Sistemas Recomendadores
IIC-3633

Recomendación basada en factores latentes
Parte 1

Esta clase

1. Concepto de representación latente
2. Reducción de dimensionalidad
3. Recomendación basada en factores latentes (SVD)

Tarea 1

Se enviará el enunciado este viernes 25 de Agosto.

Se extiende al plazo para el viernes 29 de Septiembre.

https://www.yelp.com/

Recomendación basada en factores latentes

Aprender vectores latentes que representan a los usuarios y a los items para generar la recomendación.

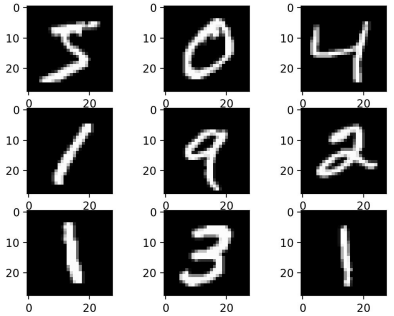
¿Qué son los factores latentes?

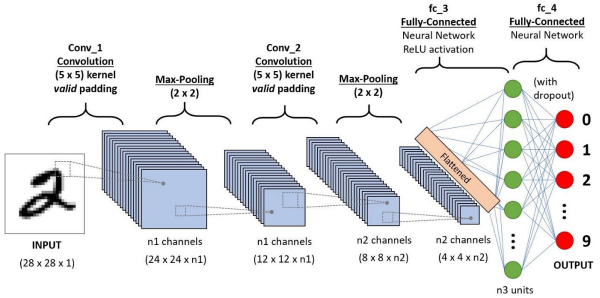
Es una forma de representar vectores de alta dimensionalidad en vectores de un tamaño definido por el usuario.

Contienen información implícita que no se puede distinguir por un humano.

En recomendación en lugar de representar usuarios e ítems como vectores de ratings , una opción es reducir dimensionalidad y representarlos como vectores latentes que van a contener implícitamente sus preferencias.

Reducción de Dimensionalidad





Reducir dimensionalidad me permite ver el panorama general de lo que aprende el modelo.

Reducción de Dimensionalidad

Contexto

- Muchos modelos de datos trabajan en dimensiones muy altas (k > 10.000)
- En alta dimensionalidad se capturan patrones importantes con mucha información, pero…
- ***Dificultad***: Maldición de la dimensionalidad (link)

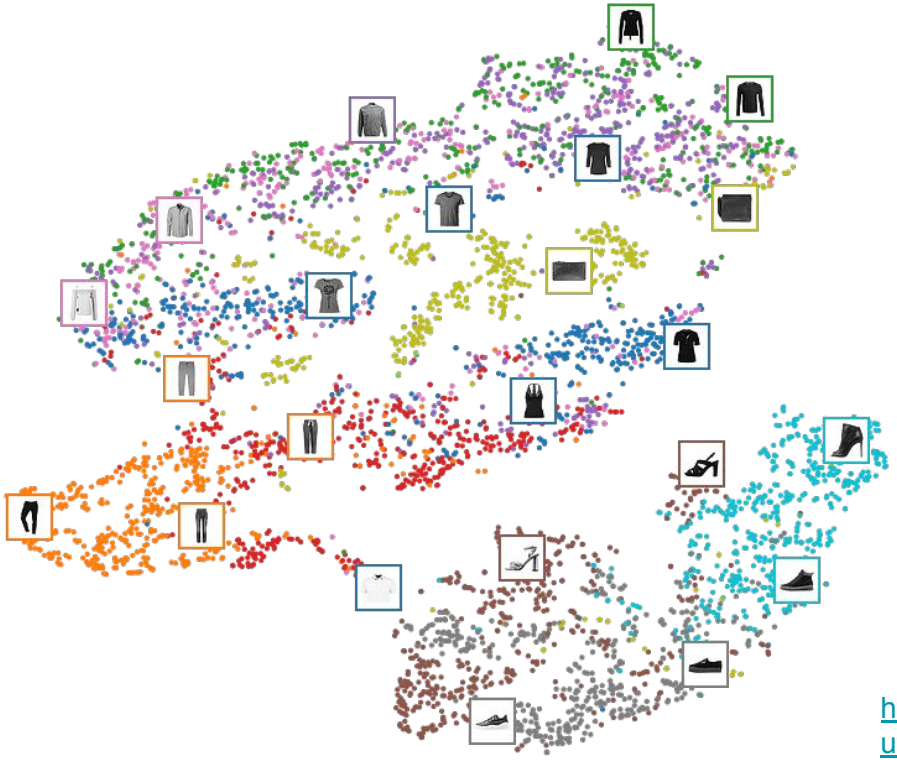
Problema

- ¿Cómo encontrar similitud entre usuarios en un espacio de menor dimensión, ***latente***?

Soluciones para reducir dimensionalidad.

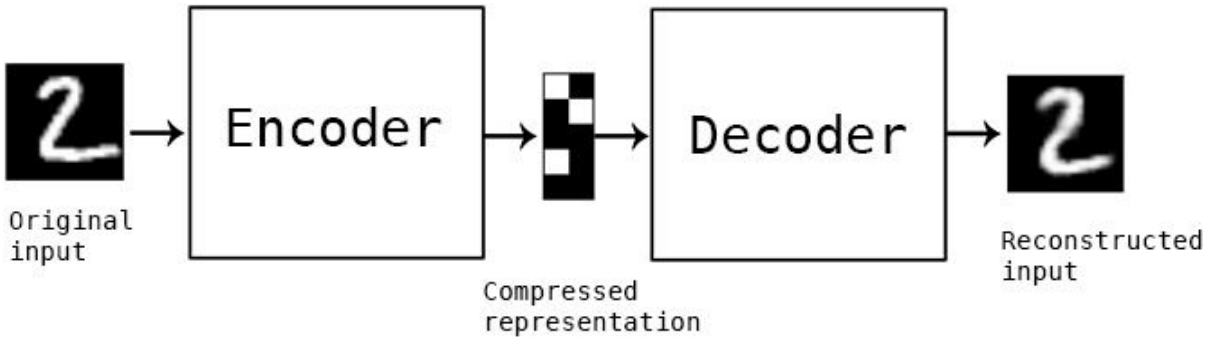
- Principal component analysis (PCA).
- t-Distributed Stochastic Neighbor Embedding (t-SNE).
- UMAP
- Factorización matricial SVD.
- Autoencoders.

Ejemplo de Reducción de Dimensionalidad



https://blog.devgenius.io/semi-supervised-learning-with-autoencoders-33f36305e816

Reducción de dimensionalidad utilizando autoencoder

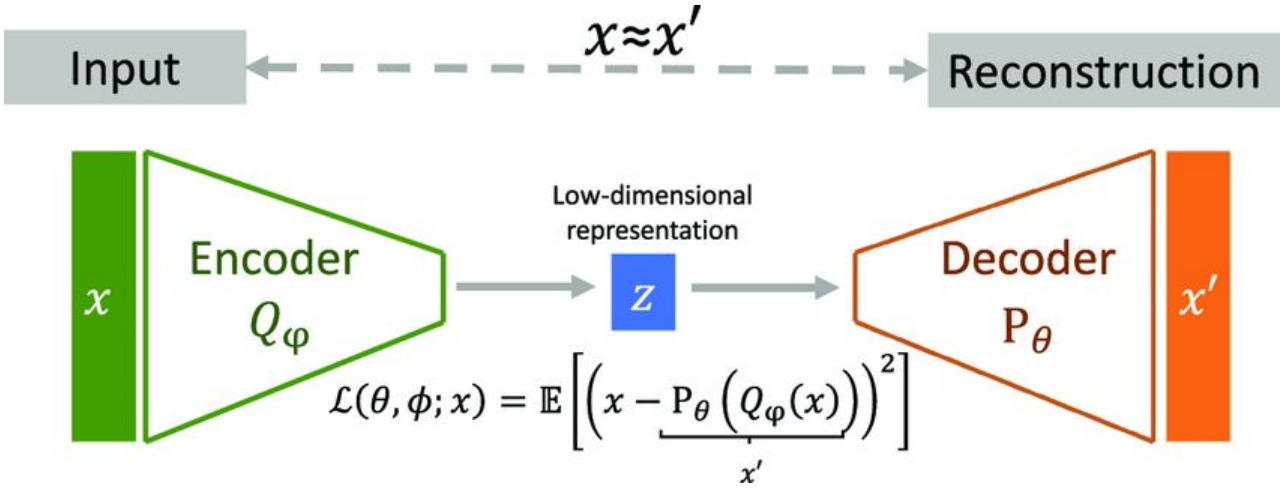


Puede recibir de input

- una matriz sparsed
- una imagen
- audio
- video
- etc…

EMBEDDING DE DIMENSIÓN REDUCIDA QUE REPRESENTA AL INPUT ORIGINAL

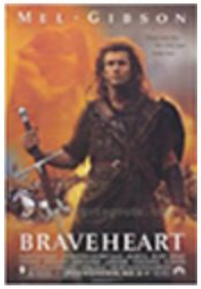
Reducción de dimensionalidad utilizando autoencoder

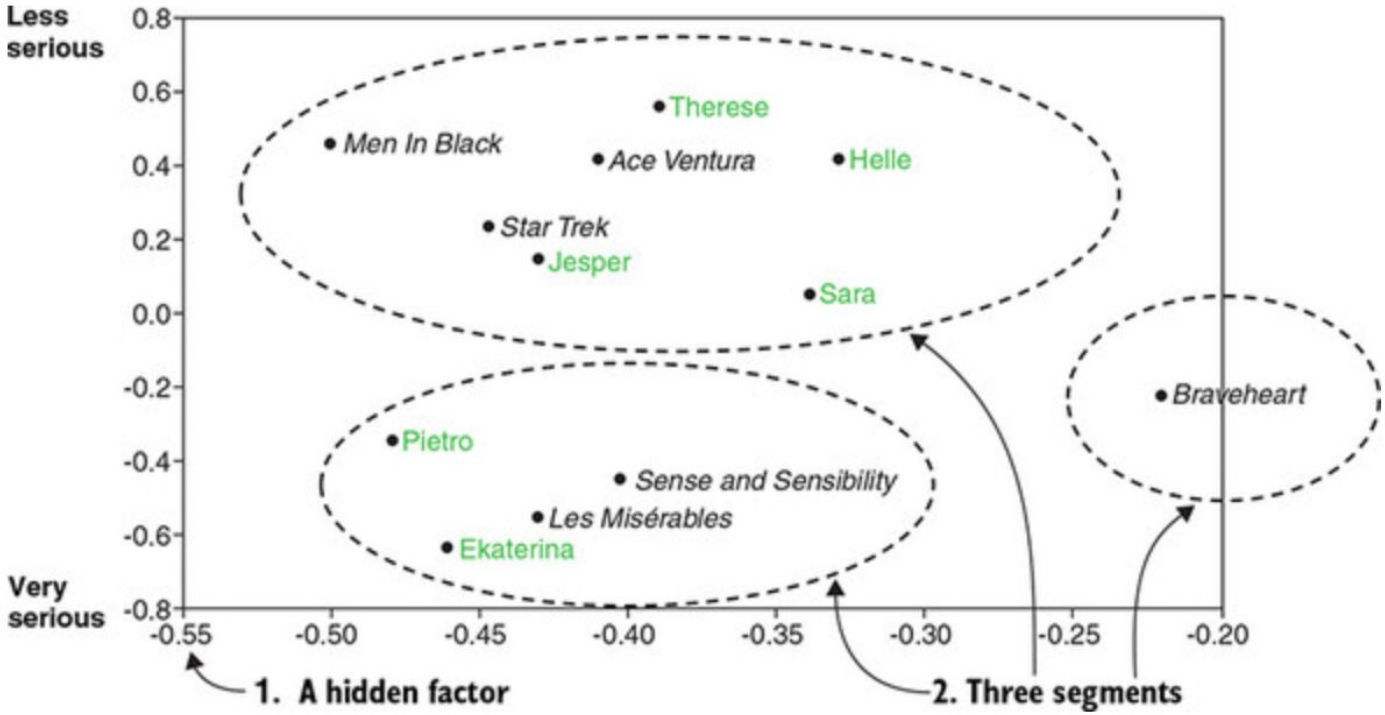


Modificar los parámetros del encoder y el decoder para que distribución de los datos sea lo más parecida posible.
KL Divergence

Reducción de dimensionalidad para recomendación

Reducir dimensionalidad de matriz de ratings para usuarios e items.

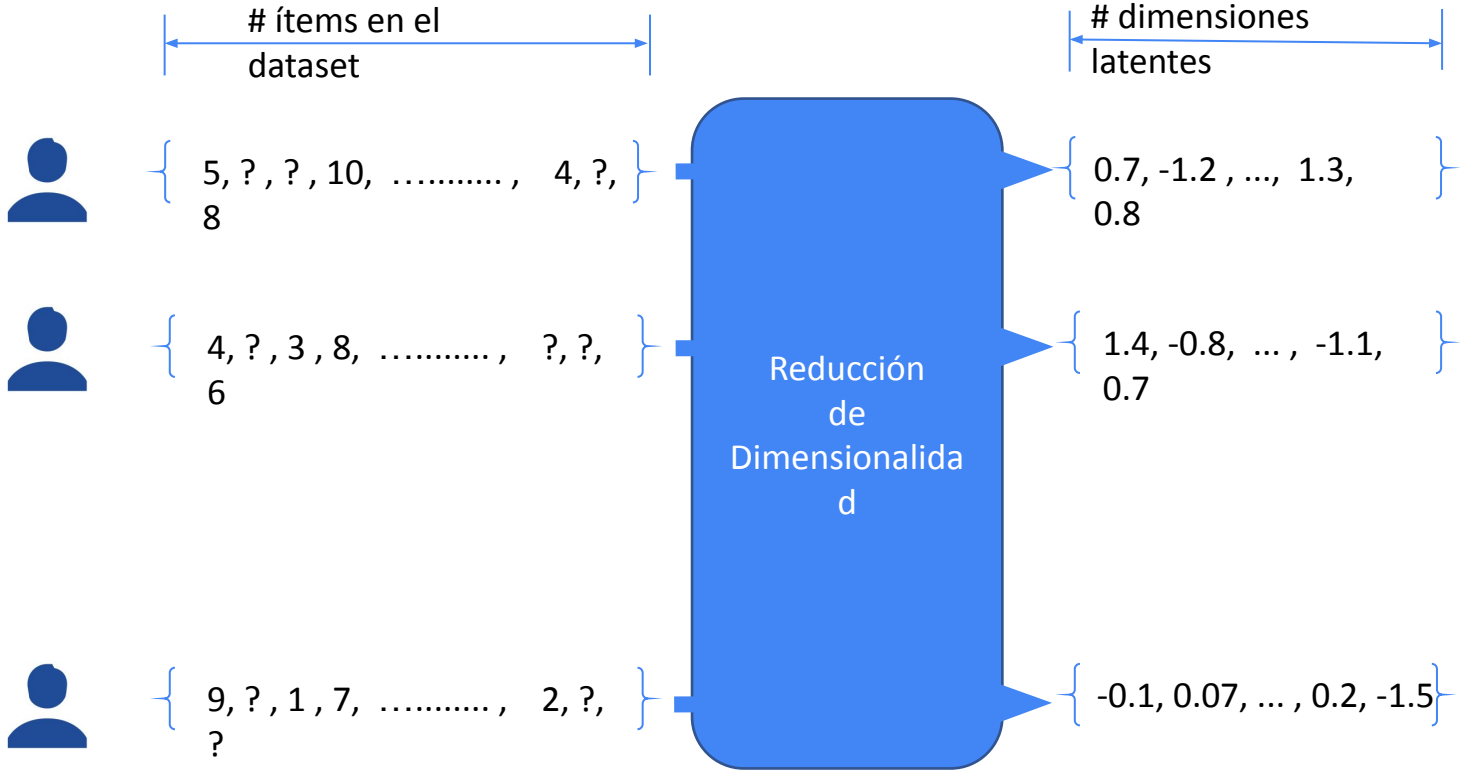
						
	Comedy	**Action**	**Comedy**	**Action**	**Drama**	**Drama**
Sara	5	3	✗	2	2	2
Jesper	4	3	4	✗	3	3
Therese	5	2	5	2	1	1
Helle	3	5	3	✗	1	1
Pietro	3	3	3	2	4	5
Ekaterina	2	3	2	3	5	5



- Eje Y muestra cuán SERIA es la película / usuario y el eje X tenemos que darle una interpretación (ej. AVENTURA)
- En este caso la película BRAVEHEART es un outlier y no se identifica con ninguno de los usuarios

Relación con Matriz Usuarios-Items

K dimensiones latentes es un meta-parámetro



Factorización Matricial - Introducción

Filtrado Colaborativo

Recomendar lo que usuarios similares han consumido

$$\hat{r}_{u,i} = \bar{r}_u + \frac{\sum_{j \in V(u)} sim(u,j)(r_{j,i} - \bar{r}_j)}{\sum_{j \in V(u)} sim(u,j)}$$

Factorización Matricial (SVD)

Genera vectores latentes por cada usuario e ítem del set de datos

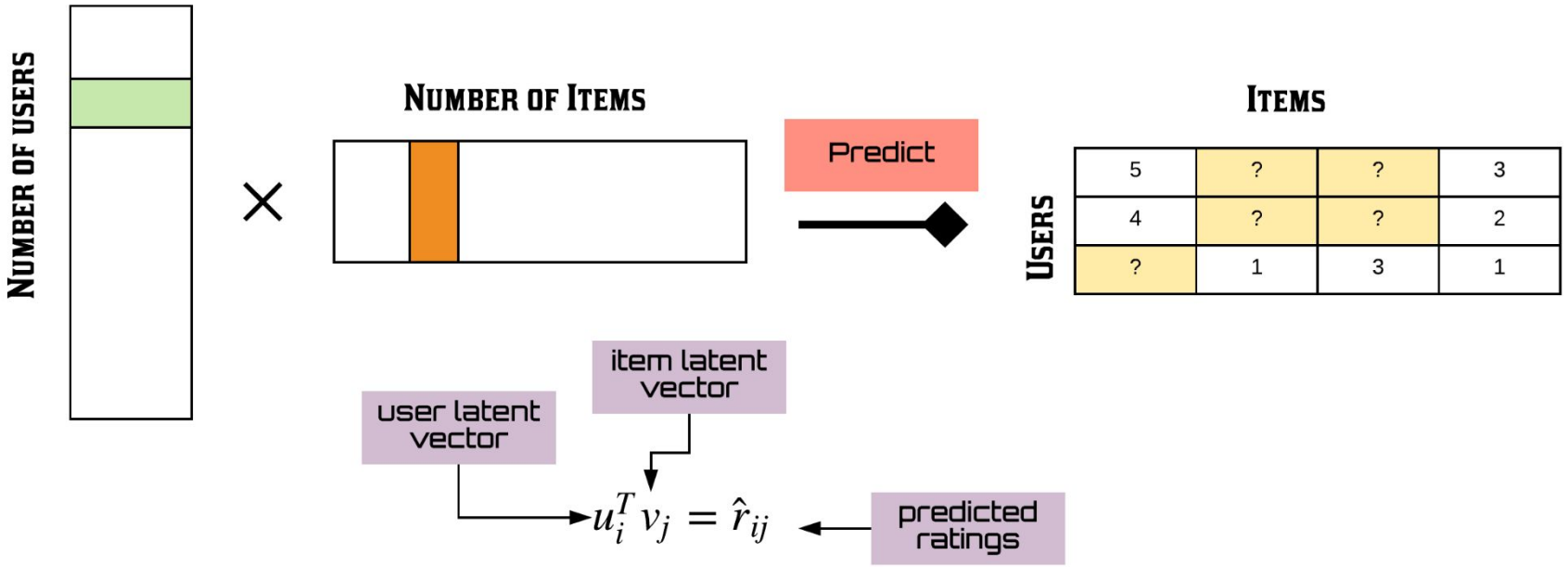
$$\hat{r}_{ui} = \vec{q}_i \cdot \vec{p}_u$$

donde

$\vec{q}_i$: vector latente para el ítem i

$\vec{p}_u$: vector latente para el usuario u

Recomendación basada en factores latentes (factorización matricial SVD).



Singular value decomposition (SVD)

SVD es un método que permite descomponer la matriz de ratings en tres matrices más simples.

Tenemos una matriz R de tamaño m x n, SVD nos permite descomponerla como el producto de tres matrices:

R = UΣV^T

- **U :** matriz m x m ortogonal (U^T U = U U^T = U).
- **Σ :** matriz m x n diagonal (con ceros fuera de la diagonal).
- **V :** matriz n x n ortogonal (V^T V = V V^T = I).
- **^T :** transposición de una matriz.

Singular value decomposition (SVD)

- Los elementos de la diagonal de **Σ** son los valores singulares de R.

- **U** y **V** son los vectores singulares izquierdos y derechos de R, respectivamente.

- Los valores singulares de Σ se ordenan en orden decreciente.

Por qué no usamos SVD para aprender los vectores de usuarios e ítems?

Las dimensiones de la matriz son demasiado grandes y es muy costoso computacionalmente

El algoritmo para hacer SVD original está hecho para una matriz que tiene todos los valores.

Necesitamos agregarle regularización para prevenir overfitting.

Por qué reducimos dimensionalidad?

Para disminuir la cantidad de computo

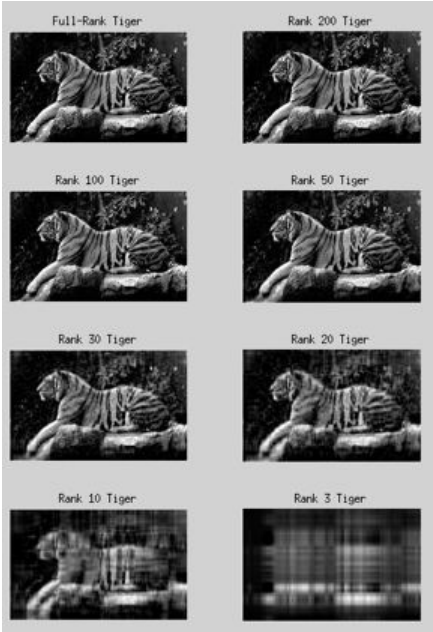
Para reducir la maldición de la dimensionalidad, ahora tendré vectores más pequeños y más densos.

En alta dimensión los vectores pueden quedar a distancias muy similares.

Pérdida de Información al reducir dimensionalidad

Imagen Original

SVD con diferentes valores de k



Factorización Matricial – Desventajas SVD

- **Pérdida de información**
- SVD genera sobre-ajuste a los datos de entrenamiento porque no tiene regularización.
- No funciona para matrices con muchos campos vacíos (sparse)

Código en python de implementación de SVD

https://colab.research.google.com/drive/1GBXpBU6f0PLzy1pke3PrNA6RI9lsHIRY?usp=sharing

Reflexión

¿Qué información captura el modelo de factorización matricial?

¿Qué representan los vectores latentes de usuario e ítems?

Factorización Matricial – Vectores Latentes

- Los vectores latentes capturan atributos NO OBSERVABLES de los ítems y si a los usuarios les gustan dichos atributos

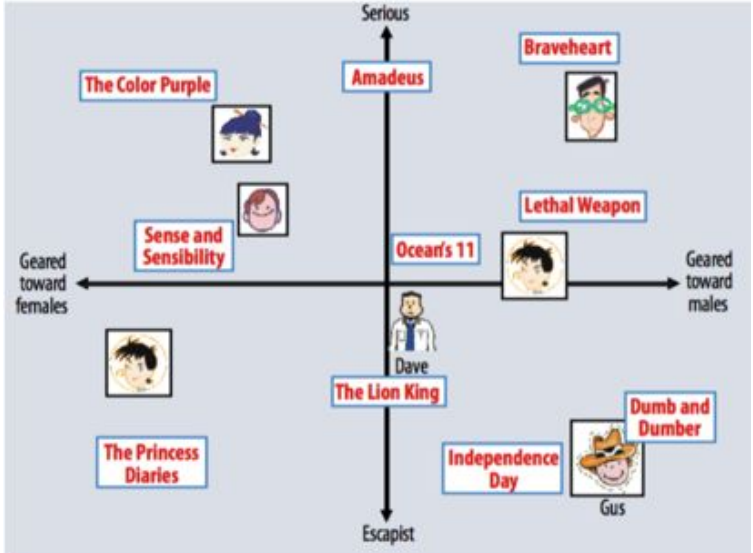
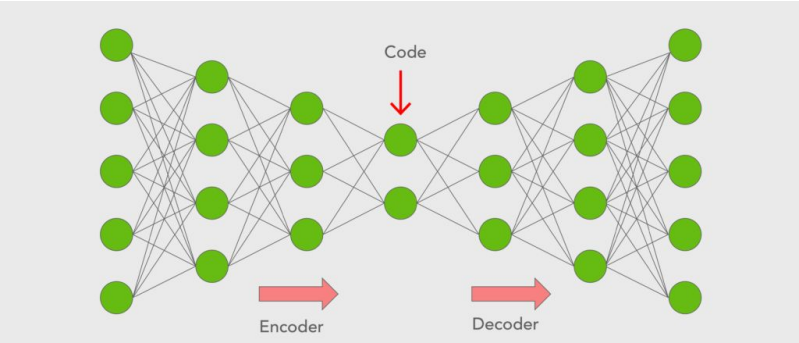


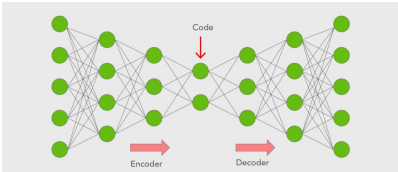
Figure 2. A simplified illustration of the latent factor approach, which characterizes both users and movies using two axes—male versus female and serious versus escapist.

Autoencoders para recomendación

- El objetivo es rellenar los ratings que faltan.
- Incluir ruido sobre la matriz de rating.
- Esconder aleatoriamente ratings en la matriz y entrenar al autenconder para predecirlos.
- Luego cuando le entregue una nueva matriz de ratings con ratings que no sabemos el auto-encoder será capaz de llenarlas.

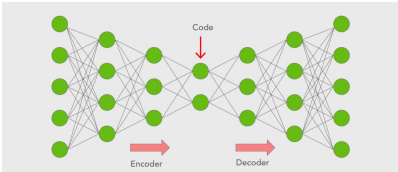


	Item 1	Item 2	Item 3
User 1	4		4
User 2		2	
User 3	4	3	1



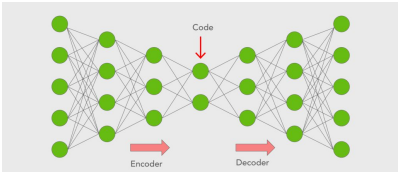
	Item 1	Item 2	Item 3
User 1	4	3	4
User 2	5	2	3
User 3	4	3	1

	Item 1	Item 2	Item 3
User 1		3	
User 2	5		3
User 3		3	



	Item 1	Item 2	Item 3
User 1	4	3	4
User 2	5	2	3
User 3	4	3	1

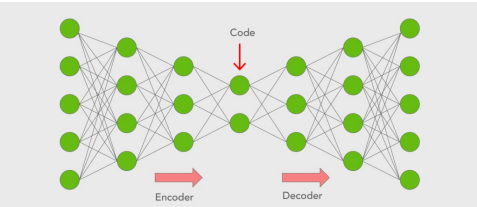
	Item 1	Item 2	Item 3
User 1		3	4
User 2			3
User 3	4	3	1



	Item 1	Item 2	Item 3
User 1	4	3	4
User 2	5	2	3
User 3	4	3	1

Recomendación con autoencoders

	Item 1	Item 2	Item 3	Item 4
User 1		3		



	Item 1	Item 2	Item 3	Item 4
User 1	**2**	3	**3**	**4**

Recomendación

1. Ítem 4 (4)
2. Item 3 (3)
3. Item 1 (2)