

2. Aprendizaje e Inteligencia Computacional

2.1. Breve historia

Desde los inicios de la computación a mediados de los años cuarenta del siglo XX se pensó en la posibilidad de crear máquinas capaces de aprender. Posiblemente el primer artículo sobre aprendizaje computacional es el de McCulloch y Pitts¹ que muestra cómo emular compuertas lógicas proposicionales con modelos artificiales de redes neuronales. Por su parte, en el artículo seminal de 1950 Turing² propuso crear una máquina capaz de aprender como parte del programa para la construcción de computadoras que exhibieran una inteligencia general, a la cual le llamó la Máquina Niño (*Child Machine*).³ Desde entonces, en buena medida inspirados por estas ideas, los científicos computacionales empezaron a crear sistemas básicos de aprendizaje desde diferentes perspectivas. Algunos intentaron emular los procesos dentro del cerebro mediante redes neuronales artificiales. Un poco más tarde se empezaron a demostrar pruebas de convergencia en redes neuronales sencillas o perceptrones (e.j., Rosenblatt⁴). Esta línea de investigación ha evolucionado

¹ McCulloch, W. S., Pitts, W. (1943). **A logical calculus of the ideas immanent in nervous activity.** *Bulletin of Mathematical Biophysics* 5: 115-133.

² Turing, A. (1950). **Computing Machinery and Intelligence.** *Mind* 59: 433-460.

³ Morales, E. (2013). **Educando al “niño computacional” de Turing.** *Ciencias*, pp. 32-39.

⁴ Rosenblatt, F. (1958). **The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain.** *Cornell Aeronautical Laboratory, Psychological Review* 5(6):386-408.

de manera continua hasta nuestros días y actualmente se cuenta con redes neuronales de muchas capas para implementar el llamado aprendizaje profundo (*Deep Learning*). Otros investigadores buscaron cómo aprender a jugar juegos sencillos mediante estrategias de búsqueda, aprendizaje de posiciones y ajuste de una función de recompensa.⁵ De esta última se derivó el aprendizaje por refuerzo, en el que un agente computacional aprende cuál es la mejor acción a realizar en cada instante de tiempo para maximizar una función de recompensa. Recientemente se desarrolló un sistema que le ganó al campeón mundial de Go⁶ el cual combina aprendizaje por refuerzo con aprendizaje profundo. En los años sesenta y setenta se desarrollaron algoritmos para aprender representaciones simbólicas, como Hunt⁷ y Feigenbaum.⁸ Algunos de los algoritmos más utilizados en aprendizaje computacional generan árboles de decisión y sus variantes, las cuales se han aplicado a una gran cantidad de situaciones reales. En la misma época otros investigadores se enfocaron al desarrollo de sistemas inspirados en genética y evolución, por ejemplo Rechenberg,⁹ Schwefel,¹⁰ Fogel¹¹ y Holland.¹² De aquí surgieron varios algoritmos que se engloban en la computación evolutiva y se utilizan principalmente para resolver problemas de optimización complejos en los cuales no es posible utilizar las técnicas matemáticas tradicionales.¹³

⁵ Samuel, A. (1959). **Some Studies in Machine Learning Using the Game of Checkers.** *IBM Journal*. 3 (3): 210-229.

⁶ Silver, D., et. al. (2016). **Mastering the game of Go with deep neural networks and tree search.** *Nature*, 529: 484-489.

⁷ Hunt, E. B., Hovland, C. I. (1963). **Programming a Model of Human Concept Formation.** en E. A. Feigenbaum y J. Feldman (Eds.) *Computers and Thought*. MacGraw-Hill, 310-325.

⁸ Feigenbaum, E. A. (1963). **The Simulation of Verbal Learning Behaviour.** En E.A. Feigenbaum and J. Feldman (Eds.) *Computers and Thought*. MacGraw-Hill, 310-325.

⁹ Rechenberg, I. (1971). **Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution,** TU Berlin.

¹⁰ Schwefel, H. P. (1975). **Evolutionsstrategie und numerische Optimierung,** TU Berlin.

¹¹ Fogel, L. J. Owens, A. J., Walsh, M. J. **Artificial Intelligence through Simulated Evolution,** John Wiley & Sons, (1966).

¹² Holland, J. (1962). **Outline for a Logical Theory of Adaptive Systems.** *JACM* 9: 297-314.

¹³ Ver capítulo de Computación Evolutiva en este texto.

Además de aprender modelos u optimizar funciones, algunos investigadores trataron de formalizar el razonamiento humano de manera alternativa a los enfoques lógicos tradicionales, como la lógica de predicados. Uno de éstos fue Lofti Zadeh que creó la lógica difusa¹⁴ en la cual las expresiones no sólo toman valores de verdadero y falso (1 y 0) sino también valores reales entre estos dos extremos. Esta línea de investigación, al igual que las anteriores, se ha desarrollado y refinado de forma continua hasta nuestros días. Todas éstas forman parte del Aprendizaje e Inteligencia Computacional, que son el tema de este capítulo.

2.2. Aprendizaje Computacional

El Aprendizaje Computacional (*Machine Learning*) se dirige al estudio y desarrollo de algoritmos con la capacidad de mejorar su desempeño con la experiencia. Otros nombres comúnmente utilizados para denotar el área son Aprendizaje Automático, Aprendizaje Maquinal y Aprendizaje de Máquina. El énfasis de esta especialidad es crear algoritmos que puedan inducir modelos automáticamente sin la intervención o asistencia humana. Dado que la capacidad de aprender es una característica distintiva de la inteligencia no es de extrañar que el Aprendizaje Computacional sea un área central de la Inteligencia Artificial.

Para entender la utilidad práctica y relevancia del área podemos pensar en cómo se resuelven comúnmente muchos problemas. Generalmente las personas crean modelos o programas para resolverlos; sin embargo, algunos son difíciles de formalizar, no siempre se cuenta con expertos en el dominio o se tiene demasiada información. Por ejemplo, si se requiere escribir un programa para reconocer personas, no es claro cómo programarlo, a qué expertos consultar o cómo limitar el número de fotos y videos de personas existentes. Los algoritmos de aprendizaje permiten generar automáticamente dichos programas o modelos a partir de datos y con ellos resolver problemas.

¹⁴ Zadeh, L. (1965). **Fuzzy Sets**. *Information Sciences* 8: 338-353.

Esto claramente abre una gran cantidad de posibles aplicaciones que son difíciles de formalizar o programar por una persona y que requieren contar con grandes cantidades de datos para su desarrollo.

El reciente éxito y publicidad que han recibido estos algoritmos se deben principalmente a tres factores: i) disponibilidad de una gran cantidad de datos de todo tipo, ii) máquinas más poderosas con altas capacidades de almacenamiento y mayores velocidades de procesamiento, y iii) la facilidad de utilizar estas herramientas por las empresas para resolver problemas reales.

Algunos autores clasifican al área de aprendizaje computacional en términos de tres tipos o categorías de modelos: geométricos, probabilistas y lógicos. Otra clasificación posible se da en términos de las tareas que abordan. Una caracterización burda, la cual se adopta en este capítulo, es: 1) Sistemas que crean descripciones, resúmenes, identifican prototipos o casos anómalos a partir de datos; 2) sistemas que segmentan/agrupan datos automáticamente; 3) sistemas que crean automáticamente modelos para predecir salidas, ya sea para tareas de clasificación (salidas discretas) o de regresión (salidas continuas); 4) sistemas que encuentran automáticamente dependencias entre variables, ya sean éstas probabilistas, funcionales, asociativas o causales; y 5) sistemas que aprenden a crear planes o estrategias de control para resolver problemas de decisión secuenciales. A continuación se presenta una breve descripción de cada una de éstas.

2.2.1. Resúmenes y detección de prototipos

Con el incremento acelerado en la generación de datos es importante tener la capacidad de crear automáticamente resúmenes que resalten la información más relevante. Los estadísticos entendieron rápidamente la utilidad de esta funcionalidad y generaron medidas, como la media o la moda, y medidas de dispersión, como la desviación estándar, para tener una idea rápida de la naturaleza de los datos. En aprendizaje computacional algunos investigadores han

tratado de encontrar los datos prototípicos dentro de una gran cantidad de datos para hacer inferencias, facilitando el análisis y reduciendo la velocidad de procesamiento. Las personas que trabajan con análisis de texto también han creado sistemas que generan resúmenes de textos automáticamente para su fácil lectura y comprensión. También se pueden usar técnicas para reducción de datos identificando únicamente los datos atípicos, los cuales son muchas veces los de mayor interés.

2.2.2. Segmentación

Los sistemas que segmentan automáticamente datos se conocen como algoritmos de agrupamiento o *clustering*.^{15,16} También se conocen como algoritmos de aprendizaje no supervisado. La idea de estos algoritmos es agrupar automáticamente datos de tal forma que los elementos en cada grupo se parezcan entre sí, bajo algún criterio, y no se parezcan —o se parezcan poco— con los elementos de otros grupos, bajo el mismo criterio. Por ejemplo, los datos de clientes de empresas comerciales se pueden agrupar bajo ciertos criterios para diseñar campañas publicitarias diferenciadas. Los grupos pueden ser disjuntos o tener traslapes, organizarse en jerarquías, ser exhaustivos, la pertenencia a los grupos puede ser probabilística, etc. La medida de similaridad se define generalmente por la proximidad que existe entre los elementos en un espacio multidimensional y hay una gran cantidad de medidas posibles para este efecto. Si los datos son exclusivamente numéricos se pueden usar medidas de distancia, como la Euclideana. Si los datos no sólo se describen por números, se usan otras medidas de similaridad que dejan de cumplir con las propiedades de las métricas. También se les puede dar más peso a algún tipo variable o atributo que a otro(s) para que sus valores tengan más influencia en la creación de los grupos. Por otro lado, también hay diferentes estrategias para formar los grupos, algunas agrupan a los elementos gradualmente, otras

¹⁵ Estivill-Castro, V., Yang, J. (2001). **Non-crisp Clustering by Fast, Convergent, and Robust Algorithms**. En *Proc. of PKDD*, pp. 103-114.

¹⁶ Estivill-Castro, V., Houle, M. E. (2001). **Data Structures for Minimization of Total Within-Group Distance for Spatio-temporal Clustering**. En *Proc. PKDD*, pp. 91-102.

parten de un grupo global el cual se separa en subgrupos de forma recurrente, otros parten de particiones formadas originalmente de forma aleatoria las cuales se modifican hasta cumplir con un criterio dado de antemano, etcétera.

Uno de los algoritmos más conocidos de *clustering* es el de *k-means*. El algoritmo es relativamente simple y parte del número de particiones k que se quieren formar. El algoritmo selecciona aleatoriamente k elementos y calcula la similaridad de todos los datos con dichos k elementos. Cada dato se asocia al elemento k más cercano con lo que se forman k grupos y se obtiene su centroide o elemento “promedio”. El proceso se realiza de manera recurrente con respecto a los centroides, con lo que se forman, posiblemente, nuevos grupos. El proceso se repite hasta que los centroides no cambian. Existen muchas extensiones y variantes a este algoritmo.

Los datos se pueden agrupar desde diferentes perspectivas y de diferentes formas. Se han desarrollado medidas de calidad para los grupos que se forman tomando en cuenta la densidad de elementos, qué tan compactos son los grupos, etc. Algunos retos del área son cómo lidiar con diferentes representaciones en los datos, cómo crear grupos con formas arbitrarias, cómo incluir restricciones y cómo hacerlos interpretables, entre muchos otros.

2.2.3. Predicción

Existen algoritmos que crean automáticamente modelos para predecir salidas, ya sea para tareas de clasificación (salidas discretas) o de regresión (salidas continuas). La tarea de predicción es posiblemente la que más atención ha recibido en la literatura por su gran cantidad de posibles aplicaciones. A estos algoritmos también se les conoce como aprendizaje supervisado. La información de entrada dada a estos algoritmos son ejemplos, descritos generalmente con atributos o variables, asociados con una etiqueta. Cuando la etiqueta es nominal se conocen como algoritmos de clasificación, cuando es numérica se conoce como regresión o estimación.

2.2.3.1. Clasificación

La meta de los algoritmos de clasificación es inducir automáticamente un modelo a partir de los datos para que dada la descripción de un nuevo ejemplo produzca una etiqueta de salida. Por ejemplo, a partir de una descripción de atributos de un paciente prediga una enfermedad o dados los valores de variables de una línea de producción detecte una falla. En la tarea de clasificación existe una gran cantidad de algoritmos como árboles de decisión, máquinas de soporte vectorial y aprendizaje basado en instancias, entre otros. En este capítulo se describen brevemente estos tres enfoques.

En un árbol de decisión cada nodo representa un atributo y sus ramas los posibles valores que dicho atributo puede tomar. En caso de atributos continuos, éstos se dividen automáticamente en dos rangos que son mayores o menores que un cierto valor. Las hojas de los árboles son los valores de las clases que queremos predecir. El algoritmo selecciona de manera heurística el atributo que mejor particiona los datos en términos de las clases. Una vez que selecciona un atributo, el algoritmo se vuelve a aplicar a los datos de cada una de sus ramas hasta que: i) se queda con ejemplos de una sola clase y etiqueta la hoja con esa clase, ii) se queda sin ejemplos y etiqueta la hoja con la clase mayoritaria de su padre, o iii) se queda sin atributos y etiqueta la hoja con la clase mayoritaria de sus datos. Para clasificar un nuevo ejemplo, se recorre el árbol de acuerdo a los valores de los atributos del ejemplo y al llegar a una hoja se regresa la clase asociada a esa hoja. Existen varias medidas de selección de atributos, pero la más conocida se basa en ganancia de información la cual selecciona el atributo que reduce más la entropía o dispersión de las clases en los datos. Esto es, el atributo en donde los datos contenidos en sus ramas tienen más claramente definidas alguna de las clases.

Otro algoritmo muy usado en clasificación son las máquinas de soporte vectorial o de vectores de soporte. Una forma de construir un clasificador entre dos clases es tratar de encontrar un plano, o hiperplano en caso de

más dimensiones, que mejor divida a las clases. En las máquinas de soporte vectorial el encontrar ese hiperplano se plantea como un problema de optimización en donde se busca el plano que divida a las clases y que esté más alejado de los ejemplos de las clases contrarias. Como en muchos problemas las clases no son linealmente separables (no existe un hiperplano que las divida perfectamente) se recurre a dos estrategias; la primera es usar variables de holgura, lo cual permite tener “excepciones” de algunos ejemplos, y la segunda, mucho más conocida e importante, es la que se conoce como el “*kernel trick*”, que consiste en crear una representación de más dimensiones que la original con la esperanza de que en ese nuevo espacio se pueda encontrar un mejor hiperplano.

Finalmente, están los algoritmos de aprendizaje basados en instancias, los cuales en vez de inducir automáticamente un modelo que sirva para predecir una clase almacenan todos los datos y buscan cuál o cuáles son los datos más parecidos al ejemplo a clasificar y los utilizan para predecir un valor. El algoritmo más conocido es el de k vecinos más cercanos. La idea es obtener del conjunto de datos los k elementos que sean más parecidos –bajo cierta medida– al ejemplo que se quiere clasificar, analizar las clases de esos k vecinos y regresar la clase mayoritaria. Al igual que en los algoritmos de *clustering*, existen diferentes medidas de similaridad que se pueden utilizar para encontrar los vecinos más cercanos. Muy relacionado a estos algoritmos está el razonamiento basado en casos donde cada ejemplo, llamado caso, se representa por la descripción de un problema, la solución empleada y los resultados obtenidos. Cuando se quiere resolver un nuevo problema, se consultan las descripciones de los problemas en la base de casos, se obtienen los más parecidos y se utilizan sus soluciones, con posibles adecuaciones, para resolver el problema en cuestión.

La gran mayoría de los algoritmos de clasificación utiliza una representación de los ejemplos basada en pares atributo-valor produciendo en muchos casos modelos proposicionales. Existen otros algoritmos que trabajan con

representaciones más expresivas utilizando grafos¹⁷ o lógica de primer orden. Una de ellas es la programación lógica inductiva^{18,19} la cual recibe de entrada datos e información relacional y produce modelos expresados en términos de relaciones. Esto permite inducir modelos que contengan funciones, variables y definiciones recursivas los cuales son más expresivos que los modelos generados por los algoritmos anteriores. Algunos de los retos de los sistemas de clasificación son el ruido (valores erróneos y faltantes) que puede existir en los datos, una gran cantidad de datos, pocos datos pero muchos atributos, muchas clases posibles, entre otros.

2.2.3.2. Regresión

La meta de los algoritmos de estimación o regresión es inducir automáticamente, a partir de los datos, un modelo que dada la descripción de un ejemplo produzca un valor continuo de salida. Por ejemplo, predecir el valor de una temperatura de salida en base a los valores de las variables de un proceso industrial. Algunos de los algoritmos más usados son las redes neuronales, regresión lineal, regresión localmente pesada, procesos gaussianos, etcétera.

Además del método clásico de mínimos cuadrados, posiblemente el algoritmo más conocido de regresión sea el de las redes neuronales. A continuación se describe la red más conocida como *feed-forward* con retro-propagación. Esta red neuronal es un grafo dirigido organizado por capas, en donde todos los nodos de una capa están conectados hacia todos los nodos de la siguiente capa y reciben información de todos los nodos de la capa anterior. Cada conexión está asociada con un peso. A cada nodo le llega la suma de todas las salidas de los nodos de la capa anterior multiplicadas por sus respectivos pesos y a ese valor generalmente se le aplica una función continua no

¹⁷ Gonzalez, J.A. Holder, L.B. (2000). **Graph Based Concept Learning**. En *Proc. Seventeenth National Conference on Artificial Intelligence (AAAI)*, 2000.

¹⁸ Morales, E. F. (1997). **PAL: A Pattern-Based First-Order Inductive System**. *Machine Learning* 26: 227-252.

¹⁹ MacKenney, R. (1999). **Learning using higher Order Functions**. En *Proc. Inductive Logic Programming (ILP)*, pp 42-46.

lineal diferenciable que produce la salida de ese nodo. Esa salida, junto con las salidas de todos los nodos, de esa capa, se propaga de la misma forma hacia la siguiente capa, hasta producir una o más salidas finales. Los pesos de las redes se inicializan aleatoriamente y el aprendizaje consiste en ajustar los pesos de la red para que con base en los valores de ejemplos conocidos se produzcan las salidas deseadas. Este ajuste se hace gradualmente alterando los pesos en la dirección que produce el máximo descenso en la superficie del error. Para esto se obtiene el negativo de la derivada o gradiente del error con respecto a los pesos y se suma en forma ponderada al peso actual. El error de salida se distribuye entre todos los pesos de las capas anteriores. Este proceso se realiza con todos los datos de entrenamiento varias veces hasta que los pesos producen resultados aceptables o cambian poco los pesos de la red.

Las redes neuronales fueron populares en los inicios de la Inteligencia Artificial y aunque dejaron de usarse por varios años recuperaron su popularidad cuando se difundió el algoritmo de entrenamiento que acabamos de describir; recientemente ha habido mucho interés a partir de que se logró entrenar redes de “muchas” capas (más de 5). Aunque algunas de las ideas originales de entrenamiento de muchas capas, conocido ahora como aprendizaje profundo, se plantearon hace tiempo, la disponibilidad de grandes cantidades de datos y las velocidades de procesamiento de las máquinas actuales fueron las que permitieron realizarlo. Estas redes se han utilizado con éxito en análisis de imágenes, procesamiento de lenguaje natural y en juegos.

Algunos de los retos del área son, como en todas las demás, la cantidad de ruido que exista en los datos, si se tienen datos numéricos y nominales, la posibilidad de realizar un sobre-ajuste en los modelos y el tiempo de entrenamiento de las redes.

2.2.4. Análisis de dependencias

Algunos algoritmos encuentran automáticamente dependencias entre variables, ya sean éstas probabilísticas, funcionales, asociativas o causales. Estos mo-

delos permiten determinar si el valor de una variable depende de los valores de las otras. Por ejemplo, cuál es la probabilidad de que llueva dado que la presión atmosférica y la temperatura tienen ciertos valores, o qué tan frecuente ocurre y con qué confianza, que las personas que compran leche y pan, también compren mantequilla y mermelada.

Una de las técnicas más usadas para capturar dependencias probabilísticas son los modelos gráficos probabilistas, como las redes bayesianas.²⁰ Estos modelos consisten de grafos acíclicos dirigidos en los que los nodos representan variables aleatorias y los arcos sus dependencias probabilistas, y cada grafo representa una distribución de probabilidad conjunta de varias variables. Este formalismo permite evaluar los valores más probables de un conjunto de variables dados los valores de otras variables, y modela el razonamiento probabilista basado en la regla de Bayes. Los algoritmos de aprendizaje de redes bayesianas inducen tanto la estructura de la red como las dependencias entre las variables y se expresan como tablas de probabilidad condicional de cada variable dados sus padres.²¹ El aprendizaje paramétrico o de los valores de las tablas de probabilidad condicional se basa generalmente en obtener las frecuencias de los valores de las variables relacionadas. Las técnicas de aprendizaje estructural se basan en métodos que utilizan medidas de ajuste y estrategias de búsqueda, o en métodos basados en pruebas de independencia.

Uno de los métodos más usados, que forma la base de otros algoritmos de aprendizaje de redes bayesianas, es el de Chow y Liu.²² En este algoritmo se obtiene la información mutua entre todos los pares de variables, se ordenan estas medidas de mayor a menor, donde la de mayor información mutua representa un grafo inicial (con dos nodos) y se agregan las siguientes ramas mientras no se formen ciclos hasta que se hayan incluido todas las variables.

²⁰ Ver el capítulo de Conocimiento y Razonamiento en este texto.

²¹ Luis-Velázquez, R., Sucar, L. E., Morales, E. F. (2010). **Inductive Transfer for Learning Bayesian Networks**. *Machine Learning* 79 (1-2): 227-255.

²² Chou, C. K., Liu, C.N. (1968), **Approximating discrete probability distributions with dependence trees**. *IEEE Transactions on Information Theory*, IT-14 (3): 462-467.

Algunas extensiones a este algoritmo permiten determinar la dirección de las ramas y construir grafos más sofisticados (poliárboles o redes multiconectadas). Dentro de los retos del área podemos mencionar el razonamiento con variables continuas y mixtas así como dominios en donde exista una gran cantidad de dependencias, lo cual complica el algoritmo de inferencia.

Otra forma de obtener relaciones entre variables es por medio de las reglas de asociación. Estos algoritmos buscan grupos de variables y valores que se repitan frecuentemente en los datos. Con estos grupos de variables y valores repetidos, construyen reglas que tienen subgrupos de pares atributo-valor tanto en el antecedente como en el consecuente. El algoritmo más conocido del área es el de *Apriori*.²³ En éste se buscan primero todos los pares atributo-valor que cumplan con un mínimo de repeticiones o soporte en los datos. Con estos se buscan todas las parejas de pares atributo-valor con el soporte deseado. Esto continúa con tercias y así sucesivamente, hasta que no exista ningún subconjunto de pares atributo-valor más grande con el soporte requerido. Con cada subconjunto encontrado se crean todas las posibles reglas que cumplan con un nivel de confianza (soporte del subconjunto entre el soporte del antecedente de la regla) formadas por las combinaciones de antecedentes y consecuentes de los elementos de los subconjuntos. Esta área está muy relacionada con los algoritmos de patrones frecuentes y los de descubrimiento de subgrupos, los cuales se basan en buscar subconjuntos de pares atributo-valor frecuentes. Uno de los retos de las reglas de asociación es cómo encontrar todos estos conjuntos frecuentes de manera eficiente.

Tanto las redes bayesianas como las reglas de asociación se pueden modificar para resolver problemas de clasificación, donde el interés es predecir el valor de la variable que determina a la clase. Aunque se han desarrollado también algoritmos para encontrar dependencias funcionales y causales, no han sido tan populares como las redes bayesianas y las reglas de asociación.

²³ Agrawal, R., Srikant, R. (1994). **Fast algorithms for mining association rules in large databases. Proceedings of the 20th International Conference on Very Large Data Bases, VLDB**, pp. 487-499, Santiago, Chile.

2.2.5. Planeación

Finalmente, existen algoritmos que aprenden a crear planes o estrategias de control para resolver problemas de decisión secuencial. Por ejemplo, decidir una serie de acciones para cumplir una meta, como ganar un juego o controlar algún dispositivo, como un robot.

Posiblemente el más conocido en años recientes sea el aprendizaje por refuerzo.^{24,25} En este tipo de aprendizaje un agente aprende por prueba y error cuál es la mejor acción a realizar en cada momento para maximizar el valor esperado de su recompensa total. Para aprender qué acción se debe realizar en cada estado (aprender una política) el agente selecciona una de sus posibles acciones, con la cual se mueve –posiblemente– a otro estado y recibe una recompensa. La selección de acciones es aleatoria inicialmente y se ajusta gradualmente tomando en cuenta la recompensa recibida y la acumulada en otros estados. Uno de los algoritmos más conocidos es el de Q-learning,²⁶ el cual ajusta gradualmente el valor de la recompensa total esperada de cada par estado-acción. El valor de cada par estado-acción se ajusta incrementalmente calculando la diferencia entre dicho valor y la suma de la recompensa recibida al tomar la acción y el máximo valor esperado de todos los siguientes pares estado-acción. Algunos de los retos de esta área son cómo lidiar con espacios de estados y acciones continuos, cómo aprender con muchas variables, qué hacer en ambientes no estacionarios y cómo reducir la velocidad de aprendizaje.

²⁴ Morales, E. F., Sammut, C. (2004). **Learning to Fly by Combining Reinforcement Learning with Behavioural Cloning.** En *Proc. of the Twenty-first International Conference on Machine Learning (ICML)*, pp. 598-605.

²⁵ García, E. O., Muñoz de Cote, E. , Morales, E. F. (2014). **Transfer Learning for Continuous State and Action Spaces.** *International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI)* 28 (7).

²⁶ <https://en.wikipedia.org/wiki/Q-learning>

2.2.6. Otros enfoques

Existen muchos otros algoritmos y estrategias de aprendizaje. Algunos que han perdido popularidad se basan en abducción como el aprendizaje basado en explicaciones, el aprendizaje de macro-operaciones, usados en planeación y los algoritmos de descubrimiento y aprendizaje por medio de analogías, entre otros.

Algunas de las variantes más utilizadas actualmente son los ensambles, en los cuales se inducen varios modelos a la vez y se combinan sus resultados para resolver alguna tarea. También se han desarrollado algoritmos de aprendizaje por transferencia, en donde se utiliza información de una tarea resuelta para resolver otra de manera más eficiente y/o más efectiva.

Dentro de los algoritmos de clasificación se han desarrollado algoritmos de aprendizaje semi-supervisado, en donde se combinan datos etiquetados y no etiquetados (esto es, con y sin una clase asociada) para tratar de construir un mejor clasificador. También se han desarrollado algoritmos para tratar de predecir un conjunto de clases al mismo tiempo o algoritmos multi-etiqueta.^{27,28}

Finalmente, existen algoritmos que buscan que un agente nunca deje de aprender, lo que se conoce como *life-long learning*. Existe mucha expectativa y algunos investigadores creen que dada la velocidad con la cual se desarrolla el área, pronto se va a lograr crear una inteligencia artificial superior a la de los humanos.

²⁷ Zaragoza, J. H., Sucar, L. E., Morales, E. F., Bielza, C., Larrañaga, P. (2011). **Bayesian Chain Classifiers for Multidimensional Classification**. En *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2192-2197.

²⁸ Hernández, J., Sucar, L. E., Morales, E. F. (2014). **Multidimensional hierarchical classification**. *Expert Systems with Applications* 41 (17): 7671-7677.

2.3. Lógica Difusa

La Lógica Difusa es un enfoque de computación basado en grados de verdad en lugar del tradicional verdadero/falso (1 ó 0) de la lógica Booleana.²⁹ De acuerdo a Zadeh, la membresía de un elemento a un conjunto difuso es gradual con valores en el intervalo $[0, 1]$. Un ejemplo clásico es clasificar individuos en el conjunto de personas altas considerando su estatura. En los conjuntos típicos o duros, un elemento pertenece o no al conjunto, por lo que es necesario establecer un límite inferior (ej. 1.75 mts.) arriba del cual todos son altos y debajo ninguno lo es. En los conjuntos difusos la membresía es gradual y alguien que mide 1.75 mts. es alto en un grado, por ejemplo, de 0.7, e incluso uno que mide 1.70 mts. se puede considerar alto en un grado, por ejemplo, de 0.3; por su parte quien mida 1.82 mts. es indudablemente alto y su pertenencia al conjunto es 1. Todos los que midan más de 1.82 seguirán siendo altos en un grado de 1 y quienes midan, por ejemplo, 1.60 o menos no pertenecerán a dicho conjunto y su grado de pertenencia será 0.

Existen diferentes operaciones para combinar valores de membresía en conjuntos difusos: una intersección difusa es el valor menor o mínimo de la membresía de cada elemento en ambos conjuntos; la unión de conjuntos difusos es el valor más alto o máximo de los dos valores difusos y el complemento es la diferencia entre su valor y 1.

También existen definiciones para operaciones entre relaciones difusas, las cuales se establecen entre dos elementos en algún grado de pertenencia entre 0 y 1. Sean $R(X, Y)$ y $S(Y, Z)$ dos relaciones difusas donde X , Y y Z son conjuntos difusos; hay varias operaciones binarias aplicables sobre ellas que resultan en una tercera relación también difusa, como el Producto Círculo o *Circllet*, el Producto Subtriángulo, el Producto Supertriángulo y el Producto Cuadrado.

²⁹ Zadeh, L. (1963). **Fuzzy Sets**. *Information Sciences* 8: 338-353.

Los productos relacionales difusos se han utilizado como base de las Redes Neurales Relacionales Difusas con las que se han desarrollado diversas aplicaciones de clasificación por el grupo del INAOE,³⁰ principalmente para el reconocimiento del llanto del bebé con fines de diagnóstico.

Las operaciones entre relaciones difusas se aplican también a los sistemas expertos. Por ejemplo, sean X , Y y Z un conjunto de pacientes, síntomas y enfermedades respectivamente. Mediante relaciones difusas y sus operaciones se pueden representar proposiciones como: el paciente x muestra el síntoma y en grado G ; el síntoma y es uno de los que caracterizan la enfermedad z en grado G ; el paciente x tiene al menos uno de los síntomas de la enfermedad z en grado G ; x es un síntoma de la enfermedad z en grado G ; x tiene todos los síntomas de la enfermedad z y quizás más en grado G ; o x tiene exactamente los síntomas de enfermedad z y no otros en grado G ; donde G es un valor real entre 0 y 1.

Un Sistema Experto Difuso (SED)³¹ utiliza este tipo de proposiciones difusas y lógica difusa en vez de booleana para razonar acerca del problema; específicamente, consiste de una colección de funciones de membresía y de un conjunto de reglas de forma:

SI x es A ENTONCES y es B

donde x y y son variables lingüísticas y A y B son valores lingüísticos determinados por conjuntos difusos en el universo de discurso X y Y , respectivamente; por ejemplo:

Regla 1: SI *la velocidad es rápida* ENTONCES *distancia_de_frenado es larga*

³⁰ Suaste-Rivas, I., Reyes-Galaviz, O. F., Diaz-Mendez, A., Reyes-Garcia, C. A. (2004). **Implementation of a Linguistic Fuzzy Relational Neural Network for Detecting Pathologies by Infant Cry Recognition**. En LNCS 3315: Advances in Artificial Intelligence, Edited by C. Lemaitre, C. A. Reyes and J. Gonzalez, Springer, Berlin, pp. 953-962.

³¹ Baghel, A., Tilotma, S. (2013). **Survey on Fuzzy Expert System**. *International Journal of Emerging Technology and Advanced Engineering Website*, 3(12):230-233.

Regla 2: SI *la velocidad es lenta* ENTONCES *distancia_de_frenado es corta*

El rango de la variable *velocidad* (el universo del discurso) es de 0 a 220 km/h, pero esta gama incluye conjuntos difusos, tales como *lento*, *medio* y *rápido*. El universo del discurso de la variable *distancia_de_frenado* puede estar entre 0 y 300 mts. e incluye los conjuntos difusos *corta*, *media* y *larga*. Así, las reglas difusas se refieren a conjuntos difusos y los SED pueden reducir hasta en un 90 por ciento el número de reglas requeridas para modelar un dominio en relación a un sistema experto convencional.

Otra marcada diferencia con respecto a los sistemas expertos convencionales es el proceso de inferencia, el cual consiste básicamente de cuatro etapas:

1. Fusificación: las funciones de membresía definidas sobre las variables de entrada son aplicadas a sus valores actuales, para determinar el grado de verdad de cada premisa de la regla.
2. Inferencia: se puede definir como un proceso de mapeo de una entrada a una salida, utilizando la teoría de conjuntos difusos.
3. Composición: todos los subconjuntos difusos asociados a cada variable de salida se combinan para formar un subconjunto difuso único de salida.
4. Defusificación (opcional): se emplea cuando es útil convertir el conjunto de salidas difusas a un número preciso (número duro). Las dos técnicas más comunes de defusificación son *Centroide Ponderado* y *Máximo*.

Algunos de los beneficios de los SED son reducir el costo de desarrollo de aplicaciones, los costos de ejecución y los costos de mantenimiento, ya que al emplear un número limitado de reglas para codificar el conocimiento de alto

nivel son menos propensos a errores, manejan información vaga, incierta e imprecisa y son mucho más fáciles de mantener.

2.3.1. Sistemas de control difuso

Una de las áreas de aplicación más exitosa de la lógica difusa es el control difuso. Los sistemas de éste se forman con reglas de inferencia difusa, donde cada regla tiene como antecedentes y consecuentes las condiciones de pertenencia de variables de estado y las acciones de control respectivamente.³² El primer sistema fue diseñado para mantener la presión de una caldera y la velocidad de su pistón constante en una máquina de vapor.

2.3.2. Mapas cognitivos difusos (Fuzzy Cognitive Maps, FCM)

Ron Axelrod introdujo los mapas cognitivos como una manera formal de representar el conocimiento científico social y modelar la toma de decisiones en sistemas sociales y políticos.³³ El objetivo principal de construir un mapa cognitivo en torno a un problema es predecir las decisiones que se tomarán, dejando que los temas relevantes interactúen entre sí. Estas predicciones se pueden usar para determinar si una decisión es consistente con toda la colección de afirmaciones causales declaradas en el sistema.

El término mapa cognitivo difuso (FCM) fue acuñado por Bart Kosko³⁴ para describir un modelo con dos características principales:

1. Las relaciones causales entre los nodos son difusas. En vez de usar solamente los signos para indicar causalidad positiva o negativa, se asocia un número a la relación para expresar el grado de proximidad entre dos conceptos.

³² Mc Neill, D., Freiburger P. **Fuzzy Logic**, Simon & Shuster, New York, 1993.

³³ Axelrod, R. **Structure of Decision**, Princeton University Press, US, 1976.

³⁴ Bart, K. (1986). **Fuzzy Cognitive Maps**, *Int. J. Man-Machine Studies* 24:65-75.

2. El sistema es dinámico con retroalimentación, por lo que cambiar un nodo afecta a otros, lo que a su vez puede afectar al nodo que inicia el cambio. La presencia de retroalimentación añade un aspecto temporal al funcionamiento de los FCM ya que su estructura se puede ver como una red neuronal artificial recurrente, donde los conceptos se representan por neuronas y las relaciones causales por enlaces ponderados o bordes que las conectan.

2.4. Algoritmos Evolutivos³⁵

Los Algoritmos Evolutivos (AE) son técnicas de solución de problemas de optimización y búsqueda que utilizan operadores de distintos tipos y que imitan procesos evolutivos como elementos clave para explorar y explotar los espacios de búsqueda. Ésta es una técnica de búsqueda basada en la teoría de la evolución de Darwin que ha cobrado tremenda popularidad alrededor del mundo durante los últimos años. Entre los algoritmos evolutivos más conocidos se encuentran los Algoritmos Genéticos, Programación Evolutiva, Estrategias Evolutivas y la Programación Genética. Los AE mantienen una población de individuos que evolucionan de acuerdo a reglas de selección y otros operadores, que se conocen como *operadores genéticos* como la *cruza*, *recombinación* y *mutación*.

El AE más conocido es el algoritmo genético. Básicamente, este algoritmo mantiene una “población” de soluciones del problema que se aborda. La idea es generar nuevas soluciones en un proceso iterativo que pretende converger a una solución que satisfaga cierto criterio de paro establecido por el usuario. Nuevas soluciones se generan mediante la combinación y alteración de las soluciones más “aptas”, donde la aptitud de las soluciones se evalúa mediante la función objetivo del problema bajo análisis.

³⁵ Para mayor información ver capítulo de Cómputo Evolutivo en este texto.

2.5. Aplicaciones

En México se han desarrollado una gran cantidad de aplicaciones que utilizan técnicas de Aprendizaje e Inteligencia Computacional, como sigue:

2.5.1. Medicina

Un área en donde se han realizado varios desarrollos es la de análisis de llanto de bebés. En particular, una de estas aplicaciones selecciona automáticamente un modelo para clasificar patrones de llanto infantil con fines de diagnóstico.³⁶ La selección se hace por medio de un algoritmo genético y permite generar modelos personalizados para el tipo de muestras de llantos infantiles. Como parte de un proyecto integral para el análisis y clasificación de llanto infantil se han desarrollado otros sistemas con computación suave³⁷ donde se comparan sistemas evolutivos con redes neurales para clasificar patologías a partir del llanto infantil. En otro trabajo³⁸ se presenta un sistema híbrido que consiste en una Máquina de Soporte Vectorial Difusa dedicado al reconocimiento automático del llanto de bebé; en otro más del mismo grupo se implementa una Red Neuronal Relacional Difusa para la detección de patologías a partir del llanto infantil.³⁹

³⁶ Rosales-Pérez, A., Reyes Garcia, C. A., Gonzalez, J. A., Reyes-Galaviz, O., Escalante, H. J., Orlandi, S. (2105). **Classifying Infant Cry Patterns by the Genetic Selection of a Fuzzy Model**, *Biomedical Signal Processing and Control*, 17, pp. 38-46.

³⁷ Reyes Galaviz, O. F., Reyes Garcia, C. A. (2005). **Infant Cry Classification to Identify Hypo Acoustics and Asphyxia comparing an Evolutionary-Neural System with a Neural Network System**. En *LNAI 3789: Advances in Artificial Intelligence* Gelbukh, A. Alvaro de de Albornoz, A., Terashima-Marin, H. (Eds.). Springer, Berlin. pp. 949-958.

³⁸ Barajas-Montiel, S. E., Reyes-García, C. A. (2006). **Fuzzy Support Vector Machines for Automatic Infant Cry Recognition**. En *LNCIS 345*, D. S. Huang, K. Li, G.W. Irwin (Eds.). Springer, Berlin, pp. 876-881.

³⁹ Suaste-Rivas, I., Reyes-Galaviz, O. R., Diaz-Mendez, A., Reyes-Garcia, C. A. (2004). **Implementation of a Linguistic Fuzzy Relational Neural Network for Detecting Pathologies by Infant Cry Recognition**. En *LNCS 3315: Advances in Artificial Intelligence*, Edited by Christian Lemaitre, Carlos A. Reyes and Jesus Gonzalez, Springer, Berlin, pp. 953-962.

Con el desarrollo y utilización masiva de teléfonos celulares cada vez más inteligentes, recientemente se han desarrollado una gran diversidad de aplicaciones. Dentro del ámbito médico se han desarrollado sistemas capaces de predecir el estado de ánimo de un paciente bipolar⁴⁰ o el nivel de estrés de un trabajador exclusivamente con la información que generan los sensores del teléfono.⁴¹ La aplicación a pacientes bipolares es importante porque en estados de fuerte depresión los pacientes pueden cometer suicidio. La segunda aplicación ha demostrado su utilidad ya que el estrés percibido de las personas está asociado con problemas de salud. En ambas aplicaciones se utilizaron técnicas de aprendizaje computacional, como clasificadores, clasificación semi-supervisada y aprendizaje por transferencia.

Otras aplicaciones en salud tienen que ver con el análisis de imágenes médicas y las series de tiempo. Entre éstas, se puede mencionar la segmentación de imágenes de Cervix mediante AE;⁴² la generación automática de modelos para clasificar tipos y subtipos de leucemia a partir de imágenes de médula;⁴³ la identificación de episodios epilépticos a partir de señales de electroencefalogramas,⁴⁴ entre otras.

⁴⁰ Maxhuni, A. Muñoz-Meléndez, A., Osmani, V., Perez, H., Mayora, O., Morales, E. F. (2016). **Classification of bipolar disorder based on analysis of voice and motor activity of patients.** *Pervasive and Mobile Computing* 31: 50-66.

⁴¹ Maxhuni, A., Hernandez-Leal, P., Sucar, L. E. Osmani, V., Morales, E. F. Mayora, O. (2016). **Stress Modelling and Prediction in Presence of Scarce Data.** *Journal of Biomedical Informatics* 63: 344-356.

⁴² Marquez-Grajales, A., Acosta-Mesa, H. G., Mezura-Montes, E., Hernández-Jiménez, R. (2016). **Cervical image segmentation using active contours and evolutionary programming over temporary acetowhite patterns.** *Proceedings of CEC 2016*: 3863-3870.

⁴³ Escalante, H. J., Montes-y-Gómez, M., González, J. A., Gómez-Gil, P., Leopoldo Altamirano Robles, L., Reyes García, C. A., Reta, C., Rosales-Pérez, A. (2012). **Acute leukemia classification by ensemble particle swarm model selection.** *Artificial Intelligence in Medicine* 55(3): 163-175.

⁴⁴ Gómez-Gil, P., Juárez-Guerra, E., Alarcón Aquino, V., Ramírez-Cortés, Rangel-Magdaleno, J. J. (2014). **Identification of Epilepsy Seizures Using Multi-resolution Analysis and Artificial Neural Networks.** *Recent Advances on Hybrid Approaches for Designing Intelligent Systems*, 337-351.

2.5.2. Industria

En México se han desarrollado aplicaciones para tratar de mejorar los procesos de producción de tubos. Por ejemplo, para optimizar el proceso de fabricación para que los tubos cumplan con los requerimientos mecánicos, como niveles de dureza. Igualmente, se han aplicado exitosamente técnicas de inteligencia y aprendizaje computacional para detectar fallas en baleros incrustados en electrodomésticos.⁴⁵ Otro aspecto importante ha sido la creación de herramientas de limpieza de datos⁴⁶ que pueden usarse en distintas aplicaciones.

El Sector Energético es otra área en donde es posible aplicar una gran cantidad de técnicas de Aprendizaje e Inteligencia Computacional. En particular, se desarrolló una metodología híbrida basada en regresión vectorial y algoritmos genéticos para la predicción de la velocidad del viento.⁴⁷ De igual manera se desarrolló un sistema inteligente de soporte a la operación de plantas de generación eléctrica.⁴⁸ Éste es un sistema de soporte para la toma de decisiones (DSS) basado en un formalismo conocido como redes bayesianas con nodos temporales (Temporal Nodes Bayesian Network: TNBN).

⁴⁵ Saucedo-Espinosa, M. A., Escalante, H. J., Berrones, A. (2017). **Detection of defective embedded bearings by sound analysis: a machine learning approach.** *J. Intelligent Manufacturing* 28(2): 489-500.

⁴⁶ Ibargüengoytia, P. H., García, U.A., Herrera-Vega, J., Hernández-Leal, P., Morales, E. F. Sucar, L. E., Orihuela-Espina, F. (2013). **On the Estimation of Missing Data in Incomplete Databases: Autoregressive Bayesian Networks.** En *Proc. of the Eighth International Conference on Systems ICONS-2013*, pp. 111-116.

⁴⁷ G. Santamaría-Bonfil, A. Reyes-Ballesteros, C. Gershenson. (2016). **Wind speed forecasting for wind farms: A method based on support vector regression.** *Renewable Energy* 85:790-809.

⁴⁸ Arroyo, G., Sucar, L. E. **Sistema Inteligente de Soporte a la operación de Plantas de Generación Eléctrica.** Por aparecer en *Komputer Sapiens*.

2.5.3. Sociedad

Los sistemas de Aprendizaje e Inteligencia Computacional pueden aplicarse a problemas que beneficien en general a la sociedad. Por ejemplo, se desarrolló un sistema de inferencia difuso entrenado como red neuronal dedicado al reconocimiento de emociones a partir de la voz. Este sistema es la base de una aplicación comercial de apoyo a *call centers*.⁴⁹ Otra aplicación consiste en predecir el ozono en la Ciudad de México a partir de información de las estaciones de monitoreo ambiental.⁵⁰ En una línea diferente de aplicación se desarrollaron algoritmos dedicados a mejorar la calidad de transporte de los individuos, capaces de aprender patrones de baches y otras anomalías de la superficie de la carretera, cuya representación utiliza bolsa de palabras (*bag of words*).^{51,52}

2.5.4. Interfaces

Se han desarrollado una serie de sistemas basados en modelos de inteligencia computacional dedicados al procesamiento de diferentes tipos de bioseñales y al posterior reconocimiento de patrones o a su clasificación. Entre éstos se encuentran trabajos para clasificar automáticamente el habla imaginada en un ambiente multi-objetivo a partir de electroencefalogramas (EEG) utilizan-

⁴⁹ Pérez-Espinosa, H., Reyes-García, C. A., Villaseñor-Pineda, L. **Acoustic feature selection and classification of emotions in speech using a 3D continuous emotion model.** *Biomedical Signal Processing and Control*, Elsevier Ltd, Electronic Publication, 7(1):79-87, January 2012, doi: 10.1016/j.bspc.2011.02.008.

⁵⁰ Sucar, E., Perez-Brito, J., Ruiz-Suarez, C., Morales, E. (1997). **Learning Structure from Data and its Application to Ozone Prediction.** *Applied Intelligence* 7(4): 327-338.

⁵¹ González L. C., Moreno, R., Escalante, H. J., Martínez, F., Carlos, M. R. (2017). **Learning Roadway Surface Disruption Patterns Using the Bag of Words Representation.** Por aparecer en *IEEE Transactions on Intelligent Transportation Systems*.

⁵² Carlos, M. R., Aragón, M. E., González, L. C., Escalante, H. J., Martínez, F. **Evaluation of detection approaches for Road Anomalies based on accelerometer readings—Addressing who’s who.** En revisión en *IEEE Transactions on Intelligent Transportation Systems*.

do sistemas de inferencia difusa.⁵³ La disciplina en que se desarrollan estos modelos se conoce como Interfaces Cerebro-Computadora (*Brain Computer Interfaces*).

2.6. Investigación en México

Existen diferentes grupos de investigación en México dedicados al Aprendizaje e Inteligencia Computacional. Entre éstos destacan el Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE), Puebla, con sus laboratorios de Aprendizaje Computacional y Reconocimiento de Patrones y de Procesamiento de Bioseñales y Computación Médica, en los que se realiza investigación básica y aplicada; el Instituto Tecnológico de Estudios Superiores de Monterrey (ITESM) con su grupo de Investigación en Sistemas Inteligentes, enfocado a la optimización y logística, inteligencia ambiental, semántica Web, salud, previsión e inteligencia de negocios, entre otros, y su grupo de Investigación en Modelos de Aprendizaje Computacional, enfocado a la innovación de técnicas estadísticas para el reconocimiento de patrones y de técnicas de aprendizaje de modelos computacionales (*machine learning*), así como de modelos de visualización, en aplicaciones como seguridad informática, integridad personal y resiliencia personal; el Instituto Tecnológico de Culiacán enfocado al desarrollo de Sistemas Tutoriales Inteligentes, que ha impulsado el *Workshop on Intelligent Learning Environments WILE* (Taller de Ambientes Inteligentes de Aprendizaje); la Universidad Veracruzana con su cuerpo académico de Investigación y Aplicaciones de la Inteligencia Artificial, enfocado al descubrimiento de conocimiento en datos, sistemas multi-agentes, procesamiento digital de imágenes y cómputo evolutivo; la Universidad Panamericana con su grupo de Visión, Máquinas y Señales, que realiza investigación aplicada en procesamiento de imágenes, señales biológicas y áreas médicas, diseño de algoritmos de aprendizaje y sistemas de control robóticos; el Insti-

⁵³ Torres-García, A. A., Reyes-García, C. A., Villaseñor-Pineda, Gregorio García-Aguilar, G. (2016). **Implementing a fuzzy inference system in a multi-objective EEG channel selection model for imagined speech classification.** *Expert Systems With Applications* 59:1–12.

tuto Tecnológico de Tijuana con sus laboratorios Sistemas Híbridos Difusos Inteligentes y Computación Evolutiva, y el Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas de la UNAM, donde se hace investigación en Computación Evolutiva, Aprendizaje aplicado a la Interacción Humano-Computadora y aprendizaje de máquina aplicado al procesamiento de imágenes y lenguaje natural.^{54,55} Otros grupos destacados son el grupo de Optimización Evolutiva Multi-Objetivo del CINVESTAV; el laboratorio Evovisión del CICESE, que realiza investigación en visión computacional mediante algoritmos evolutivos, el grupo de Análisis de datos multidimensionales y reconocimiento de patrones del Centro de Investigación en Matemáticas (CIMAT) en Guanajuato y el laboratorio de Big DATA del INFOTEC.

2.7. Conclusiones

Actualmente se vive un creciente interés en el área de Inteligencia Artificial y, en particular, en los sistemas de aprendizaje e inteligencia computacional. Esto se debe en parte a los avances recientes del área, impulsados por la gran cantidad de datos generados y disponibles para su análisis, los avances en las computadoras para procesar cada vez más rápidamente información y en las aplicaciones comerciales que se están desarrollando utilizando algoritmos maduros de aprendizaje e inteligencia artificial. La capacidad “inteligente de las cosas y sistemas” será determinada por su nivel de Coeficiente Intelectual Máquina (MIQ) que se convertirá en una de las especificaciones más importantes para seleccionar productos. En general se espera un futuro con inteligencia incremental en muchas áreas, como ciudades inteligentes, carreteras inteligentes, edificios inteligentes, vehículos autónomos extremadamente inteligentes, sistemas administrativos, educativos, legales y gubernamentales

⁵⁴ Gibran Fuentes-Pineda, G., Koga, H., Watanabe, T. (2011). **Scalable Object Discovery: A Hash-based Approach to Clustering Co-occurring Visual Words**. *Transactions on Information and Systems*, E94-D(10):2024–2035.

⁵⁵ Gibran Fuentes-Pineda, G., Meza-Ruiz, I. V. (2015). **Sampled Weighted Min-Hashing for Large-Scale Topic Mining**. *Proceedings of the Mexican Conference on Pattern Recognition*, Vol. 9116, pp. 203-213.

inteligentes, así como variantes de teléfonos, televisores y computadores inteligentes. Algunos autores creen que se va a lograr crear una super-inteligencia y se están cuestionando sus posibles consecuencias para la humanidad, desde los enfoques pesimistas, que vaticinan el fin de la raza humana, hasta los optimistas, que predicen un futuro en donde los problemas de salud, alimentación, energía y contaminación, por mencionar algunos, se van a resolver definitivamente con la ayuda de esta super-inteligencia.