $S(t_1 \rightarrow t_2) = \{S | S \text{ is a script that transforms } t_1 \text{ into } t_2\}$

The Stochastic Edit Distance

- The weights are probabilities
  - $\gamma(a \to b)$: substitution probability
  - $\gamma(\epsilon \to b)$: insertion probability
  - $\gamma(a \to \epsilon)$: deletion probability
  - γ: ending probability
- Now $\gamma(S) = \prod_{s_i \in S} \gamma(s_i) \cdot \gamma$

Definition:

The stochastic tree edit distance (δ) is a function such that:

$$p(t_1, t_2) = \sum_{S \in S(t_1 \to t_2)} \gamma(S) \qquad \delta(t_1, t_2) = -\log p(t_1, t_2)$$

where $S(t_1 \to t_2) = \{S | S \text{ is a script that transforms } t_1 \text{ into } t_2\}$

Normalization

Depending on the framework we have to ensure:

- In the generative model:

$$\sum_{t_1,t_2} p(t_1, t_2) = 1$$

$$\sum_{a,b} \gamma(a \to b) + \sum_{a} \gamma(a \to \epsilon) + \sum_{b} \gamma(\epsilon \to b) + \gamma = 1$$

- In the discriminative model:

$$\sum_{t_2} p(t_1, t_2) = 1 \quad \forall t_1$$

$$\sum_{b} \gamma(a \to b) + \sum_{b} \gamma(\epsilon \to b) + \gamma(a \to \epsilon) = 1 \quad \forall a$$

$$\sum_{b} \gamma(\epsilon \to b) + \gamma = 1$$

Normalization

Depending on the framework we have to ensure:

- In the generative model:

$$\sum_{t_1,t_2} p(t_1,t_2) = 1$$

$$\sum_{a,b} \gamma(a \to b) + \sum_{a} \gamma(a \to \epsilon) + \sum_{b} \gamma(\epsilon \to b) + \gamma = 1$$

- In the discriminative model:

$$\sum_{t_2} p(t_1,t_2) = 1 \quad \forall t_1$$

$$\sum_{b} \gamma(a \to b) + \sum_{b} \gamma(\epsilon \to b) + \gamma(a \to \epsilon) = 1 \quad \forall a$$

$$\sum_{b} \gamma(\epsilon \to b) + \gamma = 1$$

Normalization

Depending on the framework we have to ensure:

- In the generative model:

$$\sum_{t_1,t_2} p(t_1, t_2) = 1$$

$$\sum_{a,b} \gamma(a \to b) + \sum_{a} \gamma(a \to \epsilon) + \sum_{b} \gamma(\epsilon \to b) + \gamma = 1$$

- In the discriminative model:

$$\sum_{t_2} p(t_1, t_2) = 1 \quad \forall t_1$$

$$\sum_{b} \gamma(a \to b) + \sum_{b} \gamma(\epsilon \to b) + \gamma(a \to \epsilon) = 1 \quad \forall a$$

$$\sum_{b} \gamma(\epsilon \to b) + \gamma = 1$$

Normalization

Depending on the framework we have to ensure:

- In the generative model:

$$\sum_{t_1,t_2} p(t_1, t_2) = 1$$

$$\sum_{a,b} \gamma(a \to b) + \sum_{a} \gamma(a \to \epsilon) + \sum_{b} \gamma(\epsilon \to b) + \gamma = 1$$

- In the discriminative model:

$$\sum_{t_2} p(t_1, t_2) = 1 \quad \forall t_1$$

$$\sum_{b} \gamma(a \to b) + \sum_{b} \gamma(\epsilon \to b) + \gamma(a \to \epsilon) = 1 \quad \forall a$$

$$\sum_{b} \gamma(\epsilon \to b) + \gamma = 1$$

Which edition weights should we use?

- Usually an expert fixes them
- Why not learn them?
- In a probabilistic framework the immediate criterium is to search for the weights that maximize the expectation of a training set
- This was done for strings by:
  - Ristad & Yianilos in 1998 (generative model)
  - Oncina & Sebban in 2006 (discriminative model)
- In this proyect we extend those results to trees

Which edition weights should we use?

- Usually an expert fixes them
- Why not learn them?
- In a probabilistic framework the immediate criterium is to search for the weights that maximize the expectation of a training set
- This was done for strings by:
  - Ristad & Yianilos in 1998 (generative model)
  - Oncina & Sebban in 2006 (discriminative model)
- In this proyect we extend those results to trees

Which edition weights should we use?

- Usually an expert fixes them
- Why not learn them?
- In a probabilistic framework the immediate criterium is to search for the weights that maximize the expectation of a training set
- This was done for strings by:
  - Ristad & Yianilos in 1998 (generative model)
  - Oncina & Sebban in 2006 (discriminative model)
- In this proyect we extend those results to trees

Which edition weights should we use?

- Usually an expert fixes them
- Why not learn them?
- In a probabilistic framework the immediate criterium is to search for the weights that maximize the expectation of a training set
- This was done for strings by:
  - Ristad & Yianilos in 1998 (generative model)
  - Oncina & Sebban in 2006 (discriminative model)
- In this proyect we extend those results to trees

Which edition weights should we use?

- Usually an expert fixes them
- Why not learn them?
- In a probabilistic framework the immediate criterium is to search for the weights that maximize the expectation of a training set
- This was done for strings by:
  - Ristad & Yianilos in 1998 (generative model)
  - Oncina & Sebban in 2006 (discriminative model)
- In this proyect we extend those results to trees

EM Algorithm-based Approach

We adapt EM algorithm to the learning of $\delta(\cdot, \cdot)$ from (input,output) tree pairs.

- EM is suited for learning the parameters of a stochastic model.
- Function expectation counts the number of times each operation is used for changing an input tree into an output one
- Function maximization normalizes these counters to fulfill optimization constraints.

EM Algorithm-based Approach

We adapt EM algorithm to the learning of $\delta(\cdot,\cdot)$ from (input,output) tree pairs.

- EM is suited for learning the parameters of a stochastic model.
- Function expectation counts the number of times each operation is used for changing an input tree into an output one
- Function maximization normalizes these counters to fulfill optimization constraints.

EM Algorithm-based Approach

We adapt EM algorithm to the learning of $\delta(\cdot,\cdot)$ from (input,output) tree pairs.

- EM is suited for learning the parameters of a stochastic model.
- Function expectation counts the number of times each operation is used for changing an input tree into an output one
- Function maximization normalizes these counters to fulfill optimization constraints.

Outline

Application: Music Recognition

- The corpus consists of a set of monophonic 8-12 bar incipits of 20 worldwide well known tunes of different musical genres: (Bolero, Cucaracha, Jinglebells, Guantanamera, Happy Birthday, Yesterday, ...)
- For each song a canonic tune was created by writing the score in a musical notation application and exported to MIDI and MP3 format
- The MP3 files were given to some amateur and professional musicians who listened to each song (mainly to identify the requested range of the tune to be played) and played in MIDI keyboard and guitars several times the same tune with different embellishments. We colected 20 interpretations of each tune.
- They were given the guidelines:
  - to follow a meter and set it in the MIDI file
  - to establish the correct tonality in the MIDI file
  - to play only one note at a time (play in a monophonic way)

Application: Music Recognition

- The corpus consists of a set of monophonic 8-12 bar incipits of 20 worldwide well known tunes of different musical genres: (Bolero, Cucaracha, Jinglebells, Guantanamera, Happy Birthday, Yesterday, ...)
- For each song a canonic tune was created by writing the score in a musical notation application and exported to MIDI and MP3 format
- The MP3 files were given to some amateur and professional musicians who listened to each song (mainly to identify the requested range of the tune to be played) and played in MIDI keyboard and guitars several times the same tune with different embellishments. We colected 20 interpretations of each tune.
- They were given the guidelines:
  - to follow a meter and set it in the MIDI file
  - to establish the correct tonality in the MIDI file
  - to play only one note at a time (play in a monophonic way)

Application: Music Recognition

- The corpus consists of a set of monophonic 8-12 bar incipits of 20 worldwide well known tunes of different musical genres: (Bolero, Cucaracha, Jinglebells, Guantanamera, Happy Birthday, Yesterday, ...)
- For each song a canonic tune was created by writing the score in a musical notation application and exported to MIDI and MP3 format
- The MP3 files were given to some amateur and professional musicians who listened to each song (mainly to identify the requested range of the tune to be played) and played in MIDI keyboard and guitars several times the same tune with different embellishments. We colected 20 interpretations of each tune.
- They were given the guidelines:
    - to follow a meter and set it in the MIDI file
    - to establish the correct tonality in the MIDI file
    - to play only one note at a time (play in a monophonic way)

Application: Music Recognition

- The corpus consists of a set of monophonic 8-12 bar incipits of 20 worldwide well known tunes of different musical genres: (Bolero, Cucaracha, Jinglebells, Guantanamera, Happy Birthday, Yesterday, ...)
- For each song a canonic tune was created by writing the score in a musical notation application and exported to MIDI and MP3 format
- The MP3 files were given to some amateur and professional musicians who listened to each song (mainly to identify the requested range of the tune to be played) and played in MIDI keyboard and guitars several times the same tune with different embellishments. We colected 20 interpretations of each tune.
- They were given the guidelines:
  - to follow a meter and set it in the MIDI file
  - to establish the correct tonality in the MIDI file
  - to play only one note at a time (play in a monophonic way)

Application: Music Recognition

- The corpus consists of a set of monophonic 8-12 bar incipits of 20 worldwide well known tunes of different musical genres: (Bolero, Cucaracha, Jinglebells, Guantanamera, Happy Birthday, Yesterday, ...)
- For each song a canonic tune was created by writing the score in a musical notation application and exported to MIDI and MP3 format
- The MP3 files were given to some amateur and professional musicians who listened to each song (mainly to identify the requested range of the tune to be played) and played in MIDI keyboard and guitars several times the same tune with different embellishments. We colected 20 interpretations of each tune.
- They were given the guidelines:
  - to follow a meter and set it in the MIDI file
  - to establish the correct tonality in the MIDI file
  - to play only one note at a time (play in a monophonic way)

Application: Music Recognition

- The corpus consists of a set of monophonic 8-12 bar incipits of 20 worldwide well known tunes of different musical genres: (Bolero, Cucaracha, Jinglebells, Guantanamera, Happy Birthday, Yesterday, ...)
- For each song a canonic tune was created by writing the score in a musical notation application and exported to MIDI and MP3 format
- The MP3 files were given to some amateur and professional musicians who listened to each song (mainly to identify the requested range of the tune to be played) and played in MIDI keyboard and guitars several times the same tune with different embellishments. We colected 20 interpretations of each tune.
- They were given the guidelines:
  - to follow a meter and set it in the MIDI file
  - to establish the correct tonality in the MIDI file
  - to play only one note at a time (play in a monophonic way)

Application: Music Recognition

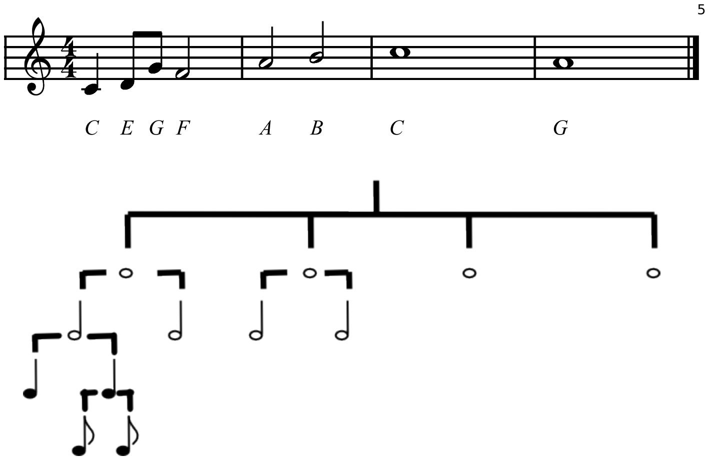
- The corpus consists of a set of monophonic 8-12 bar incipits of 20 worldwide well known tunes of different musical genres: (Bolero, Cucaracha, Jinglebells, Guantanamera, Happy Birthday, Yesterday, ...)
- For each song a canonic tune was created by writing the score in a musical notation application and exported to MIDI and MP3 format
- The MP3 files were given to some amateur and professional musicians who listened to each song (mainly to identify the requested range of the tune to be played) and played in MIDI keyboard and guitars several times the same tune with different embellishments. We colected 20 interpretations of each tune.
- They were given the guidelines:
  - to follow a meter and set it in the MIDI file
  - to establish the correct tonality in the MIDI file
  - to play only one note at a time (play in a monophonic way)

Application: Music Recognition

Objective:

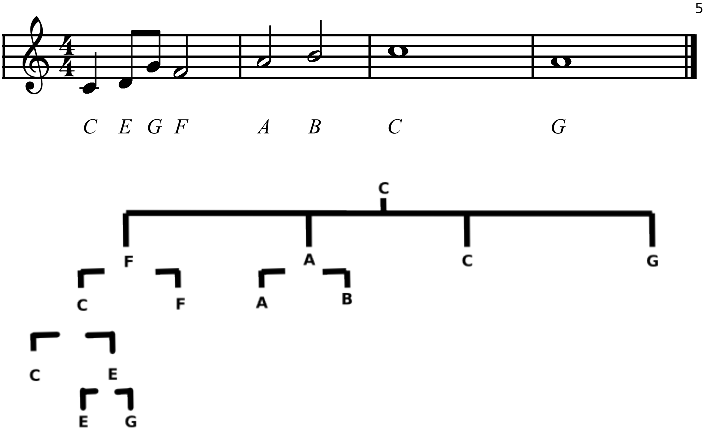
- to learn a stochastic tree edit distance in order to recognize the song played by musicians
- Use of a tree-structured representation

Music Representation



- Logarithmic scale of musical notes is represented by the depth in the tree

Music Representation



- Leaves are labeled by the pitch of the notes
- Internal nodes are labeled according to musical rules

Outline

Experimental Setup

- We exclude the canonic tunes and we use a 5-fold cross validation on the remaining set of tunes
- Learning
  - For each tune in the training set we create a pair `(canonic-tune, tune)`.
  - We learn the edit weight from these learning tree pairs.
- Testing
  - We use a 1-NN with the canonic tunes as a set of prototypes
  - We classify each tune of the test set in the class of the canonic tune which is at the minimal stochastic edit distance
- We compare the results obtained with the same testing procedure using non learned diagonal weight matrix.

Results

- 80% of correct classification learning the edit weight
- 51% with a non learned diagonal weight matrix

Outline

Future work

- Related with Learning Distances
  - Apply the learning method to other problems
    - already done in image recognition
  - Taking into account the context of the operations
  - Learning stochastic tree transducers
  - Apply the ideas to other structures
    - subclases of graphs
    - multistrings (some work in progress)
- Related with Music
  - Larger datasets
  - from sound instead of MIDI files
    - it is a task in a large spanish project: transcription, interactivity

Future work

- Related with Learning Distances
  - Apply the learning method to other problems
    - already done in image recognition
  - Taking into account the context of the operations
  - Learning stochastic tree transducers
  - Apply the ideas to other structures
    - subclases of graphs
    - multistrings (some work in progress)
- Related with Music
  - Larger datasets
  - from sound instead of MIDI files
    - it is a task in a large spanish project: transcription, interactivity

Future work

- Related with Learning Distances
  - Apply the learning method to other problems
    - already done in image recognition
  - Taking into account the context of the operations
  - Learning stochastic tree transducers
  - Apply the ideas to other structures
    - subclases of graphs
    - multistrings (some work in progress)
- Related with Music
  - Larger datasets
  - from sound instead of MIDI files
    - it is a task in a large spanish project: transcription, interactivity

Future work

- Related with Learning Distances
  - Apply the learning method to other problems
    - already done in image recognition
  - Taking into account the context of the operations
  - Learning stochastic tree transducers
  - Apply the ideas to other structures
    - subclases of graphs
    - multistrings (some work in progress)
- Related with Music
  - Larger datasets
  - from sound instead of MIDI files
    - it is a task in a large spanish project: transcription, interactivity

Future work

- Related with Learning Distances
  - Apply the learning method to other problems
    - already done in image recognition
  - Taking into account the context of the operations
  - Learning stochastic tree transducers
  - Apply the ideas to other structures
    - subclases of graphs
    - multistrings (some work in progress)
- Related with Music
  - Larger datasets
  - from sound instead of MIDI files
    - it is a task in a large spanish project: transcription, interactivity

Future work

- Related with Learning Distances
  - Apply the learning method to other problems
    - already done in image recognition
  - Taking into account the context of the operations
  - Learning stochastic tree transducers
  - Apply the ideas to other structures
    - subclases of graphs
    - multistrings (some work in progress)
- Related with Music
  - Larger datasets
  - from sound instead of MIDI files
    - it is a task in a large spanish project: transcription, interactivity

Future work

- Related with Learning Distances
  - Apply the learning method to other problems
    - already done in image recognition
  - Taking into account the context of the operations
  - Learning stochastic tree transducers
  - Apply the ideas to other structures
    - subclases of graphs
    - multistrings (some work in progress)
- Related with Music
  - Larger datasets
  - from sound instead of MIDI files
    - it is a task in a large spanish project: transcription, interactivity

Future work

- Related with Learning Distances
  - Apply the learning method to other problems
    - already done in image recognition
  - Taking into account the context of the operations
  - Learning stochastic tree transducers
  - Apply the ideas to other structures
    - subclases of graphs
    - multistrings (some work in progress)
- Related with Music
  - Larger datasets
  - from sound instead of MIDI files
    - it is a task in a large spanish project: transcription, interactivity

Future work

- Related with Learning Distances
  - Apply the learning method to other problems
    - already done in image recognition
  - Taking into account the context of the operations
  - Learning stochastic tree transducers
  - Apply the ideas to other structures
    - subclases of graphs
    - multistrings (some work in progress)
- Related with Music
  - Larger datasets
  - from sound instead of MIDI files
    - it is a task in a large spanish project: transcription, interactivity