

Generación de secuencias de obras de arte basado en vecindad y RNN*

Ignacio Gatti, Silvia Schiaffino, and Jorge Andrés Díaz Pace

ISISTAN (CONICET - UNCPBA), Campus Universitario, Tandil, Argentina
{ignacio.gatti, silvia.schiaffino, andres.diazpace}@isistan.unicen.edu.ar

Resumen En los museos las obras de arte son distribuidas en un espacio físico por curadores. Dicha distribución no es causa y efecto del azar, sino buscando un delicado equilibrio entre emoción y razón. A partir de la digitalización y acceso masivo a las obras de arte, surge el interrogante de si es posible generar una secuencia automática de obras de artes de acuerdo a los intereses del espectador. Este trabajo busca definir el problema de la *generación de secuencias de obras de arte*. Se presentan dos enfoques que abordan la problemática haciendo énfasis en la organización intrínseca de la secuencia basadas en técnicas de *vecindad* y de *Recurrent Neural Networks*. Se entiende que esta perspectiva se acerca más al tipo de recomendación que haría un curador. Los enfoques son evaluados sobre un dataset que consiste de 52 recorridos definidos por los curadores del Museo del Prado y el Rijksmuseum. Si bien los resultados son preliminares, se observa que los tours predichos por ambos enfoques presentan semejanzas con los tours originales.

Keywords: Recomendación de secuencias · Recurrent Neural Network · Deep Autoencoder · Ontología.

1. Introducción

Desde la postrimería de la Modernidad, los museos han sido el lugar donde el patrimonio cultural se almacena y se expone al público [6]. Son lugares sociales, donde las personas van a contemplar las obras de arte y la cultura de una nación. Generalmente, las obras de arte se distribuyen a lo largo de sus salas. Dicha tarea está a cargo de los curadores, que organizan las obras de artes buscando un delicado equilibrio entre emoción y razón [6]. El color, la forma, la línea, la composición, el autor, el estilo se presentan de forma caótica y es el curador quien busca un orden que permita al espectador disfrutar de ellas.

Hoy en día las obras de arte también se encuentran digitalizadas y circulan por la web. Incluso, varios museos ofrecen recorridos virtuales sobre su patrimonio (por ejemplo, los tours virtuales del Museo del Prado¹). Más allá de la

* Parcialmente financiado por CONICET PIP Project No. 11220150100030CO y por ANPCyT Project No. 2016-2973.

¹ <https://www.museodelprado.es/mi-prado/recorridos-recomendados>

digitalización, el problema del curador sigue siendo el mismo: organizar una secuencia de obras de arte que sea cautivante para el espectador de principio a fin. De aquí surge un interrogante: ¿es posible generar una secuencia automática de obras de artes?

A priori, el problema puede verse como una variación de la generación de playlists donde la diferencia radica en el ítem a organizar: en el caso de la playlist son canciones; en el nuestro, obras de arte. Este trabajo se focaliza en definir el problema de la *generación de secuencia obras de arte*. Se hipotetiza que es factible definir dicha problemática siguiendo la perspectiva adoptada en el dominio de la música [12,2,4]. De esta manera es posible tomar distancia respecto del enfoque clásico, que pone el foco en la optimización del espacio recorrido más que en la organización interna de la secuencia [15,11,10]. Al mismo tiempo se proponen algunas variantes respecto de [12] y [4] que son necesarias para explotar el dominio de las obra de arte.

Se proponen dos enfoques: basado en la *vecindad* y basado en *Recurrent Neural Networks* (RNN). El primero es un enfoque híbrido que utiliza una técnica de *deep autoencoder* para extraer *features visuales* [7] y un *grafo social de influencia* para extraer *features contextuales* [5]. Mientras que el segundo enfoque utiliza la misma técnica de *deep autoencoder* para extraer *features visuales* [7] y una técnica de RNN para predecir el siguiente elemento de la secuencia [7]. La evaluación de ambos enfoques se realiza siguiendo la metodología de [3,4]. El objetivo de la evaluación es analizar la calidad de la secuencia generada. Para ello, se toman como referencia 52 tours definidos por los curadores del Museo del Prado y el Rijksmuseum². Si bien los resultados son preliminares, se observa que los tours predichos por ambos enfoques presentan semejanzas con los tours originales.

El resto del artículo se organiza de la siguiente manera: en la sección 2 se discuten trabajos relacionados con la *generación de secuencias de obras de arte*; en la sección 3 se presentan los enfoques desarrollados; en la sección 4 se provee una evaluación de ambos enfoques; finalmente, en la sección 5 se esboza una conclusión y posibles trabajos futuros.

2. Trabajos relacionados

En la actualidad y hasta donde sabemos, la literatura aborda la problemática de la *generación de secuencias de obras de arte* como un problema de optimización. [15,11,10] corren el foco desde la generación de secuencias hacia la optimización del recorrido. Es decir, dada la distribución de las obras dentro del museo, el objetivo de ellos es encontrar aquel “recorrido que maximice la satisfacción del usuario y minimice la distancia total” [15]. Estos enfoques modelan la situación como un problema de optimización, donde se debe maximizar la satisfacción del usuario teniendo presente algunas restricciones (por ejemplo, tiempo, distancia). En síntesis, en la literatura el enfoque clásico consiste en mapear la problemática a un modelo de Programación Lineal (PL) [15,11,10].

² <https://www.rijksmuseum.nl/en/tours>

Sin embargo, en el dominio de la música se aborda el problema de la generación desde otra perspectiva. Aquí se define el problema de la generación de playlists como la creación de una playlist dada una semilla o cierta información semántica [1]; donde la semilla puede ser un conjunto de canciones escuchadas anteriormente (denominado historia) y la información semántica es alguna *feature* como “música disco”. En el caso de una canción es posible obtener dos tipos de *features*: *auditivas* y *contextuales* [8]. Las primeras, se obtienen directamente de la señal de audio; las segundas, del contexto musical[8].

[12] proponen un enfoque interesante para la generación de listas de reproducción de música. Ellos modelan este problema como una cadena de Markov de primer orden, donde cada estado corresponde a una canción. Para calcular las probabilidades de cada transición analizan tres modelos de distribución de probabilidades: uniforme, basada en la popularidad del artista y basada en los k-vecinos más cercanos (utilizando una medida de similitud basadas en *tags* y *features* extraídos de la señal de audio). En la misma línea, [13] definen un enfoque basado en *hipergrafos*. Esta estructura define un arco como un subconjunto arbitrario de los nodos, en lugar de un par [13]. Así, ellos modelan cada arco como un *feature* y le asocian un peso a cada uno según la importancia que tiene para la generación de la lista. La lista de reproducción es modelada como una cadena de Markov; esto les permite calcular los pesos utilizando el algoritmo L-BFGS-B [13]. Para obtener la lista de reproducción usan algoritmo de *random walk* a través del *hipergrafo*. [4] proponen un enfoque basado en RNN para resolver este problema. Su idea consiste en modelar una lista de reproducción como una secuencia de segmentos musicales y el objetivo es predecir el siguiente segmento. Hipotetizan que cada canción tiene sus propios segmentos musicales (por ejemplo, introducción, verso, coro) y que éstos pueden traducirse a *features* utilizando la salida de una “hidden layer” de un clasificador basado en *Deep Neural Network* (DNN) [4]. En síntesis, el objetivo de este modelo es lograr una transición de calidad entre las canciones.

El trabajo presentado sigue la perspectiva adoptada en el dominio de la música [12,4]. De esta manera, se define el problema desde una arista diferente a la propuesta por [15,11,10], haciendo énfasis en la *generación de secuencias de obras de arte* como un conjunto ordenado de obras de arte donde cada una influye en la contemplación de las demás. Asimismo, el enfoque presentado se diferencia de [4]: primero, utilizando un deep autoencoder para la extracción de *features*; segundo, prediciendo la próxima obra de arte de la secuencia, dando un paso más respecto [4], cuyo modelo predice el próximo segmento que mejor se acopla a la semilla.

3. Enfoque

Este trabajo presenta dos enfoques para la generación de secuencias de obras de arte. Si bien ambos se inspiran en enfoques propuestos en el dominio de la música [12,4], los propuestos en este trabajo buscan hacer hincapié en la naturaleza visual de la obra de arte para la generación de secuencias.

De manera análoga a la generación de playlists, se puede definir el problema de la generación de secuencias de obras de arte como la creación de una secuencia dada una semilla o alguna información semántica. Pero en este caso la *semilla* sería un conjunto de obras de arte contempladas previamente (denominada historia) y la información semántica sería una característica como “retratos”.

Formalmente, este problema puede definirse como la predicción de una obra de arte en el tiempo t dada la una historia $h = \alpha_a^0, \alpha_b^1, \dots, \alpha_i^{t-1}$. La historia es una *serie temporal* [3] donde en cada momento t se observa una obra de arte α_i .

3.1. Enfoque basado en *vecindad*

Uno de los enfoques clásicos para la generación de playlists consiste en explorar la “co-ocurrencia entre canciones basándose en similitud que existen entre ellas” [12]. Este enfoque, conocido como enfoque basado en la *vecindad* [2], mapea las canciones a vectores en un espacio n-dimensional de manera tal de poder operar con ellos algebraicamente. El objetivo es encontrar un subconjunto de canciones que sean “vecinas” y seleccionar una de ellas al azar [12,2].

A partir de lo expresado anteriormente es posible definir un enfoque que pueda seleccionar la próxima obra de arte de una serie temporal buscando la obra “más similar” con respecto a la historia h . Siguiendo el enfoque de [12], se debe mapear cada obra de arte a un vector n-dimensional.

Generación de *features* La obra de arte de arte es un objeto complejo que puede ser representado de diversas maneras. Siguiendo la taxonomía de [8] presentada para el dominio de la música, se puede definir a la obra de arte a partir de sus *features visuales* o de sus *features contextuales*.

Las *features visuales* son aquellas características que se relacionan con lo plástico, tal como el color, la línea, el estilo, la composición [6]. En este trabajo, se utiliza una técnica de deep autoencoder para su extracción. Básicamente, el objetivo de esta técnica es conseguir una representación n-dimensional de la imagen que contiene las principales *features* [7]. Un deep autoencoder consta de 2 módulos: *encoder* y *decoder*. El primero aprende a mapear una imagen a un conjunto de features, denominado $Encode_{layer}$; mientras que el segundo aprende a reconstruir la imagen original a partir de $Encode_{layer}$. Generalmente, una vez entrenado el modelo, el *decoder* suele ser descartado y sólo se conserva el *encoder* para poder mapear una imagen a un conjunto de features. Se define la arquitectura del deep autoencoder siguiendo el modelo definido por [14]. Dicho modelo tiene la particularidad que fue diseñado para manipular obras de artes.

Las *features semánticas* son el conjunto de características que se derivan del contexto de la imagen. Generalmente se tratan de datos en formato textual. En el caso de la obra de arte se trata de todo el discurso generado en torno a ella (por ejemplo, el nombre del pintor, el año, la sinopsis generada por algún experto). En el enfoque propuesto se toma el concepto de relaciones semánticas y fuentes de conocimiento. Siguiendo las ideas de [18,16], se busca enriquecer el modelo agregando información semántica derivada del contexto de las obras. La propuesta

de este trabajo es tomar la relación de “influencia” entre los artistas, presente en la base de conocimiento *DBpedia Ontology*³. Dicha relación es representada en un *grafo social de influencias* entre los artistas. A partir de éste es posible calcular un coeficiente que refleje la influencia que ejercen algunos artistas sobre otros [5]. Básicamente, este coeficiente busca mantener cerca aquellas obras cuyos artistas están relacionados entre sí. Por ejemplo, en el dominio de las artes visuales es sabido que la obra de Dominique Ingres (1780-1867) está influenciada por la de Rafael Sanzio (1483-1520). Asimismo, Pierre Renoir (1841-1919) fue influenciado por la obra de Ingres. Con lo cual, sería factible que las obras de estos artistas estén relacionadas entre sí y aparezcan en el conjunto de obras seleccionadas. En un trabajo reciente se produjeron evidencias que utilizar un grafo de influencia social mejora el rendimiento en RS multidominios [5].

Generación de secuencia A partir del mapeo de obras de arte a un espacio n -dimensional, se propone el siguiente enfoque para la generación de secuencia de obras de arte: dada una historia $h = \alpha_a^0, \alpha_b^1, \dots, \alpha_i^{t-1}$, mapear cada obra de arte α_i^a un vector v_{α_i} utilizando el deep autoencoder. Luego calcular el vector \bar{v} , que es el promedio de todos los v_i que hay en h (eq. 1). Este vector representa la obra de arte “promedio”, es decir, una obra de arte que surgiría de la suma de todas las obras de arte en h . Siguiendo los conceptos de [6], donde destaca que las obras de arte se relacionan entre sí, se hipotetiza que \bar{v} representaría una obra de arte que podría ser parte de h .

Con \bar{v} se buscan aquellos v_j más similares utilizando la Similitud del Coseno [9]. De esta forma se obtienen así una lista de obras de arte ordenadas por la similitud con respecto a \bar{v} . En síntesis, el objetivo aquí es buscar aquella obra de arte existente en el repositorio tal que sea similar a \bar{v} (y distinta de todas las obras de arte que en h).

Finalmente se reordena la lista aplicando el *coeficiente de influencia* (eq. 2) para cada v_j . De esta forma la similitud entre dos obras de arte se calcula como se muestra en eq. 3. Luego se seleccionan los 200 vecinos más cercanos y se selecciona uno al azar [12,2].

$$\bar{v} = \frac{\sum_{i \in h}^n v_{\alpha_i}}{n} \quad (1)$$

$$inf(x, y) = \frac{1}{shortest_path(x, y)} \quad (2)$$

$$inf_x = \sum_{i \in h}^n inf(x, i)$$

$$dist_x = cos_{sim}(x, \bar{v}) * inf_x \quad (3)$$

³ <https://wiki.dbpedia.org/services-resources/ontology>

3.2. Enfoque basado en RNN

Siguiendo el enfoque propuesto por [4] es posible definir un modelo de RNN cuyo objetivo sea predecir la próxima obra de arte α_i de la serie temporal h .

Las RNN son una familia de DNN que se utilizan para procesar datos secuenciales. Estas redes tienen la particularidad de usar conexiones recurrentes, compartiendo parámetros en diferentes partes de un modelo. Por lo tanto, la información puede circular dentro de la red durante un tiempo arbitrario [7]. Matemáticamente, una RNN se define de la siguiente manera:

$$h^t = \sigma_h(W_h x^t + U_h h^{t-1} + b_h) \quad (4)$$

$$y^t = \sigma_y(W_y h^t + b_y) \quad (5)$$

donde t representa un momento puntual; x^t la entrada de la función; h^{t-1} el resultado de la función en momento $t - 1$; W_h , W_y , U_h , b_h y b_y los coeficientes de la funciones; y σ_h y σ_y las funciones de activación [7]. En otras palabras, la salida de un RNN en un momento t es una función recursiva que depende de las salidas generadas anteriormente. Este tipo de redes generalmente modelan tareas que requieren predecir el futuro desde el pasado [7].

Generación de *features* Como se dijo anteriormente, la obra de arte es un objeto complejo, que puede ser representado de diversas maneras. Sin embargo, es necesario tener presente que una RNN solo acepta como entrada válidos valores numéricos. Por eso se utiliza la técnica de deep autoencoder. Esta técnica permite representar a cada obra de arte como un conjunto de *features* numéricos que son representativos para el modelo.

De esta manera es posible mapear la serie temporal h a un conjunto de series temporales h_1, h_2, \dots, h_n donde se observa la evolución de cada feature x_i a lo largo del tiempo.

Arquitectura de RNN La arquitectura de la RNN se define siguiendo los esquemas más utilizados en la literatura, que han superado a enfoques basados en cadenas de Markov en tareas como *speech recognition* o *language modelling* [7]. Básicamente, el éxito de estas arquitecturas se basa en el uso de un tipo particular de capa denominado *Long Short Term Memory* (LSTM). Esta capa tiene la capacidad de decidir cuánta importancia otorgarle a la entrada actual y cuánto a la historia pasada [4].

En este caso, el modelo propuesto es una RNN que predice en un tiempo t las features $\hat{X} = \hat{x}_0^0, \hat{x}_1^0, \dots, \hat{x}_i^0, \dots, \hat{x}_{300}^0$. De esta forma, se podría pensar que se trata de un modelo que en cada tiempo t posee 300 unidades h_i^t trabajando, una por cada feature x_i^t (Fig. 1).

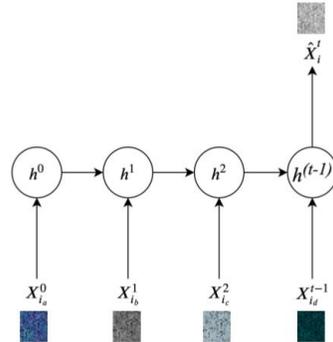


Figura 1. Diagrama de una RNN para la predicción del feature \hat{x}_i .

Reconstrucción de la obra de arte El enfoque propuesto tiene la capacidad de predecir en un tiempo t las features $\hat{X} = \hat{x}_0^0, \hat{x}_1^0, \dots, \hat{x}_i^0, \dots, \hat{x}_{300}^0$ dada una historia h . A diferencia de [4], el objetivo de este trabajo es predecir las futuras obras de arte a partir de h y no un conjunto de *features*.

Con el fin de dar como resultado final en cada tiempo t una obra de arte “válida”, se realiza una tarea reconstrucción de la obra de arte a partir de los *features* predichos. Dado que con el deep autoencoder es posible codificar todas las obras de artes del dataset, entonces es factible mapear cada obra de arte α_i a su vector de features X y almacenarlo en un repositorio, denominado R_{codes} . De esta forma, aplicando la similitud del coseno [9] entre \hat{X} y todos los códigos guardados en R_{codes} se puede obtener un X tal que $X \approx \hat{X}$. Y dado que X se encuentra asociado unívocamente a una obra de arte α , entonces $\hat{X} \approx \alpha$ (Fig. 2).

3.3. Entrenamiento de los modelos

Para el entrenamiento de los modelos se establece un proceso que consistió en los siguientes pasos: obtención de los datos, entrenamiento del modelo de deep autoencoder y entrenamiento del modelo de RNN.

Obtención de los datos Para la construcción del dataset se procede en dos pasos: por un lado se toman 40.000 obras de artes pertenecientes a 9 períodos de la historia de las artes, disponibles en WikiArt⁴. Por el otro, se toman 52 recorridos definidos por los curadores del Museo del Prado y el Rijksmuseum. Estos recorridos se encuentran disponibles en la web y son ejemplos de secuencias de obras de arte realizadas por expertos.

⁴ <https://www.wikiart.org/>

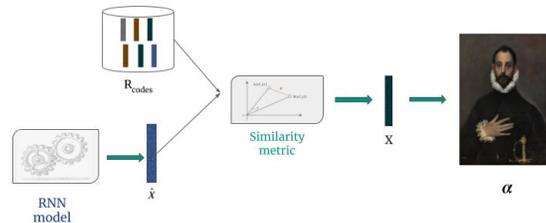


Figura 2. Recuperación de la obra de arte

Entrenamiento del modelo de deep autoencoder Siguiendo la arquitectura definida por [14], se define un modelo de deep autoencoder con 6 capas para el *encoder* y otras 6 para el *decoder*. Se usan capas Convolucionales por ser las utilizadas en el estado del arte para lidiar con imágenes [7]. Asimismo, se define una $Encode_{layer}$ de 300×1 , por ser una de las configuraciones más utilizadas en la literatura [7].

El modelo es entrenado con las 40.000 obras de artes. Cada imagen tenía un tamaño de $128 \times 128 \times 3$. Una vez entrenado, se codifican las imágenes de las obras de arte pertenecientes a los 52 recorridos. Para cada imagen se obtiene un vector de 300 features $X = [x_0, \dots, x_i, \dots, x_{300}]$.

Entrenamiento del modelo de RNN Tomando como referencia la arquitectura desarrollada por [4], se define una RNN con 300 unidades, una por cada feature x_i . Cada unidad cuenta con una capa Convolutiva y dos LSTM.

Para su entrenamiento, se toma un subconjunto de los recorridos del dataset. Específicamente, se dividen los 52 recorridos en dos datasets: uno de entrenamiento y otro de validación. El primero cuenta con 39 recorridos y se utiliza en la etapa de entrenamiento; mientras que el segundo cuenta con 13 recorridos y se usa en la etapa de evaluación.

Asimismo, el entrenamiento se divide en dos fases: búsqueda del *learning rate* óptimo y entrenamiento del modelo. Dado que el entrenamiento de una DNN consiste en la optimización de una función costo utilizando una técnica denominada *stochastic gradient descent*, la parametrización de la técnica influye en la calidad de predictiva del modelo [7]. Un valor inapropiado del *learning rate* puede provocar que el modelo “caiga” en un óptimo local o por el contrario puede producir overfit [7]. En ambos casos se obtiene un modelo con una capacidad de generalización mediocre.

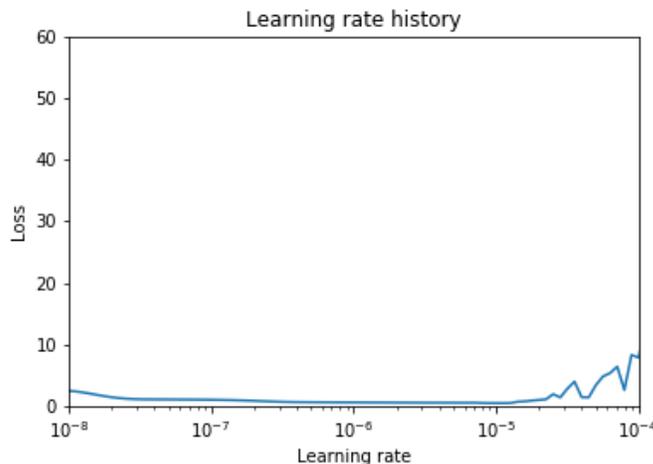


Figura 3. Comportamiento del modelo para diferentes valores del parámetro *learning rate*.

La Fig. 3 muestra el comportamiento del modelo para diversos valores del *learning rate*, observándose que el valor $10e - 6$ se aproxima al óptimo global. Una vez establecido el *learning rate* se entrena nuevamente la RNN con el fin de generar el modelo final.

4. Evaluación

Cuando se trabaja sobre series temporales, la calidad de un modelo de predictivo se mide por la capacidad de predecir los valores futuros (es decir, posteriores a un momento t) teniendo en cuenta la historia (es decir, los valores anteriores a t) [3].

En el caso de la generación de secuencias de obras de arte, la historia se corresponde con la semilla, es decir, con las obras de arte contempladas anteriormente por el espectador; mientras que los valores futuros son las obras de arte que completan el recorrido siguiendo los criterios que utilizan los curadores para organizar las obras en el espacio. Dado que ambos enfoques mapean cada obra de arte α a un vector de *features* $X = [x_0, \dots, x_i, \dots, x_{300}]$ donde cada x_i es un valor numérico proyectado a lo largo del tiempo, resulta factible evaluar a los enfoques como un modelo predictivo para series temporales.

4.1. Metodología de evaluación

Una predicción sobre una serie temporal $s = t_1, t_2, \dots, t_i$ se define como la estimación de un valor futuro \hat{t}_{i+n} a partir de los valores conocidos hasta el tiempo i [3]. Teniendo presente dicha conceptualización, [3] propone una metodología para evaluar modelos de predicción sobre una serie temporal:

- Dada una serie temporal $s = t_1, t_2, \dots, t_n$, dividir la serie en dos: $s_{train} = t_1, t_2, \dots, t_h$ y $s_{test} = t_{h+1}, t_{h+2}, \dots, t_n$
- Dado un modelo m , entrenar m utilizando únicamente s_{train}
- Predecir con m los valores hasta el tiempo n dada s_{test} , obteniendo así $\hat{s}_{test} = \hat{t}_{h+1}, \hat{t}_{h+2}, \dots, \hat{t}_n$
- Analizar el promedio de error en la predicción de \hat{s}_{test} calculando mean absolute error (MAE) [17].

Siguiendo la metodología de [3] se puede definir la siguiente metodología de evaluación para el modelo predictivo de secuencias de obras de artes:

- Dada una serie temporal $s = \alpha_a^0, \alpha_b^1, \dots, \alpha_n^t$, donde en cada momento t se observa una obra de arte α_i , dividir la serie en dos: $s_{train} = \alpha_a^0, \alpha_b^1, \dots, \alpha_i^h$ y $s_{test} = \alpha_{i+1}^{h+1}, \alpha_{i+2}^{h+2}, \dots, \alpha_n^t$.
- Dado un modelo m , entrenar m utilizando únicamente s_{train}
- Predecir con m los valores hasta el tiempo n dada s_{test} , obteniendo así $\hat{s}_{test} = \hat{\alpha}_{i+1}^{h+1}, \hat{\alpha}_{i+2}^{h+2}, \dots, \hat{\alpha}_n^t$
- Analizar el promedio de error en la predicción de \hat{s}_{test} calculando MAE [17].

Vale recordar que el enfoque propuesto descompone la serie temporal $s = \alpha_a^0, \alpha_b^1, \dots, \alpha_n^t$ en un conjunto de series temporales s_1, s_2, \dots, s_n , una por cada feature x_i . Con lo cual, la metodología detallada anteriormente se debe realizar por cada s_i , siendo el error global (MAE_G) igual al promedio de los errores obtenidos para cada s_i (eq. 6):

$$MAE_G = \frac{1}{N} \sum_i MAE_i \quad (6)$$

4.2. Dataset

El dataset cuenta con los 52 tours descriptos en 3.3. Para la experimentación se divide aleatoriamente en dos: uno de entrenamiento y otro de validación. El primero cuenta con 39 recorridos y se utiliza en la etapa de entrenamiento; mientras que el segundo posee con 13 recorridos y se usa en la etapa de evaluación. Dicha división, sigue los lineamientos propuestos por [3].

Entendiendo que se trata de un dataset pequeño, se utiliza la estrategia de k-fold [17] con el fin de realizar una experimentación más robusta.

4.3. Resultados experimentales

Siguiendo la metodología presentada se evalúa el desempeño de ambos enfoques sobre el dataset de validación analizando la métrica MAE . Se entiende que los vectores generados por el deep autoencoder para cada obra de arte i se encuentran distribuidos en el espacio de manera tal que aquellas obras de artes que son similares se encuentran más cerca. Por lo tanto, se observa cuán lejos se encuentra la predicción \hat{x}_i^t para cada valor del feature x_i en el tiempo t (Fig. 4).

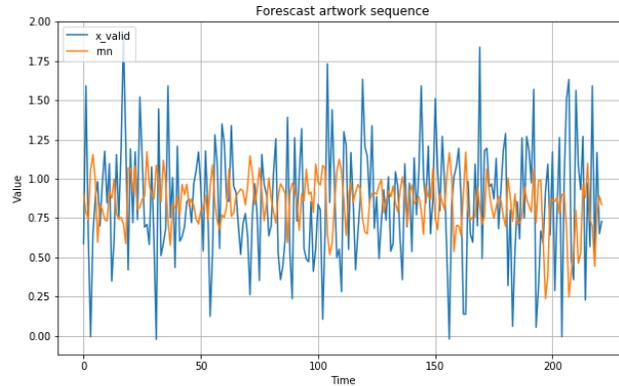


Figura 4. Predicción de s_{test} para el feature x_i .

Dado que la definición del tamaño de la ventanas se realiza de manera empírica, se comparan la variación en la calidad de la predicción para diferentes tamaño de ventanas. En promedio, cada recorrido tiene 16 obras de artes, por lo tanto se configuran modelos con tamaños de ventanas contenidos en el intervalo 3-8. Se entiende que una ventana mayor carecería de sentido porque estaría conteniendo más de la mitad del recorrido.

La Fig. 5 muestra el comportamiento de ambos enfoques para los diferentes tamaños de ventana. En líneas generales se observa que el modelo de RNN tiene una mejora 8% respecto del modelo basado en *vecindad*. Asimismo, resulta interesante observar cómo el rendimiento se degrada en ambos enfoques a medida que se aumenta el tamaño de la ventana. En principio, esto se puede interpretar cómo un indicio de que existe una relación entre la calidad de la secuencia y el tamaño de la ventana. Llevado a la práctica, esto supondría que el usuario sólo necesitaría brindarle al sistema de tres a cinco obras de su interés para que éste genere una secuencia que siga sus gustos.

La fig. 6 muestra la distribución del MAE_G para cada configuración, discriminado por tour. Es interesante observar que el enfoque de RNN mantiene un comportamiento similar a través de los diferentes tours predecidos, es decir, que el MAE_G obtenido para cada tour con las diferentes configuraciones sigue una misma tendencia. Mientras que el enfoque basado en vecindad resulta más sensible al tamaño de la historia h .

Finalmente, la fig. 7 muestra la reconstrucción de un tour recomendado por el modelo de RNN con una ventana de 3 así como el tour original (que fue diseñado por los curadores del museo del Prado). Si bien resulta complejo determinar la calidad de la secuencia generada desde un punto de vista “artístico”, vale destacar que algunas de las obras recomendadas se encuentran en el tour original.

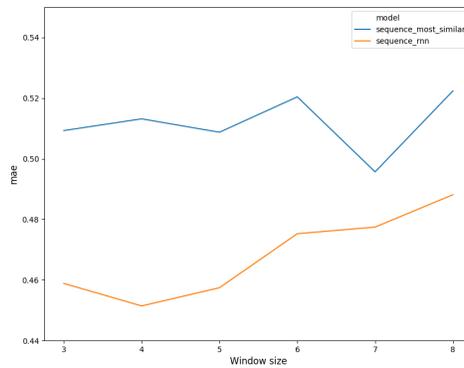


Figura 5. MAE_G discriminado por tamaño de ventana.

MAE

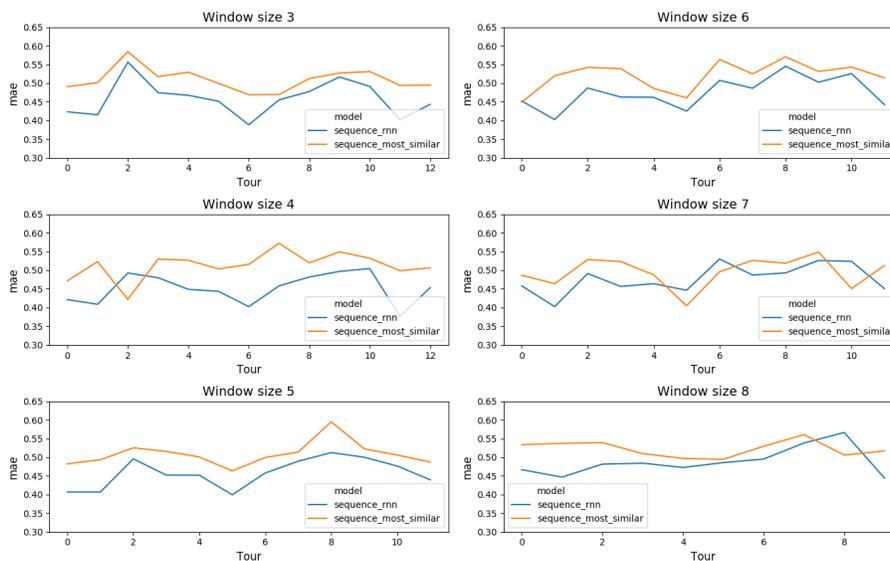


Figura 6. MAE_G discriminado por tour.

5. Conclusiones

Este trabajo presenta dos enfoques para la *generación de secuencias de obras de arte*. Por un lado, se define un modelo que tiene en cuenta la complejidad de la obra de arte para seleccionar la siguiente obra de la secuencia, utilizando

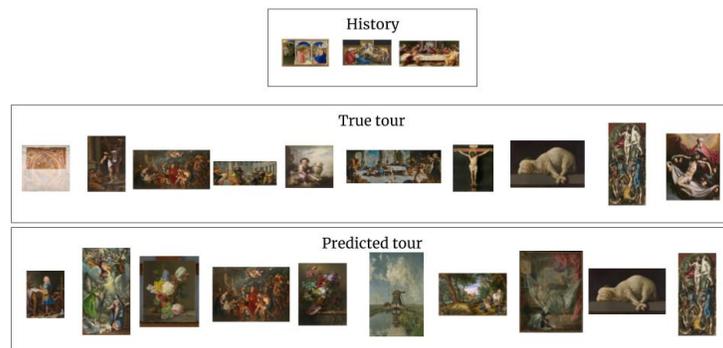


Figura 7. Reconstrucción de la secuencia generada por el modelo de RNN

un deep autoencoder para extraer *features visuales*, junto con un *grafo social de influencia* para definir *features contextuales*. Por otro lado, se desarrolla un modelo basado en deep autoencoder y RNN, cuyo objetivo es predecir la siguiente obra de arte a partir del comportamiento de las *features visuales* a lo largo del tiempo.

A diferencia de los enfoques clásicos [15,11,10], ambos enfoques presentan un abordaje de la problemática que hace énfasis en la organización intrínseca de la secuencia. Se entiende que esta perspectiva se acerca más al tipo de recomendación que haría un curador.

Los enfoques son evaluados siguiendo la metodología establecida por [3]. Si bien los resultados son preliminares, se observa que el enfoque basado en RNN tiene mejor desempeño que el enfoque basado *vecindad*. Si bien no es posible aseverar que se tratan de secuencias de calidad, se observa que en los tour generados por el modelo de RNN aparecen obras que se encuentran en el tour verdadero. Por eso, en trabajos futuros se buscará validar la calidad de las secuencias generadas con expertos en el dominio.

Por último, vale destacar posibles trabajos futuros que se desprenden de aquí. En primer lugar, parecería auspicioso desarrollar un modelo de RNN que también tenga presente las *features semánticas*. Asimismo, hasta el momento en el modelo de RNN, la predicción de cada \hat{x}_i^t se hace solo teniendo en cuenta la historia pasada del *feature* x_i , por lo tanto, sería interesante analizar cómo influyen las demás *features* x_j a la hora de predecir la *feature* \hat{x}_i^t .

Referencias

1. Barrington, L., Oda, R., Lanckriet, G.R.: Smarter than genius? human evaluation of music recommender systems. In: ISMIR. vol. 9, pp. 357–362. Citeseer (2009)

2. Bonnin, G., Jannach, D.: A comparison of playlist generation strategies for music recommendation and a new baseline scheme. In: Workshops at the twenty-seventh AAAI conference on artificial intelligence (2013)
3. Chatfield, C.: Time-series forecasting. Chapman and Hall/CRC (2000)
4. Choi, K., Fazekas, G., Sandler, M.: Towards playlist generation algorithms using rnns trained on within-track transitions. arXiv preprint arXiv:1606.02096 (2016)
5. Gatti, I., Schiaffino, S., Diaz Pace, A.: Multi-domain recommender system based on embeddings and semantic networks. In: XXIV Congreso Argentino de Ciencias de la Computación (La Plata, 2018). (2018), spanish edition
6. Gombrich, E.: The Story of Art. Phaidon Press (1995)
7. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016)
8. Knees, P., Schedl, M.: A survey of music similarity and recommendation from music context data. ACM Trans. Multimedia Comput. Commun. Appl. **10**(1), 2:1–2:21 (Dec 2013)
9. Lops, P., de Gemmis, M., Semeraro, G.: Content-based Recommender Systems: State of the Art and Trends, pp. 73–105. Springer US, Boston, MA (2011)
10. Lykourantzou, I., Claude, X., Naudet, Y., Tobias, E., Antoniou, A., Lepouras, G., Vassilakis, C.: Improving museum visitors' quality of experience through intelligent recommendations: A visiting style-based approach. In: Intelligent Environments (Workshops). pp. 507–518 (2013)
11. Mathias, M., Moussa, A., Zhou, F., Torres-Moreno, J.M., Poli, M.S., Josselin, D., El-Bèze, M., Linhares, A.C., Rigat, F.: Optimisation using natural language processing: Personalized tour recommendation for museums. In: 2014 Federated Conference on Computer Science and Information Systems. pp. 439–446. IEEE (2014)
12. McFee, B., Lanckriet, G.R.: The natural language of playlists. In: ISMIR. vol. 11, pp. 537–542 (2011)
13. McFee, B., Lanckriet, G.R.: Hypergraph models of playlist dialects. In: ISMIR. vol. 12, pp. 343–348. Citeseer (2012)
14. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015)
15. Rossi, S., Barile, F., Galdi, C., Russo, L.: Recommendation in museums: paths, sequences, and group satisfaction maximization. Multimedia Tools and Applications **76**(24), 26031–26055 (2017)
16. Semeraro, G., Lops, P., De Gemmis, M., Musto, C., Narducci, F.: A folksonomy-based recommender system for personalized access to digital artworks. J. Comput. Cult. Herit. **5**(3), 11:1–11:22 (Oct 2012)
17. Shani, G., Gunawardana, A.: Evaluating Recommendation Systems, pp. 257–297. Springer US, Boston, MA (2011)
18. Wang, Y., Stash, N., Aroyo, L., Hollink, L., Schreiber, G.: Using semantic relations for content-based recommender systems in cultural heritage. In: Proceedings of the 2009 International Conference on Ontology Patterns-Volume 516. pp. 16–28. CEUR-WS. org (2009)