Universidad Simón Bolívar
Decanato de Estudios Profesionales
Coordinación de Ingeniería de la Computación

@títuloProyecto

Por:
Alejandro Flores V.

Realizado con la asesoría de:
Emely Arráiz B.

PROYECTO DE GRADO
Presentado ante la Ilustre Universidad Simón Bolívar
como requisito parcial para optar al título de
Ingeniero de Computación

Sartenejas, septiembre de 2014

UNIVERSIDAD SIMÓN BOLÍVAR
DECANATO DE ESTUDIOS PROFESIONALES
COORDINACIÓN DE INGENIERÍA DE LA COMPUTACIÓN

ACTA FINAL PROYECTO DE GRADO

@tíTULOPROYECTO

Presentado por:
Alejandro Flores V.

Este Proyecto de Grado ha sido aprobado por el siguiente jurado examinador:

Emely Arráiz B.

@jurado1

@jurado2

Sartenejas, @día de @mes de @año

Resumen

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Palabras clave: @palabra1, @palabra2, @palabra3.

Agradecimientos

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Índice general

Índice de figuras

Índice de Tablas

Índice de algoritmos

Acrónimos y Símbolos

KDD	Knowledge Discovery in Databases
MD	Minería de Datos
SI	Selección de Instancias
NN	Nearest Neighbor

$\in$	Relación de pertenencia, «*es un elemento de*»

Dedicatoria

A @personasImportantes, por @razonesDedicatoria.

Introducción

El avance de la ciencia y la tecnología durante las últimas décadas ha traido como consecuencia un aumento sin precedentes en la candidad de datos generados y recopilados por la actividad humana. El *Proyecto Genoma Humano*, el *Instituto SETI* y el *Gran Colisionador de Hadrones*, tienen algo en común: generan una enorme cantidad de datos, por lo que resulta imposible usarlos y mucho menos analizarlos de forma tradicional.

Por esta razón, nuevos cambos de estudio, como el Descubrimiento de Conocimiento en Bases de Datos (*KDD*) y Minería de Datos (*DM*), emergen para afrontar el creciente problema que se genera al intentar usar y analizar enormes cantidades de datos.

Bajar complejidad, disminuir los datos.

Capítulo 1

Selección de Instancias

En este capítulo se describe el proceso de reducción de datos y sus diferentes estrategias. En particular, se hace especial énfasis en el problema de *Selección de Instancias*: se define formalmente, se describen sus principales características, y se realiza un breve análisis del estado del arte.

1.1. Reducción de Datos

Como parte del proceso de "*Knowledge Discovery in Databases*" (*KDD*), la fase de *Preprocesamiento de los Datos* juega un rol fundamental para la aplicación efectiva de técnicas de *Minería de Datos* (*MD*). Una de las estrategias de mayor uso durante la fase de preprocesamiento es la de *Reducción de Datos*.

El problema de *Reducción de Datos* consiste en decidir qué datos deben ser utilizados durante la aplicación de algoritmos de *MD* con el objetivo de construir modelos representativos de los datos originales. Dicha decisión debe basarse en la relevancia de los datos con respecto a los objetivos que se persiguen, o inclusive, por limitaciones técnicas. En términos prácticos, la importancia del problema de *Reducción de Datos* radica en los siguientes factores: *a*) *Tiempo y Espacio*: Mientras mayor sea el número de datos a utilizar, mayor será el espacio necesario para almacenarlos y el tiempo requerido para analizarlos. *b*) *Sensibilidad al ruido*: Al aumentar el número de instancias en el conjunto de datos, también lo hace la probabilidad de aparición de datos atípicos, inconsistentes o redundantes. Su

eliminación se vuelve necesaria para evitar un impacto negativo en los modelos de representación creados a partir de los datos.

En función de estos criterios, y basados en la definición de los datos, se han formulado diferentes estrategias para llevar a cabo la fase de reducción. En los procesos de *KDD*, el conjunto de datos está definido en función de un conjunto de clases Ω y un conjunto T de n observaciones de un evento, cada observación con m mediciones, donde:

Definición 1. Una **instancia** t_i (con $i = 1 \dots n$) es una observación del evento; donde $t_i = (v_{i,1}, v_{i,2}, \dots, v_{i,m})$ es una tupla de m valores/mediciones (un punto en un espacio m-dimensional). Adicionalmente, cada instancia en t_i pertenece a la *clase* $\omega_{t_i} \in \Omega$.

Definición 2. Un **atributo** p_j (con $j = 1 \dots m$) define el conjunto de mediciones «*de un mismo tipo*» para todas las observaciones, *i.e.* $p_j = \{v_{i,j} \mid i = 1 \dots n\}$. Cada atributo puede presentarse en diferentes formatos: *nominales*, *discretos*, o *continuos*.

A continuación se presentan las estrategias de *Reducción de Datos* más estudiadas en la literatura:

- **Selección de Instancias** [BL97, LM02]
  Busca la reducción del conjunto de datos mediante la selección de un subconjunto de instancias, de forma tal que dicho subconjunto conserve las capacidades de representación del conjunto original.
  La sección 1.2 está dedicada a describir esta estrategia en amplitud.

- **Selección de Atributos** [BL97, LM98]
  Esta técnica permite eliminar atributos del conjunto de datos original, que no contribuyen (o que influyen negativamente) a la construcción de un modelo representativo.

- **Discretización de Atributos** [FI93, LHTD02]
  Esta estrategia busca convertir atributos *continuos* en *discretos* (cuantificando el espacio de posibles valores), o disminuir el número de valores *discretos* (combinando valores adyacentes).

1.2. Selección de Instancias

Dado un conjunto inicial de instancias $T = \{t_i \mid i = 1 \dots n\}$ donde $t_i = (v_{i,1}, v_{i,2}, \dots, v_{i,m})$ y $\omega_i \in \Omega$ (siendo Ω el conjunto de posibles clases para las instancias en T), el problema de *Selección de Instancias* (*SI*) consiste en seleccionar un $R \subseteq T$ que mantenga (o mejore) la capacidad de representación del conjunto original T.

Más aún, este problema puede ser formulado como un *problema de optimización*, donde se busca el $R^* \subseteq T$ de menor cardinalidad, que mantenga (o mejore) la capacidad de representación del conjunto original.

En particular, la literatura se ha enfocado en la aplicación del problema de *SI* para su uso en clasificadores [GK14, Tou02]. El subconjunto seleccionado se usa como conjunto de entrenamiento, en base al cuál el clasificador estima la clase $\hat{\omega}$ de instancias previemente desconocidas. En este sentido, el problema de optimización de *SI* busca conseguir un $R^* \subseteq T$ *consistente* y de cardinalidad mínima, donde:
Definición 3. Un conjunto R es **consistente** con T, *si y solo si* toda instancia $t \in T$ es clasificada correctamente (*e.i.* $\hat{\omega}_t = \omega_t$) mediante el uso de un clasificador M y las instancias en R como conjunto de entrenamiento.

La complejidad del problema de selección ha sido estudiada por diferentes autores: *Bien* y *Tibshirani* [BT12] describen la reducción del problema de *SI* al problema de *Conjunto de Cobertura* ("*Set Cover*" en inglés), cuya versión de optimización es NP-Dura. Adicionalmente, *Wilfong* [Wil91] y *Zukhba* [Zuk10] muestran que el problema de selección es NP-Completo.

En general, la literatura relacionada con el problema de *SI* se ha enfocado en el uso de clasificadores k-NN por su simplicidad, y sobretodo, por su capacidad de representación de modelos sin información adicional sobre la distribución de los datos. El caso particular del problema de *SI* para su uso con clasificadores k-NN también es conocido como *Selección de Prototipos* (*SP*). A continuación se describen los clasificadores NN.

1.2.1. Regla del Vecino Más Cercano (NN)

Inicialmente descrita por *Fix* y *Hodges* [FH51], la regla del *Vecino Más Cercano* ("*Nearest Neighbor*", *NN*) es una regla de inferencia basada en la idea de

que instancias con atributos similares (cercanas en un espacio de m dimensiones) tienden a compartir la misma clase. La regla NN estima la clase $\hat{\omega}_x$ de un punto x en un espacio m-dimensional, dado un conjunto T de instancias de entrenamiento y una función de distancia φ entre dos puntos en dicho espacio:

$$\hat{\omega}_x = \omega_{t^*} \ , \ t^* = \underset{t \in T}{\arg\min}\, \varphi(t, x) \tag{1.1}$$

La generalización de la regla de inferencia NN se conoce como el clasificador k-NN: dado un $k \in \mathbb{N}$, se estima la clase $\hat{\omega}_x$ de un punto x en función a la clase de las k instancias más cercanas a x. En general, se usa la estrategia del «*voto de la mayoría*», asignando la clase más común entre las k instancias más cercanas. En particular, el clasificador 1-NN corresponde a la regla NN.

k-NN es un clasificador no paramétrico, de *aprendizaje perezoso* (debido a que la etapa de aprendizaje consiste en guardar el conjunto de entrenamiento), caracterizado por su sencillez en términos de implementación. Esa simplicidad, y su probada utilidad para numerosas aplicaciones, han hecho del clasificador k-NN uno de los más estudiados en la literatura.

Uno de los trabajos de mayor relevancia es el de *Cover* y *Hart* [CH67], quienes mostraron que cuando el número de instancias de entrenamiento tiende a infinito, el clasificador k-NN garantiza un error no mayor al doble de la tasa de error de Bayes: la menor tasa de error posible para un clasificador dado. Adicionalmente, probaron que para un conjunto de entrenamiento de cardinalidad finita, el clasificador 1-NN es admisible dentro de la clase de clasificadores k-NN: *e.i.* No existe $k > 1$ tal que k-NN tenga menor probabilidad de error frente a 1-NN, para toda posible distribución de los datos.

Adicionalmente algunos trabajos en geometría computacional han contribuido significativamente en la comprensión del problema. En este sentido, la regla NN para espacios euclideanos puede definirse de forma alternativa en función de *Diagramas de Voronoi* [Vor08]; donde el espacio $\mathbb{R}^m$ se encuentra particionado en *Celdas de Voronoi*, cada una definida por una instancia $t \in T$ donde t es el *vecino más cercano* para todos los puntos dentro del espacio dentro de dicha celda (ver Figura 1.1). Esto ha permitido el desarrollo de nuevos enfoques para la búsqueda de vecinos más cercanos basados en *Diagramas de Voronoi*, como el descrito por *Kolahdouzan* y *Shahabi* [KS04].

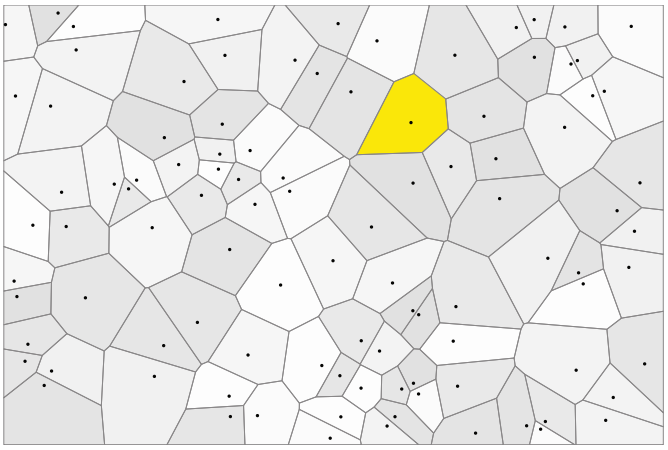

FIGURA 1.1: Diagrama de Voronoi para instancias en un espacio $\mathbb{R}^2$. En amarillo la Celda de Voronoi de un punto $t \in T$, representando el espacio de puntos para los que t es su *vecino más cercano*.

Similarmente, esta relación ha permitido avances importantes en términos de complejidad. En particular, mediante el uso de *kd*-trees [Ben75] (árboles de búsqueda binaria en múltiples dimensiones) se ha logrado disminuir la complejidad en tiempo de clasificación, de $\mathcal{O}(n)$ (de un enfoque "ingenuo" revisando todas las instancias) a $\mathcal{O}(\log n)$, a costas de un aumento en el tiempo necesario para el entrenamiento del clasificador: de $\mathcal{O}(1)$ a $\mathcal{O}(n \log n)$, el tiempo necesario para la construcción del árbol.

Sin embargo, los clasificadores k-NN presentan ciertas propiedades desalentadoras; el problema de conseguir el vecino más cercano de un punto dado, requiere –en cualquiera de los casos– almacenar todas las instancias de entrenamiento: *e.i.* $\mathcal{O}(n)$ en espacio. Adicionalmente, trabajos más recientes [KL04] muestran que en espacios euclideanos de altas dimensiones la búsqueda del vecino más cercano requiere $\mathcal{O}(n)$ en tiempo: un fenómeno conocido como la «*maldición de la dimensionalidad*» ("*curse of dimensionality*" en inglés). Finalmente, según *Shwartz* y *David* [SSBD14] los clasificadores NN tienden a sobreajustar el modelo con respecto al conjunto de entrenamiento (*overfitting* en inglés); efecto que puede mitigarse aumentando el k del clasificador [DGKL94, SSBD14], y eliminando instancias del conjunto de datos [GKK13].

1.2.2. Definiciones relevantes

A continuación se definen algunos conceptos relevantes para la descripción de métodos de selección de instancias. Dado un conjunto de instancias $Q \subseteq T$:

Definición 4. Los **asociados** en Q de una instancia t son aquellas instancias en Q para las cuales t pertenece a su conjunto de k instancias más cercanas:

$$asociados_Q(t) = \{q \in Q \mid t \in kNN(q)\} \tag{1.2}$$

Definición 5. Los **enemigos** en Q de una instancia t son aquellas instancias en Q con una *clase* diferente a la *clase* de t:

$$enemigos_Q(t) = \{q \in Q \mid \omega_q \neq \omega_t\} \tag{1.3}$$

1.3. Algorítmos de aproximación para Selección de Instancias

Debido a la complejidad del problema de *SI*, la literatura se ha enfocado en la definición de heurísticas para conseguir soluciones aproximadas. De nuevo, el uso de clasificadores k-NN es una práctica extendida a lo largo de estos trabajos, por lo que las caracteristicas de la regla NN han servido para el desarrollo de muchos métodos de selección. Sin embargo, nuevos estudios se han realizado basándose en heurísticas diferentes con el fin de desarrollar

1.3.1. Métodos basados en la regla NN

- *Condensed Nearest Neighbor* (CNN) [Har68]
  Inicialmente el conjunto R se inicializa con una instancia cualquiera. Luego se itera sobre cada instancia $t \in T$; si t no es clasificada correctamente usando R, t se agrega a R. CNN reduce considerablemente el conjunto de datos, pero no asegura un conjunto consistente ni mínimo, pues depende del orden en el que son revisadas las instancias en T.

- *Edited Nearest Neighbor* (ENN) [Wil72]
  Comienza con $R = T$. Luego itera sobre las instancias en R; aquellas que no sean bien clasificadas usando R son eliminadas. Tiende a eliminar instancias ruidosas o cercanas a los bordes de decisión. Sin embargo, depende del orden en que itera sobre las instancias, y presenta bajas tasas de reducción dado que mantiene puntos internos.

- *Repeated Edited Nearest Neighbor* (RENN) [Wil72]
  Aplica ENN al conjunto de datos R (inicialmente $R = T$) hasta que no ocurran cambios en R. Amplía la distancia entre clases y "suaviza" los bordes de decisión.

- *Reduced Nearest Neighbor* (RNN) [Gat72]
  RNN extiende a CNN, usandola como solución inicial $R = R_{CNN}$. Luego, itera sobre cada instancia $t \in R$: si todas las instancias en T son correctamente clasificadas usando $R \setminus \{t\}$, se elimina t de R. En caso contrario, se mantiene R y continua la iteración. La precisión de RNN puede mejorar respecto a CNN, pero es más costoso y su consistencia depende de la consistencia del conjunto resultante de CNN y del orden en que se iteren las instancias en R.

1.3.2. Métodos basados en eliminación ordenada

- *Decremental Reduction Optimization Procedure 1* (DROP1) [WM97]
  Comienza con una solución inicial $R = T$. Itera sobre cada instancia $t \in R$: si todos sus *asociados* en R son correctamente clasificados con $R \setminus \{t\}$, t se elimina de R. Reduce considerablemente el conjunto de datos inicial, pero obtiene baja precisión de clasificación, y el subconjunto resultante depende del orden en que se iteró sobre T.

- *Decremental Reduction Optimization Procedure 2* (DROP2) [WM97]
  Es una mejora sobre DROP1 en la cuál se elimina una instancia t cuando todos sus *asociados* en T son clasificadas correctamente usando $R \setminus \{t\}$. Además, DROP2 ordena las instancias con respecto a la distancia de su *enemigo* más cercano, en un intento de eliminar primero instancias centrales, y luego los puntos en los bordes de decisión.

- *Decremental Reduction Optimization Procedure 3* (DROP3) [WM97]
  Dado que el orden en que se iteran las instancias en DROP2 se ve alterado por puntos ruidosos, DROP3 filtra instancias ruidosas antes de ordenar el conjunto de entrenamiento.

1.3.3. Métodos basados en muestreo aleatorio

- *Random Mutation Hill Climbing* (RMHC) [Ska94]
  Se selecciona un subconjunto de instancias aleatorias R de tamaño fijo. En cada iteración el algoritmo intercambia una instancia en R por una en $T \setminus R$; si el cambio mejora la precisión, se mantiene, en caso contrario se deshace.

1.3.4. Métodos basados en metaheurísticas

Las metaheurísticas son métodos de búsqueda estocástica de propósito general, usadas para encontrar soluciones óptimas o casi óptimas a problemas de optimización combinatoria. Por esta razón, muchos trabajos se han enfocado en el uso de estas técnicas para conseguir soluciones al problema de *SI*.

Algunos de los primeros trabajos se enfocaron en adaptar el algoritmo de *Búsqueda Tabú* para solucionar el problema de *SI*; en particular, los estudios de *Cerverón et al.* [CF01] y *Zhang et al.* [ZS02] describen dos enfoques diferentes de modificación del algoritmo.

Sin embargo, la mayoría de los estudios se han enfocado en el uso de *Algoritmos Evolutivos* (*AE*), adaptándolos para la búsqueda de soluciones al problema de selección. Entre ellos destaca el trabajo realizado por *Cano et al.* [CHL03]; un completo estudio comparativo entre algoritmos "tradicionales" de *SI* y adaptaciones de *Generational Genetic Algorithm* (GGA), *Steady-State Genetic Algorithm* (SGA), *CHC Adaptive Search Algorithm* (CHC) y *Population-Based Incremental Learning* (PBIL). Con este estudio resulta evidente la utilidad de los *AE* frente a los algoritmos tradicionales de *SI* en función de la capacidad de reducción y precisión de los conjuntos seleccionados.

Existen también otras adaptaciones y modificaciones sobre *AE*, entre los que destacan: *Estimation of Distribution Algorithm* (EDA) [SLI+01], *Intelligent Genetic Algorithm* (IGA) [HLL02], *Steady-State Memetic Algorithm* (SSMA) [GCH08] y *Genetic Algorithm* [GPY08] basado en Error Cuadrático Medio, *Clustered Crossover* y *Fast Smart Mutation* (GA-MSE-CC-PSM).

1.3.5. Criterios de comparación

Para comparar métodos de *SI* se consideran una serie de criterios usados para evaluar las ventajas y desventajas de cada algoritmo. A continuación se describen los factores más relevantes:

- *Reducción*: El objetivo principal de métodos de *SI* es el de reducir número de instancias del conjunto de datos. Esto no solo disminuye el espacio necesario para almacenar los datos, sino que acelera el proceso de clasificación.
- *Precisión*: Un algoritmo exitoso debe reducir el conjunto de datos, afectando en la menor medida posible su capacidad de generalización.
- *Tiempo*: A pesar de que el proceso preprocesamiento y aprendizaje debe realizarse solo una vez, la complejidad de los algoritmos pueden volverlos poco prácticos para su uso sobre conjuntos de datos "grandes".

Capítulo 2

Metaheurísticas para seleccionar instancias

2.1. Metaheurísticas

Las metaheurísticas son métodos estocásticos de búsqueda de propósito general sobre espacios combinatorios. Son usados generalmente para tratar problemas de optimización combinatoria, donde su complejidad hace imposible evaluar todas las soluciones factibles en un tiempo razonable; estos algoritmos son capaces de conseguir "buenas" soluciones a un problema en un período de tiempo mucho menor. Sin embargo, para muchos problemas la complejidad de estos algoritmos sigue siendo un factor prohíbitivo, debido al uso de funciones "costosas" para la evaluación de soluciones intermedias (*fitness*).

La idea es desarrollar algoritmos que recorran solo una fracción del espacio de soluciones, y que sean capaces de encontrar soluciones óptimas o casi óptimas al problema en cuestión. Para lograrlo, las metaheurísticas combinan procesos de *diversificación* e *intensificación* (o *exploración* y *explotación* respectivamente) [Yan08]. La fase de *diversificación* implica la generación de soluciones diversas con el objeto de explorar el espacio de búsqueda, mientras que la fase de *intensificación* se refiere al mejoramiento de soluciones (conseguir óptimos locales) mediante el uso de métodos de búsqueda local. La selección de las mejores soluciones asegura la convergencia a soluciones óptimas, mientras que la exploración aleatoria de soluciones evita que el algoritmo quede "atrapado" en óptimos locales. La combinación

en el uso de ambos procesos hace posible conseguir buenas soluciones al problema sin la necesidad de recorrer el espacio de búsqueda completo.

Cada metaheurística está caracterizada por las estratégias que usa para cada fase, así como el orden y la frecuencia en que las aplica; esto permite clasificarlas en función de su similitud. En este sentido, a continuación se describe un conjunto de metaheurísticas caracterizadas por tener a la naturaleza como fuente de inspiración.

2.2. Metaheurísticas inspiradas en la naturaleza

La habilidad de la naturaleza para moldear soluciones a situaciones complejas, mediante procesos y reglas caracterizadas por su simplicidad, la ha convertido en una fuente inagorable de inspiración para el desarrollo de algoritmos de optimización. Estos algoritmos a menudo presentan buen desempeño para aproximar soluciones a todo tipo de problemas, dado que no requieren información sobre la distrubución del espacio de búsqueda. Por esta razón, existe una amplia literatura sobre enfoques bio-inspirados [BS12] para resolver gran variedad de problemas en diversas áreas de computación.

En particular, los enfoques más comunes en la literatura sobre metaheurísticas inspiradas en la naturaleza se apoyan en *a*) la evolución de poblaciones (Algoritmos Evolutivos) y *b*) el comportamiento colectivo (Inteligencia de Enjambre).

2.2.1. Algoritmos Evolutivos

Los Algoritmos Evolutivos (AE) son metaheurísticas basadas en procesos de evolución biológica con el objetivo de explorar en amplitud espacios de solución con distribución desconocida. Con el fin de replicar los procesos evolutivos, los AE mantienen un conjunto de soluciones candidatas al problema (una *población* de *cromosomas*/*individuos*), que modifican iterativamente apoyandose en el uso de operadores de *mutación*, *recombinación* y/o *selección*.

Los AE codifican cada cromosoma como una cadena de genes de tamaño l (análogo a la estructura del ADN), donde cada gen representa una parte de la solución al problema en cuestión. A partir de esta representación, los AE definen

un conjunto de operadores que cumplen la función de las estrategias de *exploración* y *explotación*:

- *Mutación*: Modifica los genes de soluciones intermedias con la finalidad de explorar el espacio de soluciones e introducir nueva información a la población. Simula la variabilidad en las poblaciones, fenómeno clave para la aparición de nuevos genes que aumenten la posibilidad de supervivencia.
- *Recombinación/Crossover*: Permite el intercambio de información entre individuos de la población. Simula la reproducción entre individuos, necesaria para la transmisión de genes relevantes a las siguientes generaciones.
- *Selección*: Las estrategias de selección permiten definir aquellos individuos que participarán en la fase de reproducción, y por ende los genes que pasarán a la siguiente generación. Esto simula el proceso de selección natural en el que sobreviven los individuos mejor adaptados al ambiente.

En la literatura se han desarrollado diferentes esquemas que definen el uso de estos operadores. Los AE más "tradicionales" son conocidos como *Algoritmos Genéticos* (AG) [Hol75], que suponen la aplicación más directa de los conceptos del proceso evolutivo. Sin embargo, dentro de la clase de AE existe otro grupo de algoritmos que aplican dichos conceptos de forma diferente; la clase de *Algorítmos de Estimación de Distribución* ("*Estimation of Distribution Algorithm*" - EDA) aplican los operadores de *mutación*, *recombinación* y *selección* sobre una población de soluciones implicita en un modelo de distribución probabilístico.

A continuación se describen cuatro algoritmos pertenecientes a la clase de AE: GGA, SGA y CHC, variantes del grupo de AG, y PBIL, perteneciente a los EDA.

2.2.1.1. Generational Genetic Algorithm (GGA)

GGA es el esquema "tradicional" de aplicación de los AG [Bac96, Muh91]. Mantiene una población de individuos que evolucionan durante un número de iteraciones. Su principal característica es que en cada iteración se genera una nueva población, *i.e.* un proceso de evolución *generacional*.

En cada iteración el proceso evolutivo consiste en la creación de una nueva población de tamaño `pop` mediante: *a*) la selección de los individuos para el proceso de reproducción (*padres*), *b*) la recombinación (con probabilidad `cp`) de

pares de individuos *padres* usando una estrategia particular de cruce/*crossover*, y *c*) la mutación de los individuos de la nueva población (llamados *descendencia*), usando una probabilidad de mutación de cada gen igual a `mp`. Ver el algoritmo 2.1.

Algoritmo 2.1 Generational Genetic Algorithm

Input: `pop` tamaño de la población, `cp` probabilidad de cruce, `mp` probabilidad de mutación
Output: Una solución al problema

```
1: P ← Generar población aleatoria de pop individuos
2: p* ← el mejor individuo en P
3: while ¬ Condición de parada do
4:     P' ← ∅
5:     while | P' | < pop do
6:         p1 ← Seleccionar un individuo en P
7:         p2 ← Seleccionar un individuo en P
8:         c1, c2 ← recombinar p1 y p2 con probabilidad cp
9:         Mutar c1 y c2 con probabilidad mp
10:        P' ← P' ∪ {c1, c2}
11:    P ← P'
12:    if El mejor individuo en P es mejor que p* then
13:        p* ← el mejor individuo en P
14: return p*
```

2.2.1.2. Steady-State Genetic Algorithm (SGA)

Descrito por *Whitley et al.* [WK88], SGA es una modificación del esquema general de *AG* que sigue una estrategia reproductiva no generacional. SGA comienza con una población de tamaño `pop`, y en cada iteración se producen un máximo de dos nuevos individuos (no una nueva población).

En cada iteración *a*) se seleccionan dos individuos padres de la población actual, *b*) se crea su descendencia (con probabilidad `cp`) mediante algún metodo de cruce/recombinación, *c*) se agrega variabilidad mediante la mutación (con probabilidad `mp`) de la nueva descendencia, y *d*) se sigue alguna estrategia de selección para reemplazar individuos en la población por la nueva descendencia, y así mantener el tamaño de la población igual a `pop`. Ver el algoritmo 2.2.

Algoritmo 2.2 Steady-State Genetic Algorithm

Input: `pop` tamaño de la población, `cp` probabilidad de cruce, `mp` probabilidad de mutación
Output: Una solución al problema

```
1: P ← Generar población aleatoria de pop individuos
2: p* ← el mejor individuo en P
3: while ¬ Condición de parada do
4:     p1 ← Seleccionar un individuo en P
5:     p2 ← Seleccionar un individuo en P
6:     c1, c2 ← recombinar p1 y p2 con probabilidad cp
7:     Mutar c1 y c2 con probabilidad mp
8:     Seguir algún criterio de reemplazo de individuos en P por c1 y c2
9:     if El mejor individuo en P es mejor que p* then
10:        p* ← el mejor individuo en P
11: return p*
```

2.2.1.3. CHC Adaptive Search Algorithm

CHC [Esh90] se basa en el esquema de evolución generacional aplicado por GGA; mantiene una población de individuos de tamaño fijo (`pop`), generando una nueva población en cada iteración. Sin embargo, en cada iteración CHC aplica una estrategia de reemplazo "elitista", donde sobreviven los mejores individuos entre la población actual y la descendencia producida.

La fase de reproducción aplicada por CHC tiene dos particularidades. En primer lugar, implementa un operador de recombinación llamado HUX, que intercambia la mitad de los genes que difieren entre los dos padres de forma aleatoria. Adicionalmente, CHC emplea "prevención de incesto": antes de realizar el cruce usando HUX, calcula la *distancia de Hamming* entre ambos padres; si dicha distancia es mayor a cierto umbral (inicialmente $l/4$, donde l es la longitud de los cromosomas), se realiza el cruce. En caso de no generarse ninguna descendencia durante una iteración particular, se disminuye el umbral en 1.

Durante el proceso de evolución de CHC no se aplica el operador de mutación: cuando el umbral de prevención de incesto llega a cero se considera que la población convergió, y comienza un proceso de repoblación en el que se usa la mejor solución encontrada hasta el momento. Se modifican hasta 35 % de sus genes de forma aleatoria para generar los $\texttt{pop} - 1$ individuos restantes de la nueva población, y luego continuar el proceso evolutivo.

A continuación se presenta el pseudocódigo para CHC (algoritmo 2.3).

Algoritmo 2.3 CHC Adaptive Search Algorithm

Input: `pop` tamaño de la población
Output: Una solución al problema

```
1: P ← Generar población aleatoria de pop individuos
2: p* ← el mejor individuo en P
3: μ ← l/4                                          ▷ Umbral de cruce
4: while ¬ Condición de parada do
5:     for i ∈ [1...pop/2] do
6:         p1 ← Seleccionar un individuo en P
7:         p2 ← Seleccionar un individuo en P
8:         if hamming(p1, p2) > μ then
9:             c1, c2 ← recombinar p1 y p2 usando HUX
10:            P ← P ∪ {c1, c2}
11:    if | P |= pop then
12:        μ ← μ − 1
13:        if μ = 0 then
14:            P ← Generar población de pop individuos usando p*
15:            μ ← l/4
16:    else
17:        P ← pop mejores individuos en P
18:        if El mejor individuo en P es mejor que p* then
19:            p* ← el mejor individuo en P
20: return p*
```

2.2.1.4. Population-Based Incremental Learning (PBIL)

PBIL es una metaheurística perteneciente a la clase de *Algoritmos de Estimación de Distribución* desarrollada por *Baluja* [Bal94] para su uso sobre cromosomas con representación binaria. PBIL destaca por ser más simple que los algoritmos genéticos tradicionales y por lograr mejores soluciones para gran variedad de problemas [Bal95, BC95].

Este algoritmo mantiene una población *implicita* de soluciones, mediante el uso de un vector de probabilidades V de tamaño l, donde V_i (con $i \in [1 \dots l]$) es la probabilidad que el i-esimo bit/gen de una solución en la población esté "prendido" (sea igual a 1). PBIL usa este vector de probabilidades para generar poblaciones de tamaño `pop` en cada iteración, y guiar el proceso evolutivo en base a las soluciones generadas.

Inicialmente $V_i = 0{,}5 \;\; \forall i \in [1 \dots l]$. Luego en cada iteración: *a*) se generan `pop` cromosomas binarios basados en las probabilidades en V, *b*) se "acerca" V hacia la mejor solución generada (usando una tasa de aprendizaje `lr`), *c*) se "aleja" V

de la peor solución generada (usando una tasa de aprendizaje negativa `nlr`), *d*) se sigue una estrategia de mutación sobre V en la que se aumenta o disminuye V_i en `ms` (*mutation shift*) con probabilidad de mutación `mp`. Ver algoritmo 2.4.

Algoritmo 2.4 Population-Based Incremental Learning

Input: `pop` tamaño de la población, `mp` probabilidad de mutación, `ms` mutation shift, `lr` learning rate, `nlr` negative learning rate
Output: Una solución al problema

$V \leftarrow$ Vector de probabilidades de tamaño l
$p^* \leftarrow$ Una solución cualquiera
while $\neg$ Condición de parada **do**
    $P \leftarrow$ Generar población de tamaño `pop` según las probabilidades en V
    $b \leftarrow$ El *mejor* individuo en P
    $w \leftarrow$ El *peor* individuo en P
    if b es *mejor* que p^* **then**
        $p^* \leftarrow b$
    for $i \in [1 \dots l]$ **do** ▷ Actualizar el vector de probabilidades
        $V_i \leftarrow V_i * (1 - \texttt{lr}) + b_i * \texttt{lr}$
        if $b_i \neq w_i$ **then**
            $V_i \leftarrow V_i * (1 - \texttt{nlr}) + b_i * \texttt{nlr}$
        if $\text{Unif}(0, 1) < \texttt{mp}$ **then** ▷ Mutación con probabilidad `mp`
            $V_i \leftarrow V_i * (1 - \texttt{ms}) + \text{UnifDiscreta}(0, 1) * \texttt{ms}$
return p^*

2.2.2. Inteligencia de Enjambre

2.2.2.1. Particle Swarm Optimization (PSO)

PSO se inspira en el comportamiento de organismos biológicos, en particular del vuelo de bandadas. Cada ave o partícula representa una solución del espacio de búsqueda, y tiene una posición y velocidad asociada, modificando su "vuelo" en relación a su propia experiencia y la de sus "compañeras". Diferentes estudios muestran que PSO obtiene mejores resultados que los algoritmos genéticos, y en menor tiempo de cómputo.

Inicialmente se obtienen P soluciones aleatorias, o partículas. Cada partícula i está representada por un posición en un espacio s-dimensional $\mathbf{x}_i = < x_{i1}, x_{i2}, \dots, x_{is} >$. Luego, se realizan un número de iteraciones (`MAX_ITER`) en las que se actualiza la posición de cada partícula de acuerdo a su velocidad v_i:

$$\mathbf{x}_i = \mathbf{x}_i + v_i$$

$$v_i = wv_i + c_1 Rand()(p_i - \mathbf{x}_i) + c_2 Rand()(p_g - \mathbf{x}_i)$$

Donde c_1 y c_2 con constantes, $Rand()$ es una función aleatoria $[0, 1]$, p_i es la mejor solución encontrada por la partícula i (de acuerdo a una función de evaluación/*fitness* establecida), p_g es la mejor solución global, y w es el "peso de inercia" que establece la posible variabilidad de v_i. w disminuye cada iteración de acuerdo a la siguiente fórmula:

$$w = \frac{(w_{start} - w_{end})(\texttt{MAX\_ITER} - \texttt{Iter})}{\texttt{MAX\_ITER} + w_{end}}$$

Siendo `Iter` la iteración actual del algoritmo, y w_{start} y w_{end} valores predeterminados.

2.3. Adaptación para el problema de Selección de Instancias

2.3.1. Representación

2.3.2. Función objetivo

Capítulo 3

Punto de partida

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue,

a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Capítulo 4

Evaluación Experimental

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue,

a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Conclusiones y Recomendaciones

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut

metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

1. First itemtext
2. Second itemtext
3. Last itemtext
4. First itemtext
5. Second itemtext

Bibliografía

[Bac96] T. Back. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford University Press, USA, 1996.

[Bal94] Shumeet Baluja. Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning. Technical report, 1994.

[Bal95] Shumeet Baluja. An empirical comparison of seven iterative and evolutionary function optimization heuristics. Technical report, 1995.

[BC95] Shumeet Baluja and Rich Caruana. Removing the genetics from the standard genetic algorithm. pages 38–46. Morgan Kaufmann Publishers, 1995.

[Ben75] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9):509–517, September 1975.

[BL97] Avrim Blum and Pat Langley. Selection of relevant features and examples in machine learning. *Artif. Intell.*, 97(1-2):245–271, 1997.

[BS12] S Siva Sathya Binitha S. A survey of bio inspired optimization algorithm. *International Journal of Soft Computing and Engineering*, 2(2):137–151, 2012.

[BT12] J. Bien and R. Tibshirani. Prototype selection for interpretable classification. *ArXiv e-prints*, February 2012.

[CF01] Vicente Cerveron and Francesc J Ferri. Another move toward the minimum consistent subset: a tabu search approach to the condensed nearest neighbor rule. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 31(3):408–413, 2001.

[CH67] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theor.*, 13(1):21–27, January 1967.

[CHL03] José Ramón Cano, Francisco Herrera, and Manuel Lozano. Using evolutionary algorithms as instance selection for data reduction in kdd: an experimental study. *Evolutionary Computation, IEEE Transactions on*, 7(6):561–575, 2003.

[DGKL94] Luc Devroye, Laszlo Gyorfi, Adam Krzyzak, and Gábor Lugosi. On the strong universal consistency of nearest neighbor regression function estimates. *The Annals of Statistics*, pages 1371–1385, 1994.

[Esh90] Larry J Eshelman. The chc adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. *Foundations of genetic algorithms*, pages 265–283, 1990.

[FH51] E. Fix and J. L. Hodges. Discriminatory analysis, nonparametric discrimination: Consistency properties. *US Air Force School of Aviation Medicine*, Technical Report 4(3):477+, January 1951.

[FI93] Usama M. Fayyad and Keki B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In Ruzena Bajcsy, editor, *IJCAI*, pages 1022–1029. Morgan Kaufmann, 1993.

[Gat72] Geoffrey W. Gates. The reduced nearest neighbor rule (corresp.). *IEEE Transactions on Information Theory*, 18(3):431–433, 1972.

[GCH08] Salvador García, José Ramón Cano, and Francisco Herrera. A memetic algorithm for evolutionary prototype selection: A scaling up approach. *Pattern Recognition*, 41(8):2693–2709, 2008.

[GK14] Lee-Ad Gottlieb and Aryeh Kontorovich. Near-optimal sample compression for nearest neighbors. *CoRR*, abs/1404.3368, 2014.

[GKK13] Lee-Ad Gottlieb, Aryeh Kontorovich, and Robert Krauthgamer. Efficient classification for metric data. *CoRR*, abs/1306.2547, 2013.

[GPY08] Roberto Gil-Pita and Xin Yao. Evolving edited k-nearest neighbor classifiers. *International Journal of Neural Systems*, 18(06):459–467, 2008.

[Har68] P. Hart. The condensed nearest neighbor rule (corresp.). *IEEE Trans. Inf. Theor.*, 14(3):515–516, September 1968.

[HLL02] Shinn-Ying Ho, Chia-Cheng Liu, and Soundy Liu. Design of an optimal nearest neighbor classifier using an intelligent genetic algorithm. *Pattern Recognition Letters*, 23(13):1495–1503, 2002.

[Hol75] J.H. Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. University of Michigan Press, 1975.

[KL04] Robert Krauthgamer and James R. Lee. Navigating nets: simple algorithms for proximity search. In J. Ian Munro, editor, *SODA*, pages 798–807. SIAM, 2004.

[KS04] Mohammad Kolahdouzan and Cyrus Shahabi. Voronoi-based k nearest neighbor search for spatial network databases. In *Proceedings of the Thirtieth International Conference on Very Large Data Bases - Volume 30*, VLDB '04, pages 840–851. VLDB Endowment, 2004.

[LHTD02] Huan Liu, Farhad Hussain, Chew Lim Tan, and Manoranjan Dash. Discretization: An enabling technique. *Data Min. Knowl. Discov.*, 6(4):393–423, October 2002.

[LM98] Huan Liu and Hiroshi Motoda. *Feature Extraction, Construction and Selection: A Data Mining Perspective*. Kluwer Academic Publishers, Norwell, MA, USA, 1998.

[LM02] Huan Liu and Hiroshi Motoda. On issues of instance selection. *Data Min. Knowl. Discov.*, 6(2):115–130, April 2002.

[Muh91] Heinz Muhlenbein. Evolution in time and space - the parallel genetic algorithm. In *Foundations of Genetic Algorithms*, pages 316–337. Morgan Kaufmann, 1991.

[Ska94] David B. Skalak. Prototype and feature selection by sampling and random mutation hill climbing algorithms. In William W. Cohen and Haym Hirsh, editors, *ICML*, pages 293–301. Morgan Kaufmann, 1994.

[SLI+01] Basilio Sierra, Elena Lazkano, Iñaki Inza, Marisa Merino, Pedro Larrañaga, and Jorge Quiroga. Prototype selection and feature subset

selection by estimation of distribution algorithms. a case study in the survival of cirrhotic patients treated with tips. In *Artificial Intelligence in Medicine*, pages 20–29. Springer, 2001.

[SSBD14] S. Shalev-Shwartz and S. Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[Tou02] Godfried T. Toussaint. Open problems in geometric methods for instance-based learning. In Jin Akiyama and Mikio Kano, editors, *JCDCG*, volume 2866 of *Lecture Notes in Computer Science*, pages 273–283. Springer, 2002.

[Vor08] Georges Voronoï. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. deuxième mémoire. recherches sur les parallélloèdres primitifs. *Journal für die reine und angewandte Mathematik*, 134:198–287, 1908.

[Wil72] DR Wilson. Asymptotic properties of nearest neighbor rules using edited data. *Institute of Electrical and Electronic Engineers Transactions on Systems, Man and Cybernetics*, 2:408–421, 1972.

[Wil91] Gordon Wilfong. Nearest neighbor problems. In *Proceedings of the Seventh Annual Symposium on Computational Geometry*, SCG '91, pages 224–233, New York, NY, USA, 1991. ACM.

[WK88] D. Whitley and J. Kauth. *GENITOR: A Different Genetic Algorithm*. Technical report (Colorado State University. Department of Computer Science). Colorado State University, Department of Computer Science, 1988.

[WM97] D. Randall Wilson and Tony R. Martinez. Instance pruning techniques. In *Proceedings of the Fourteenth International Conference on Machine Learning*, ICML '97, pages 403–411, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.

[Yan08] Xin-She Yang. *Nature-Inspired Metaheuristic Algorithms*. Luniver Press, 2008.

[ZS02] Hongbin Zhang and Guangyu Sun. Optimal reference subset selection for nearest neighbor classification by tabu search. *Pattern Recognition*, 35(7):1481–1490, 2002.

[Zuk10] A. V. Zukhba. Np-completeness of the problem of prototype selection in the nearest neighbor method. *Pattern Recognit. Image Anal.*, 20(4):484–494, December 2010.