Selección de Instancias

Motivación

Científicos de la UC Irvine necesitan diferenciar automáticamente ciertas señales capaces de producir bosones de Higgs y señales que no. Para esto han colectado 11 millones de ejemplos con los que se desean entrenar un modelo que permita clasificar nuevas señales.

Sin embargo, temen que el tiempo de entrenamiento o clasificación sean excesivamente largos, además sospechan que cuentan con un número desmedidamente grande de instancias por lo que le piden ayuda al respecto.

Descríbanos brevemente cómo procedería.

Definición

La **selección de instancias** se ocupa de seleccionar un subconjunto de los datos de modo que el problema de Reconocimiento de Patrones se resuelva con la "misma efectividad" que se lograría de usar todos los datos.

¿Por qué?

- El número de instancias puede incidir negativamente en el tiempo de entrenamiento o respuesta.
- Datos redundantes, no aportan información.
- Puede reducir la complejidad del modelo.
- Elimina instancias ruidosas que deterioran la eficiencia.

¿Por qué?

La **selección de instancias** permite:

- que los algoritmos de aprendizaje trabajen con grandes volúmenes de datos.
- enfocarse solo en aspectos de los datos que son de interés en el dominio del problema.
- Limpiar los datos de modo que las instancias redundantes o ruidosas sean descartadas, mejorando la calidad de los datos.

Consideraciones

- La selección de instancias se diferencia de un simple muestreo en que se guía por un criterio de irrelevancia o ruido de los datos descartados.
- ¿Cómo identificar o seleccionar datos relevantes dentro de una colección inmensa de instancias?
- Balance entre el tamaño de la muestra y el desempeño del algoritmo de aprendizaje.
- ¿Selección para un problema de clasificación o agrupamiento?

Reducción alcanzada vs Desempeño del Algoritmo de Aprendizaje

- ¿qué tanto reduce el algoritmo el conjunto de entrenamiento y qué relación tiene con el desempeño del aprendizaje?

Tolerancia al ruido

- el ruido puede provocar que se descarten pocas instancias dado que se necesitan para modelar fronteras de decisión con ruido.
- el desempeño de los algoritmos de aprendizaje puede deteriorarse, en especial si las instancias ruidosas se conservan.

Complejidad Computacional

- ¿qué tan rápido es el algoritmo? ¿Cómo escala al aumentar la cantidad de datos?

Muestreo Aleatorio Simple

- seleccionar con o sin reemplazo un subconjunto del tamaño deseado.

Muestreo Aleatorio Estratificado

- el conjunto de N instancias se divide en los subconjuntos disjuntos $N_1, N_2, \ldots, N_l$ llamados "estratos". Se realiza un muestreo en cada estrato, proporcional al número de instancias en este. Permite separar conjuntos heterogéneos en subconjuntos homogéneos.

- ¿selección para un problema de clasificación o agrupamiento?

Tipo de Selección

Condensado

- descartan instancias que no influyen en la clasificación
- tienden a conservar puntos cercanos a la frontera de decisión

Edición

- llamados noise filters eliminan instancias situadas en la región otra clase
- remueven puntos en la frontera o de diferente clase que sus vecinos

Construcción de Objetos

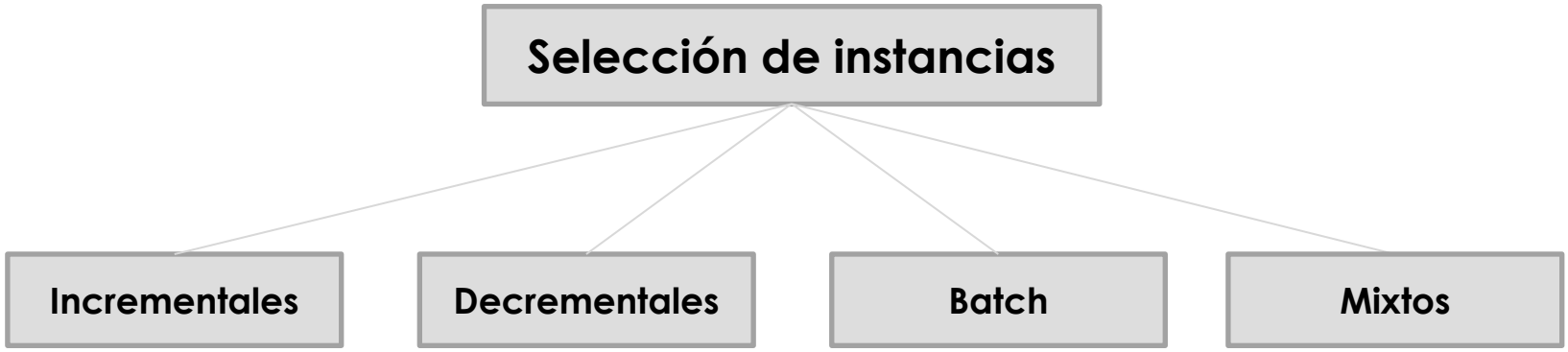
- construyen prototipos que representan a un conjunto de instancias
- cos prototipos no tienen que ser instancias del conjunto de datos original

- remueven puntos ruidosos o diferentes a sus vecinos
- suavizan las fronteras de decisión
- mantienen instancias centrales que son las "más típicas" de una clase

- asumen que los puntos interiores influencian poco las fronteras de decisión, por lo que pueden ser eliminados afectando relativamente poco el desempeño del algoritmo de aprendizaje

- las instancias seleccionadas no necesariamente pertenecen al conjunto original
- se construyen de modo que representen al conjunto de datos

Dirección de la búsqueda



-

Algunos algoritmos

Algoritmos		Referencia
Condensed Nearest Neighbor	CNN	Hart, 1968
Selective Nearest Neighbour	SNN	Ritter, 1975
Generalized Condensed Nearest Neighbor	GCNN	Chien-Hsing, 2006
Edited Nearest Neighbor	ENN	Wilson, 1972
Repeated Edited Nearest Neighbor		
All k-NN		Tomek, 1976
Multiedit		Devijver, 1980
IB2, IB3		Aha, 1991
DROP1, DROP2, …, DROP5		Wilson, 2000
Iterative Case Filtering	ICF	Brighton, 2002
C-Prunner		Ke-Ping, 2003
Support Vector Machines		Vapnik, 1995
Support Vector k-NN	SV-kNNC	Srisawat, 2006
Metodos basados en algoritmos evolutivos		Kuncheva, 1995; Cano 2003, etc.

Algunos Algoritmos Cont.

Algoritmos		Referencia
Pattern Ordered Projections	POP	Riquelme, 2003
Pair Opposite Class-Nearest Neighbor	POC-NN	Raicharoen, 2005
Generalized-Modified Chang Algorithm	GCM	Mollineda, 2002
Nearest Sub-class Classifier	NSB	Venmann, 2005
Clustering Selection	CLU	Lumini, 2006
Object Selection by Clustering	OSC	Olvera, 2007
Weighting Prototypes	WP	Paredes, 2000
Prototype Selection by Relevance	PSR	Olvera-López, 2008

Dirección de la búsqueda

Incrementales:

- comienzan por un conjunto vacío S y añaden instancias si estas cumplen un criterio determinado.
- el orden en que se analizan las instancias es muy importante.
- permite analizar nuevas instancias sin rehacer todo el proceso.

Decrementales:

- remueven instancias del conjunto original hasta que se cumplan ciertos criterios de parada.
- sensible, aunque en menor medida al orden en que se analizan las instancias.
- en general son computacionalmente más costosos que los algoritmos incrementales.
- notar que, tras una eliminación, el conjunto de datos cambia.

Dirección de la búsqueda

Batch:

- Similar a los incrementales, pero no se borran las instancias inmediatamente, solo de dejan marcadas para su eliminación al concluir la iteración

- Puede borrar grupos completos de instancias

Híbridos:

- Comienzan por un conjunto preseleccionado de instancias de donde añaden o remueven instancias que cumplan determinadas condiciones

Criterios de consistencia

Respecto al conjunto de entrenamiento:

- T´ es consistente respecto a T si un clasificador entrenado con T´ clasifica correctamente todos los objetos de T.

Respecto a la frontera de decisión:

- T´ es consistente respecto a T si determina exactamente la misma frontera de decisión, es decir si cada instancia x obtiene la misma clasificación independientemente del conjunto seleccionado para el entrenamiento.

Edited Nearest Neighbor

- Obtiene un subconjunto $S \subseteq T$ resultado de eliminar de T todas las instancias incorrectamente clasificadas utilizando la regla k-NN sobre el conjunto T
- Comienza con un conjunto $V = \emptyset$ donde se colocarán las instancias a eliminar
- Cada instancia t_i en T se clasifica utilizando la regla k-NN tomando $T - t_i$ como conjunto de entrenamiento
  - Si la clasificación es incorrecta, $V = V \cup t_i$
- Hacer $T = T - V$

Condensed Nearest Neighbor

- Obtiene un subconjunto $S \subseteq T$ tal que, cada miembro en T tiene como vecino más cercano en S una instancia de su propia clase
- Comienza colocando en S una instancia aleatoria por cada clase
- Cada instancia t_i en T se clasifica utilizando la regla 1-NN tomando S como conjunto de entrenamiento
  - Si t_i la clasificación es incorrecta, $S = S \cup t_i$

Algoritmo de Chang

- Obtiene un conjunto S resultado de fusionar pares de instancias en T que son de la misma clase y que al ser reemplazadas por un prototipo que las representa, no altera el resultado de la clasificación
- Comienza con un conjunto V que contiene una instancia aleatoria de T
- Buscar el par de instancias $t_i \in V$ y $x_j \in T$ tal que $d(t, t_j)$ sea mínima
- Si x_i, x_j son de diferente clase, $V = V \cup t_j$, $T = T - t_j$, si no:
  - Obtener prototipo $t_j^i = f(t_i, t_j)$, donde f es una función que permite obtener un prototipo representativo de t_i, t_j
  - Sustituir t_i, t_j por t_j^i de T y clasificar el resto de las instancias en T
  - Si no se cometen nuevos errores de clasificación:
    - $V = V \cup t_j^i$ - t_i , $T = T - t_j$
  - En otro caso, $V = V \cup t_j$, $T = T - t_j$,

Learning Vector Quantization (LVQ)

Obtiene un conjunto S de instancias que representa a T

- Comienza por un conjunto $S = \{s_1, s_2, \dots, s_n\}$ de instancias, que pueden ser elegidas aleatoriamente de entre T
- Para cada instancia t_i en T buscar s_k tal que $\mathrm{d}(s_k, t_i)$ sea mínima
  - Si s_k y t_i tienen igual clase, $s_k = s_k + \alpha|t_i - s_k|$
  - Si s_k y t_i tienen diferente clase, $s_k = s_k - \alpha|t_i - s_k|$
- El paso anterior se repite hasta que se cumplan ciertas condiciones de parada, por ejemplo, haber efectuado un número dado de iteraciones.

$0 < \alpha \leq 1$

¿Selección de Instancias en Procesamiento del Lenguaje Natural?

Estrategias de Consulta

Uncertainty Sampling

- se seleccionan las instancias que le provocan mayor incertidumbre al modelo.

Query-By-Committee

- se tienen varios modelos, se seleccionan las instancias entre las que existe menor acuerdo entre los modelos.

Expected Model Change

- se seleccionan las instancias que conduzcan a un mayor cambio en el modelo, por ejemplo, mayor variación de los pesos en la red neuronal.

Estrategias de Consulta

Expected Error Reduction

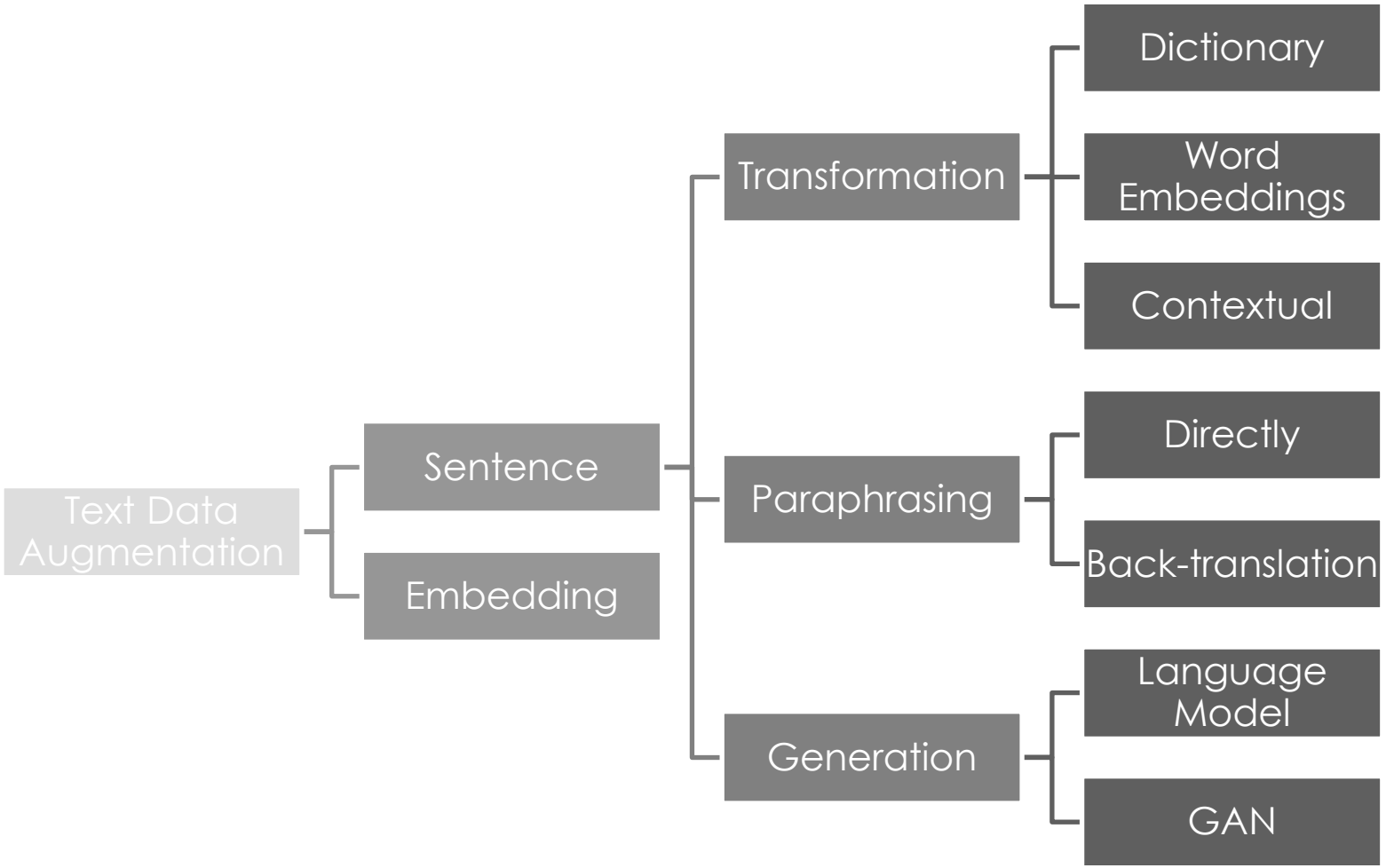
- se seleccionan las instancias que conduzcan a una reducción en el error estimado del modelo.

Variance Reduction

- se seleccionan las instancias que conduzcan a una reducción en el en la varianza del modelo.

Density-Weighted Methods

- se seleccionan las instancias que sean representativas de la distribución de los datos.



Sentence	Embedding
• Transforman, manipulan o generan directamente el texto. Por ejemplo, sustituyen palabras en el texto original.	• No operan directamente en el texto, sino en el espacio donde está representado el texto, ejemplo, embedding.

Transformation

- operaciones de reemplazo, inserción, intercambio o borrado de palabras.
- recursos como WordNet, Word Embeddings, etc.
- modelos para predecir reemplazos basados en el contexto.

Paraphrasing

- texto con contenido semántico similar,
- generación directa, back-translation, etc.

Generation

- ejemplos totalmente nuevos, herramientas como LLMs.

Fin