Apuntes Semana 14 - Unsupervised Learning

Apuntes del jueves 29 de mayo

Mariana Fernández Martínez - 2021104026

Abstract - This week highlights key advances in AI and unsupervised learning. DeepSeek V3.1 introduces major improvements in reasoning (43% boost), extended context handling (up to 1 million tokens), multilingual support (100+ languages), and reduced hallucinations, enabling accurate and versatile applications. The paper "Harnessing the Universal Geometry of Embeddings" presents vec2vec, a method for translating embeddings across models without paired data, preserving semantic and geometric structures—raising important security considerations. Additionally, core unsupervised learning concepts like clustering (e.g., K-means) and dimensionality reduction (e.g., PCA) are revisited as essential tools for pattern discovery and data simplification.

I. NOTICIA DE LA SEMANA

A. Modelo DeepSeek

DeepSeek V3.1 es la versión más avanzada del modelo de inteligencia artificial desarrollado por DeepSeek. Representa un gran avance en múltiples áreas clave de la IA, con mejoras significativas en razonamiento, manejo de contexto, precisión y soporte multilingüe.

A continuación, se destacan sus principales características:

1. **Capacidades avanzadas de razonamiento:** DeepSeek V3.1 logra hasta un 43% de mejora en tareas de razonamiento complejo frente a versiones anteriores, permitiendo resolver problemas sofisticados en áreas como matemáticas, programación y análisis científico.
2. **Procesamiento de contexto a gran escala:** Con una ventana de contexto de hasta 1 millón de tokens, el modelo puede analizar documentos extensos sin perder coherencia, como bases de código completas, artículos científicos o contratos legales.
3. **Soporte multilingüe optimizado:** Ofrece compatibilidad con más de 100 idiomas, con mejoras notables en lenguas asiáticas y de bajos recursos, alcanzando niveles cercanos a la fluidez nativa.
4. **Reducción de errores y mayor fiabilidad:** Gracias a mejoras en su arquitectura y entrenamiento, presenta un 38% menos de alucinaciones (información errónea), aumentando así la precisión y confianza en sus respuestas.

B. Paper: Harnessing the Universal Geometry of Embeddings

En el artículo se mencionan las siguientes ideas principales:

1. **Traducción sin datos emparejados**: Se propone vec 2 vec, el primer método capaz de traducir embeddings de texto de un modelo a otro sin necesidad de datos emparejados, conocimiento del encoder original, ni correspondencias predefinidas.
2. **Hipótesis de Representación Platónica Fuerte:** Los autores argumentan que los modelos de lenguaje, a pesar de tener arquitecturas y entrenamientos distintos, comparten una estructura geométrica latente universal que permite esta traducción.
3. **Preservación de semántica y geometría:** vec 2 vec no solo alinea embeddings en un espacio común, sino que preserva tanto la geometría como la semántica, permitiendo realizar tareas como inferencia de atributos e inversión (recuperación aproximada del contenido original del texto) sobre embeddings traducidos.
4. **Implicaciones de seguridad:** La capacidad de traducir y extraer información sensible de embeddings desconocidos sin acceso al encoder original tiene implicaciones serias para la privacidad en bases de datos vectoriales,

pues incluso embeddings aparentemente anónimos pueden revelar datos confidenciales.

II. UNSUPERVISED LEARNING

Set de datos compuesto solamente por el vector de features. No hay supervisión. Hay que transformar el vector de features en otro valor, usado para resolver un problema.

Reducción de dimensionalidad

- Reducción de dimensionalidad
  - Autoencoder
- Detección de Outliers
  - Evento binario
  - DBSCAN

A. Clustering

- Procedimiento para identificar grupos en los datos, basado en la similitud de los features.
- Se divide el espacio del conjunto de datos en N grupos, lo más homogéneo posible.
- No hay forma perfecta, solamente diferentes técnicas.

Múltiples definiciones de clustering

- Proceso de encontrar grupos en los datos.
- Dividir los datos en grupos homogéneos.
- Dividir los datos en grupos, donde los features de cada grupo están lo más cerca posible (similares).
- Dividir los datos en grupos, donde los features de cada grupo están lo más cerca posible (similares), y los features de diferentes grupos están lo más lejos posible.
- Las definiciones son muy amplias y pueden variar dependiendo del dominio del problema.
  - Muchísimos artículos.
- Clusters en alta dimensionalidad pueden no tener sentido para seres humanos.

B. Principios y Técnicas

Similaridad o Distancia

- Basado en medir la similaridad entre los objetos del dataset.
- Similaridad típicamente hecho con distancia.
- Objetos más cercanos en el espacio de features se consideran más similares.
- No usar datos categóricos (ejemplo: frutas).

Centralidad de cluster

- Basados en puntos centrales (centroides)
- Representa el promedio o la ubicación central de todos de los puntos en un cluster
- Algunos algoritmos como K-means calcula iterativamente.
- Otros usan los puntos centrales para unir o dividir los clusters (jerárquicos).

Minimizar una función objetivo

- La función objetivo determina la mejor disposición del clúster.
- K-means, minimiza la suma de cuadrados entre los puntos y el centroide.
- Algoritmos basados en densidad, maximizan la densidad.

Tipos de Clustering

- Jerárquicos: Forma sistemática de crear grupos de datos similares basados en una métrica.
  - Se asigna cada muestra a un separador de cluster.
  - Se agrupa cada par de clusters donde el criterio es el más pequeño.
  - Asigna para mergear el valor de similitud entre los clústeres mergeados.
  - Itera los últimos dos paso hasta tener un único cluster

- Determinístico: La muestra pertenece a un cluster o no pertenece.

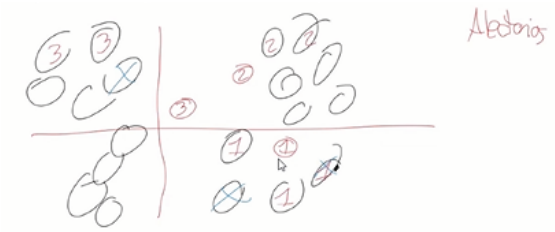
- Cluster probabilístico: Pertenece a un cluster indicando una probabilidad. Gaussian Mixture.

C. K-Means

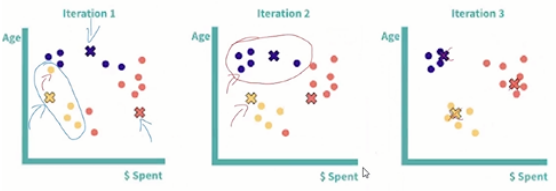
- Todas las muestras están asignadas a un cluster.
- Minimiza una función de costo: Error cuadrático entre un elemento representativo del cluster (centroid) y cada muestra del dataset.
- Proceso iterativo hasta encontrar el mínimo local.

Idea

- Mover cada centroide representativo de cada cluster, lo más cercano al centro para cada grupo.
    - El centroide puede ser un punto aleatorio del espacio o una muestra del dataset.



- Para cada muestra
    - Encontrar el centroide más cercano.
    - Asignar la muestra al cluster correspondiente.
- Recalcular los centroides
    - Average del Feature Vector de todas las muestras de cada cluster
    - Reasignar el nuevo valor a cada centroide
- Repetir

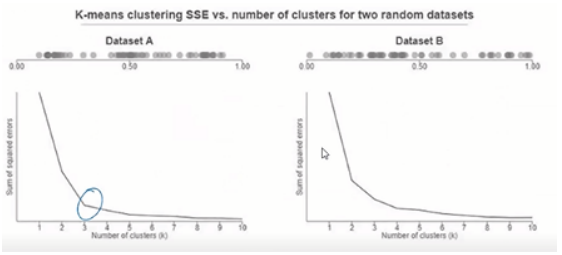


Criterio de convergencia

- Después de una iteración donde las muestras no cambiaron de cluster.
- Después de que los centroides no cambiaron.
- Después de que el valor de la función objetivo no cambia.
    - No cambia las distancias.
- Número máximo de repeticiones.

Consideraciones de K-Means

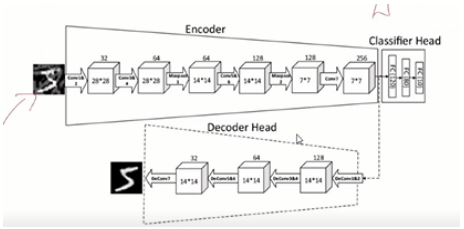
- Las *n* ejecuciones del algoritmo probablemente no den el mismo resultado.
    - Dependen de las escogencia del centroide.
- El *k* es un hiper parámetro
    - Hay algoritmos para hallar un buen K (Elbow Method), pero no hay manera perfecta.



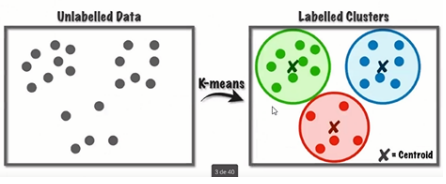
III. PRINCIPAL COMPONENT ANALYSIS

Para realizar la reducción dimensional de datasets, se aplica una transformación lineal a la matriz para proyectarla en un espacio más pequeño.

Autocoder



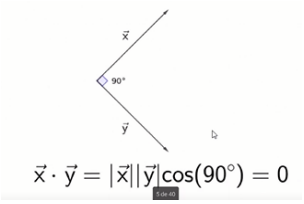
K-Means



- Reducir dimensionalidad.

- Técnica de extracción de características
    - Eliminar variables correlacionadas.
    - Solamente persevera la información importante.
- Combina variables de entrada de forma específica y elimina las menos importantes.
- Crea nuevas variables independientes / ortogonales.

Variable ortogonal



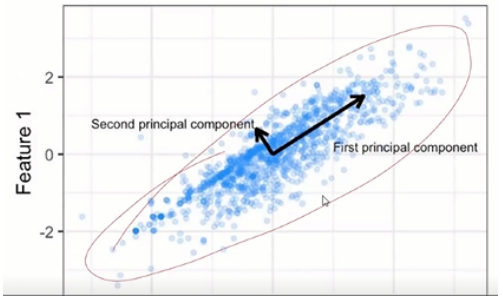
- Mis datos están proyectados a una nueva dimensionalidad.
    - Transformación Lineal.
- Maximiza la varianza de los datos.

A. ¿Cuándo usarlo?

- Muchas variables.
    - No sabemos cuáles son las más importantes.
    - No sabemos cuáles usar.
- Queremos asegurar que las variables sean independientes.
    - Se obtienen nuevas variables a partir de las originales (componentes).
    - No están mapeados a conceptos del mundo real.

Idea

- Calcular la relación de todas las variables contra todas las otras.
    - Variables equivalentes entre todos los datos no proporcionan buena información.
- Encontrar dirección de vectores y su importancia.
    - No son los features (nuevos vectores con dirección y magnitud).
    - Resultado de las comparaciones entre variables.
- Transformar los datos para alinearlos a las direcciones importantes o combinaciones de nuestras variables originales.
    - Proyección.
    - Eliminado direcciones que no son importantes.



B. Algoritmo

1. Estandarizar los datos
2. Calcular una matriz que muestra la relación de todas las variables entre todas las demás.
    a. Covarianza o Correlación (deben estar los datos estandarizados).
3. Separar matriz en dirección (eigenvector) y magnitud (eigenvalues).
    a. No cambian dirección solamente magnitud aplicando transformación lineal.
    b. Eigenvalue describe que tan importante es el componente.
4. Ordenar por los eigenvalues y elegir el top K de ellos.

C. Matriz de covarianza

- Mide la relación lineal entre todos los features.
- Cuanto se relacionan las variables unas con otras.
- Se debe usar solamente si las variables están estandarizadas.

$$\frac{1}{n-1}((\mathbf{X}-\bar{X})^T(\mathbf{X}-\bar{X})) \quad \bar{X} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

D. Matriz de correlación

- Mide la relación lineal entre todos los features

    - Igual que la matriz de covarianza
- Números acotados entre -1 y 1
    - 1 Altamente correlacionados positivamente
    - -1 Altamente correlacionados negativamente
    - 0 No existe correlación
- Si están correlacionados explican la misma característica
- No correlacionados son ortogonales.
- Tomar las varianzas de cada variable.
    - Diagonal de matriz de covarianza.
    - Calcular la desviación estándar.
- Generar matriz de multiplicación de desviaciones estándar.
- Dividir matriz de covarianza entre matriz de desviaciones estándar.

$$\Sigma = \begin{pmatrix} \mathrm{Var}(X_1) & \mathrm{Cov}(X_1, X_2) & \mathrm{Cov}(X_1, X_3) \\ \mathrm{Cov}(X_2, X_1) & \mathrm{Var}(X_2) & \mathrm{Cov}(X_2, X_3) \\ \mathrm{Cov}(X_3, X_1) & \mathrm{Cov}(X_3, X_2) & \mathrm{Var}(X_3) \end{pmatrix}$$

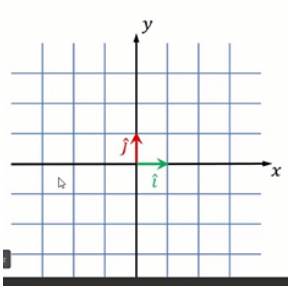
Variance-Covariance Matrix

	1	2	3	4	5
1	σ^2	COV	COV	COV	COV
2	COV	σ^2	COV	COV	COV
3	COV	COV	σ^2	COV	COV
4	COV	COV	COV	σ^2	COV
5	COV	COV	COV	COV	σ^2

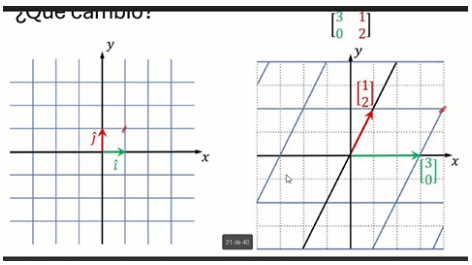
$$\frac{COV(X,Y))}{\sigma X \sigma Y}$$

E. Eigenvector y Eigenvalues

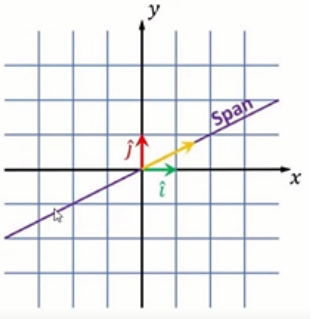
- Eigenvector: Determina la dirección en el nuevo espacio.
- En lugar de seleccionar features originales, creamos nuevos.
    - Basado en los originales.
- Vectores no nulos, que se mantienen igual después de una transformación lineal.
    - No cambian de dirección. Solo en magnitud.
- Cada eigenvector tiene un eigenvalue (factor multiplicación).
    - Indica la magnitud con la que cambia el eigenvector.
- Valores mayores de eigenvalues indican mayores niveles de importancia.
- Podríamos seleccionar los eigenvectores basados en el top K de eigenvalues.
- Estos eigenvector se llaman componentes.



- Vectores base i y j.
- Si aplicamos una transformación lineal todo el espacio va a cambiar, pero la transformación puede ser seguida basándonos en los vectores base.
- Nuestro sistema de coordenadas cambió debido a la transformación lineal.
- Todos los vectores se movieron en alguna forma, dirección, magnitud o ambos.

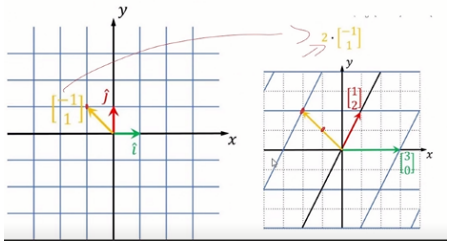


- Ahora coloquemos un vector arbitrario.
- Span: Conjunto de vectores que se refiere a todas las posibles combinaciones lineales que se pueden aplicar.
- Se puede aplicar transformaciones lineales para mantenerse en el span o no.

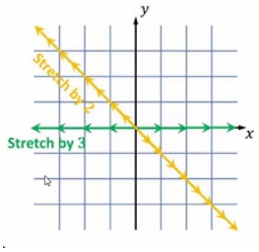


- Hay vectores que pueden ser transformados linealmente y mantienen su span.
- Ideal: Que los vectores solamente crecieran en dirección o magnitud.

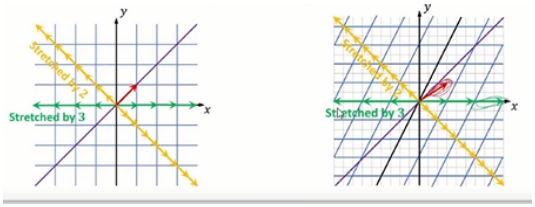
- Decimos que estos vectores están multiplicados por un escalar.
    - Acota o agranda el vector.
    - Eigenvector.



- El vector amarillo se incrementa en 2.
- Mientras el verde lo hace en 3.
- A los valores de incremento se les conoce como eigenvalues.



- Entonces, tenemos vectores que mantienen su span, y otros que no después de aplicar una transformación lineal.
    - Los eigenvectores van a mantener su span.



Definición formal

- Matriz multiplicada por un vector, que es igual al vector multiplicado por un escalar.
- Para cierta matriz A podemos encontrar un vector v tal que cuando multipliquemos la matriz A mantendrá la misma línea.
    - Span.
- Comúnmente conocemos la matriz y necesitamos encontrar λ o el vector *v*.

$$(A \cdot \vec{v}) = \lambda \vec{v}$$

- Pequeño truco para resolver la ecuación:
    - Incluir una multiplicación de matrices con Identidad, en lugar de multiplicar solamente por un escalar.
- La matriz A es nuestra matriz de correlación.

$$A\vec{v} = (\lambda I)\vec{v} \qquad \lambda \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{I}$$

¿Cuándo se puede volver vector nulo?

$$(A \cdot \vec{v}) - (\lambda I)\vec{v} = 0$$

$$(A - \lambda I)\vec{v} = 0$$

Nota: Las próximas clases podrían ser virtuales.