1 Aprendizaje Automático o Machine Learning

En 1959 Arthur Samuel en una publicación escribió: “Programming computers to learn from experience should eventually eliminate the need for much of this detailed programming effort” [Arthur1959]. Lo que nos lleva a pensar que uno pioneros del aprendizaje automático ya dejaba visualizar que los programas, a partir del aprendizaje sobre los datos históricos (la experiencia), podrían efectuar tareas de toma de decisiones sin ser programadas explícitamente dichas decisiones. Samuel define al aprendizaje automático como sigue: “El aprendizaje automático es un campo de estudio que da a las computadoras la capacidad de aprender sin ser explícitamente programadas”.

Otro investigador de aprendizaje automático Tom Mitchell propuso en 1998 la siguiente definición: “Well posed Learning Problem: A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E”. Donde se nos indica que el aprendizaje en las máquinas deberá ser parecido al aprendizaje en los humanos, por ejemplo cuando una criatura comienza a hablar a través de la experiencia de pronunciar las palabras y de su interacción con otras personas, entonces sucede que su capacidad de hablar se va perfeccionando o mejorando.

1.1 Definición

“The purpose of machine learning is to learn from training data in order to make as good as possible predictions on new, unseen, data”[Jean2016].

La dificultad radica en que debemos construir modelos que nos acerquen a una buan predicción sobre datos aún no conocidos o imprevistos.

1.2 Categoría de algoritmos

Los algoritmos de aprendizaje automático se pueden categorizar según la forma en que se realiza el aprendizaje, pero teniendo en cuenta que todos reciben un conjunto de ejemplos para aprender desde los mismos.

1.2.1 Aprendizaje supervisado (supervised learning): El algoritmo recibe datos de entrenamiento que contienen la respuesta correcta para cada ejemplo.

1.2.2 Aprendizaje no supervisado (unsupervised learning): El algoritmo busca estructuras en los datos de entrenamiento, como encontrar qué ejemplos son similares entre sí, y agruparlos en clusters.

1.3 Tipos de problemas

Teniendo en cuenta las clases de problemas que los algoritmos de aprendizaje pueden resolver, los tipos de problemas se pueden agrupar como sigue.

1.3.1 Regresión: Un problema de aprendizaje supervisado donde la respuesta a aprender es un valor continuo.

1.3.2 Clasificación: Un problema de aprendizaje supervisado donde la respuesta a aprender es un valor de un conjunto finito de posibles valores discretos. Classifiation learning is sometimes called supervised, because, in a sense, the scheme operates under supervision by being provided with the actual outcome for each of the training examples

1.3.3 Segmentación: Un problema de aprendizaje no supervisado donde la estructura a aprender es un conjunto de clusters donde cada cluster tiene similares ejemplos.

1.3.4 Análisis de red: Un problema de aprendizaje no supervisado donde la estructura a aprender es información acerca de la importancia y el rol de los nodos en una red.

1.4 Componentes esenciales

1.4.1 Ejemplos o instancias (examples): La entrada de un esquema de aprendizaje automático es un conjunto de instancias. Estas instancias son las cosas que deben ser clasificadas, asociadas o agrupadas. En el escenario estándar, cada instancia es un ejemplo individual e independiente del concepto que se debe aprender.

1.4.2 Características o atributos (features): Las instancias son caracterizadas mediante los valores de un conjunto predeterminado de atributos. Cada instancia proporciona una entrada al aprendizaje automático es caracterizado por los valores en un conjunto fijo y predefinido de características o atributos [DM2011].

1.4.3 Etiquetas (labels): Las cantidades nominales tienen valores que son símbolos distintos. Los valores mismos sirven como etiquetas o nombres, de ahí el término nominal, que viene de la palabra latina para nombre. Los atributos nominales a veces se llaman categorizados, enumerados o discretos.

1.4.4 Conjunto de entrenamiento (training set):

1.4.5 Algoritmos de aprendizaje (learning algorithms): Hipótesis, Parámetros, Función de costo, Objetivo.

1.4.6 Conjunto de prueba (test set): Para predecir el rendimiento de un clasificador sobre nuevos datos, necesitamos evaluar su tasa de error en un conjunto de datos que no desempeñó ningún papel en la formación del clasificador. Este conjunto de datos independiente se denomina conjunto de prueba.

1.5 El problema de la clasificación

En los problemas de clasificación el modelo creado debe predecir la clase, tipo o categoría de la salida.

1.5.1 Clasificación binaria (binary classification): En su forma más simple se reduce a la pregunta: dado un patrón x extraído de un dominio X, estimar qué valor asumirá una variable aleatoria binaria asociada y ∈ {± 1} [IntroML2008].

1.5.2 Clasificación multiclase (multiclass classification): Es la extensión lógica de la clasificación binaria. La principal diferencia es que ahora y ∈ {1, ... , N} puede asumir un rango de valores diferentes [IntroML2008].

1.6 Algoritmos de clasificación en WEKA

Weka es una colección de algoritmos de aprendizaje automático para tareas de minería de datos. Los algoritmos pueden ser aplicados directamente a un conjunto de datos o llamados desde código Java. Weka contiene herramientas para pre-procesamiento de datos, clasificación, regresión, clustering, reglas de asociación y visualización. También es adecuado para desarrollar nuevos esquemas de aprendizaje automático [Weka3]. Los algoritmos de clasificación de Weka que se utilizarán es el siguiente [WekaCla]:

1.6.1 BayesNet: Bayes Network learning using various search algorithms and quality measures. Base class for a Bayes Network classifier. Provides datastructures (network structure, conditional probability distributions, etc.) and facilities common to Bayes Network learning algorithms like K2 and B.

1.6.2 NaiveBayes: Class for a Naive Bayes classifier using estimator classes. Numeric estimator precision values are chosen based on analysis of the training data. For this reason, the classifier is not an UpdateableClassifier (which in typical usage are initialized with zero training instances).

1.6.3 NaiveBayesUpdateable: Class for a Naive Bayes classifier using estimator classes. This is the updateable version of NaiveBayes. This classifier will use a default precision of 0.1 for numeric attributes when buildClassifier is called with zero training instances.

1.6.4 Logistic: Class for building and using a multinomial logistic regression model with a ridge estimator. If there are k classes for n instances with m attributes, the parameter matrix B to be calculated will be an m\*(k-1) matrix.

1.6.5 MultilayerPerceptron: A Classifier that uses backpropagation to classify instances. This network can be built by hand, created by an algorithm or both. The network can also be monitored and modified during training time. The nodes in this network are all sigmoid (except for when the class is numeric in which case the the output nodes become unthresholded linear units).

1.6.6 SimpleLogistic: Classifier for building linear logistic regression models. LogitBoost with simple regression functions as base learners is used for fitting the logistic models. The optimal number of LogitBoost iterations to perform is cross-validated, which leads to automatic attribute selection.

1.6.7 SMO: Implements John Platt's sequential minimal optimization algorithm for training a support vector classifier. This implementation globally replaces all missing values and transforms nominal attributes into binary ones. It also normalizes all attributes by default. (In that case the coefficients in the output are based on the normalized data, not the original data --- this is important for interpreting the classifier). Multi-class problems are solved using pairwise classification (aka 1-vs-1). To obtain proper probability estimates, use the option that fits calibration models to the outputs of the support vector machine. In the multi-class case, the predicted probabilities are coupled using Hastie and Tibshirani's pairwise coupling method.

1.6.8 OneR: Class for building and using a 1R classifier; in other words, uses the minimum-error attribute for prediction, discretizing numeric attributes.

1.6.9 DecisionTable: Class for building and using a simple decision table majority classifier.

1.6.10 JRip: This class implements a propositional rule learner, Repeated Incremental Pruning to Produce Error Reduction (RIPPER), which was proposed by William W. Cohen as an optimized version of IREP.

1.6.11 PART: Class for generating a PART decision list. Uses separate-and-conquer. Builds a partial C4.5 decision tree in each iteration and makes the "best" leaf into a rule.

1.6.12 ZeroR: Class for building and using a 0-R classifier. Predicts the mean (for a numeric class) or the mode (for a nominal class).

1.6.13 DecisionStump: Class for building and using a decision stump. Usually used in conjunction with a boosting algorithm. Does regression (based on mean-squared error) or classification (based on entropy). Missing is treated as a separate value.

1.6.14 J48: Class for generating a pruned or unpruned C4.5 decision tree.

1.6.15 LMT: Classifier for building 'logistic model trees', which are classification trees with logistic regression functions at the leaves. The algorithm can deal with binary and multi-class target variables, numeric and nominal attributes and missing values.

1.6.16 RandomForest: Class for constructing a forest of random trees.

1.6.17 RandomTree: Class for constructing a tree that considers K randomly chosen attributes at each node. Performs no pruning. Also has an option to allow estimation of class probabilities (or target mean in the regression case) based on a hold-out set (backfitting).

1.6.18 REPTree: Fast decision tree learner. Builds a decision/regression tree using information gain/variance and prunes it using reduced-error pruning (with backfitting). Only sorts values for numeric attributes once. Missing values are dealt with by splitting the corresponding instances into pieces (i.e. as in C4.5).

1.7 Evaluación de lo aprendido

La evaluación es la clave para lograr avances reales en el aprendizaje automático.

1.7.1 Validación Cruzada (cross-validation): En la validación cruzada, usted decide sobre un número fijo de pliegues, o particiones, de los datos. Supongamos que usamos tres. Luego los datos se dividen en tres particiones aproximadamente iguales; cada uno a su vez se utiliza para las pruebas y el resto se utiliza para el entrenamiento. Es decir, utilizar dos tercios de los datos para el entrenamiento y un tercio para las pruebas, y repetir el procedimiento tres veces para que al final, cada instancia se haya utilizado exactamente una vez para la prueba. Esto se denomina triple validación cruzada, y si la estratificación se adopta también, lo que es a menudo, triple validación cruzada estratificada.

1.7.2 Validación Cruzada K-fold Estratificado (stratified k-fold cross validation): La manera estándar de predecir la tasa de error de una técnica de aprendizaje dada una única muestra fija de datos es usar la validación cruzada diez veces estratificada. Los datos se dividen aleatoriamente en 10 partes en las que la clase se representa en aproximadamente las mismas proporciones que en el conjunto de datos completo. Cada parte se extiende a su vez y el esquema de aprendizaje entrenado en los restantes nueve décimos; Entonces su tasa de error se calcula en el conjunto de retención. Así, el procedimiento de aprendizaje se ejecuta un total de 10 veces en diferentes conjuntos de entrenamiento (cada conjunto tiene mucho en común con los demás). Finalmente, las 10 estimaciones de error se promedian para obtener una estimación del error global. Pruebas extensas en numerosos conjuntos de datos diferentes, con diferentes técnicas de aprendizaje, han demostrado que 10 es sobre el número correcto de pliegues para obtener la mejor estimación de error, y también hay algunas pruebas teóricas que apoya esto. Aunque estos argumentos no son en absoluto concluyentes, y el debate continúa enfurecido en los círculos de aprendizaje automático y de minería de datos sobre cuál es el mejor esquema de evaluación, la validación cruzada diez veces se ha convertido en el método estándar en términos prácticos. Las pruebas también han demostrado que el uso de la estratificación mejora ligeramente los resultados. Por lo tanto, la técnica de evaluación estándar en situaciones en las que sólo se dispone de datos limitados es la validación cruzada diez veces estratificada. La estratificación reduce la variación, ciertamente no la elimina completamente. Cuando se busca una estimación exacta del error, es un procedimiento estándar repetir el proceso de validación cruzada 10 veces, es decir, diez veces la validación cruzada diez veces, y el promedio de los resultados. Esto implica invocar el algoritmo de aprendizaje 100 veces en conjuntos de datos que son todas las nueve décimas del tamaño del original. Obtener una buena medida de rendimiento es una empresa de computación intensiva.

1.7.3 Percentage split:

1.8 Resultados de la evaluación

Para los problemas de clasificación, es natural medir el rendimiento de un clasificador en términos de la tasa de error (error rate). El clasificador predice la clase de cada instancia: si es correcta, se cuenta como un éxito; sino, es un error. La tasa de error es sólo la proporción de errores cometidos sobre un conjunto de instancias, y mide el rendimiento general del clasificador. Por supuesto, lo que nos interesa es el probable desempeño futuro en nuevos datos, no el rendimiento pasado en datos antiguos. Ya sabemos las clasificaciones de cada instancia en el conjunto de entrenamiento, que después de todo es por qué podemos usarlo para el entrenamiento. La tasa de error en el conjunto de entrenamiento no es probable que sea un buen indicador de rendimiento futuro debido a que el clasificador se ha aprendido de los mismos datos de entrenamiento, cualquier estimación de rendimiento basada en esos datos será optimista, incluso excesivamente optimista.

La tasa de error en los datos de entrenamiento se llama error de resustitución porque se calcula resusstituyendo las instancias de entrenamiento en un clasificador que se construyó a partir de ellas. Para predecir el rendimiento de un clasificador en nuevos datos, necesitamos evaluar su tasa de error en un conjunto de datos que no desempeñó ningún papel en la formación del clasificador. Este conjunto de datos independiente se denomina conjunto de pruebas.

En tales situaciones, la gente suele hablar de tres conjuntos de datos: los datos de entrenamiento, los datos de validación y los datos de prueba. Los datos de entrenamiento son utilizados por uno o más esquemas de aprendizaje para conocer clasificadores. Los datos de validación se utilizan para optimizar los parámetros de los clasificadores, o para seleccionar uno determinado. A continuación, los datos de prueba se utilizan para calcular la tasa de error del método final optimizado. Cada uno de los tres conjuntos debe ser independiente: El conjunto de validación debe ser diferente del conjunto de entrenamiento para obtener un buen desempeño en la etapa de optimización o selección y el conjunto de pruebas debe ser diferente de ambos para obtener una estimación confiable de la tasa de error real.

Generalmente, cuanto mayor es la muestra de entrenamiento, mejor es el clasificador, aunque los retornos comienzan a disminuir una vez que se sobrepasa un cierto volumen de datos de entrenamiento. Y cuanto mayor es la muestra de prueba, más precisa es la estimación del error. La precisión de la estimación del error puede ser cuantificada estadísticamente.

El verdadero problema ocurre cuando no hay una gran cantidad de datos disponibles. Las secciones 5.3 y 5.4 revisan métodos ampliamente utilizados para hacer frente a este dilema.

En términos prácticos, es común tener un tercio de los datos para la prueba y utilizar los dos tercios restantes para el entrenamiento.

En general, no se puede saber si una muestra es representativa o no. Pero hay una comprobación simple que puede ser útil: Cada clase en el conjunto de datos completo debe estar representada en la proporción correcta en los conjuntos de entrenamiento y prueba. Deberían estar representados en la proporción adecuada en los conjuntos de entrenamiento y prueba. Si, por mala suerte, todos los ejemplos con una cierta clase se omitieran en el conjunto de entrenamiento, difícilmente podría esperar que una clase aprendiera de esos datos para funcionar bien en los ejemplos de esa clase.

Debe asegurarse de que el muestreo aleatorio se realiza de una manera que garantiza que cada clase esté representada correctamente en los conjuntos de entrenamiento y prueba. Este procedimiento se denomina estratificación, y hablamos de la retención estratificada (stratified holdout). Aunque en general vale la pena hacerlo, la estratificación sólo proporciona una salvaguardia primitiva contra la representación desigual en los conjuntos de entrenamiento y pruebas.

Una manera más general de mitigar cualquier sesgo causado por la muestra particular escogida para retener, es repetir todo el proceso, entrenamiento y prueba, varias veces con diferentes muestras aleatorias. En cada iteración, una cierta proporción, digamos dos tercios, de los datos se selecciona al azar para el entrenamiento, posiblemente con estratificación, y el resto se utiliza para la prueba. Las tasas de error en las diferentes iteraciones se promedian para obtener una tasa de error global. Este es el método de retención repetida de la estimación de la tasa de error.

PARA COMPARAR varios esquemas de aprendizaje automático: Este es un trabajo para una prueba estadística basada en límites de confianza, el tipo que conocimos anteriormente al tratar de predecir el rendimiento real de una determinada tasa de error de prueba. Lo que queremos determinar es si un esquema es mejor o peor que otro en promedio, en todos los posibles conjuntos de datos de entrenamiento y prueba que se pueden extraer del dominio. Debido a que la cantidad de datos de entrenamiento afecta naturalmente el rendimiento, todos los conjuntos de datos deben ser del mismo tamaño. De hecho, el experimento podría repetirse con diferentes tamaños para obtener una curva de aprendizaje. For each learning scheme we can draw several datasets of the same size, obtain an accuracy estimate for each dataset using cross-validation, and compute the mean of the estimates. Each cross-validation experiment yields a different, independent error estimate. What we are interested in is the mean accuracy across all possible datasets of the same size, and whether this mean is greater for one scheme or the other. This is a job for a statistical device known as the T-TEST, or Student’s t-test. Because the same cross-validation experiment can be used for both learning schemes to obtain a matched pair of results for each dataset, a more sensitive version of the t-test known as a paired t-test can be used.

1.8.1 correct() - number of correctly classified instances.

1.8.2 pctCorrect() - percentage of correctly classified instances.

1.8.3 kappa() - KAPPA STATISTIC: CONTAR EL COSTO o COUNTING THE COST: Las evaluaciones que se han discutido hasta ahora no tienen en cuenta el costo de tomar decisiones equivocadas, clasificaciones erróneas. Otros ejemplos en los que los errores cuestan diferentes cantidades incluyen las decisiones de préstamo: El costo de prestar a un deudor es mucho mayor que el costo de negocio perdido de rechazar un préstamo a un no deudor. Y la detección de mancha de aceite: El costo de no detectar una mancha real que amenaza el medio ambiente es mucho mayor que el costo de una falsa alarma. El costo de identificar malos problemas con una máquina que resulta estar libre de fallas es menor que el costo de pasar por alto los problemas con uno que está a punto de fallar. Los verdaderos positivos (TP) y negativos verdaderos (TN) son clasificaciones correctas. Un falso positivo (PF) es cuando el resultado se predice incorrectamente como sí (o positivo) cuando es realmente no (negativo). Un falso negativo (FN) es cuando el resultado se predice incorrectamente como negativo cuando es realmente positivo. La tasa positiva verdadera es TP dividida por el número total de positivos, que es TP + FN; La tasa de falsos positivos es FP dividida por el número total de negativos, que es FP + TN. La tasa de éxito global es el número de clasificaciones correctas dividido por el número total de clasificaciones: TP +TN/ TP+ TN +FP +FN, por último, la tasa de error es 1 menos esto. En la predicción multiclase, cada elemento de matriz muestra el número de ejemplos de prueba para los que la clase real es la fila y la clase prevista es la columna. Los buenos resultados corresponden a grandes números en la diagonal principal y pequeños, idealmente cero, fuera de los elementos diagonales. Una medida denominada KAPPA STATISTIC tiene en cuenta este factor previsto deduciéndolo de los éxitos del predictor y expresando el resultado como una proporción del total para un predictor perfecto. El valor máximo de Kappa es 100%, y el valor esperado para un predictor aleatorio con los mismos totales de columna es 0. En resumen, la estadística Kappa se utiliza para medir el acuerdo entre categorizaciones predichas y observadas de un conjunto de datos, mientras se corrige un acuerdo que se produce por casualidad. Sin embargo, al igual que la tasa de éxito simple, no toma en cuenta los costos.

1.8.4 areaUnderROC(int classIndex) - CURVAS ROC o ROC Curves: Están estrechamente relacionados con una técnica gráfica para evaluar los esquemas de minería de datos conocidos como curvas ROC, que se utilizan en la misma situación, donde el alumno está tratando de seleccionar muestras de instancias de prueba que tienen una alta proporción de positivos. El acrónimo significa la característica de funcionamiento del receptor (Receiver Operating Characteristic), un término usado en la detección de señal para caracterizar la compensación entre la tasa de acierto y la tasa de falsa alarma sobre un canal ruidoso. Representan la tasa de positivos verdaderos en el eje vertical con respecto a la tasa de verdaderos negativos en el eje horizontal. El primero es el número de positivos incluidos en la muestra, expresado como un porcentaje del número total de positivos (TP Rate = 100 × TP/(TP + FN)); el último es el número de negativos incluidos en la muestra, expresado como porcentaje del número total de negativos (FP Rate = 100 × FP/(FP + TN)). El eje vertical es el mismo que el gráfico de elevación excepto que se expresa como un porcentaje. El eje horizontal es ligeramente diferente: es el número de negativos en lugar del tamaño de la muestra. Each point corresponds to drawing a line at a certain position on the ranked list, counting the yes’s and no’s above it, and plotting them vertically and horizontally, respectively. Esto es sólo una forma de usar la validación cruzada para generar curvas ROC. Un enfoque más simple es recopilar las probabilidades predichas para todos los conjuntos de prueba diferentes (de los cuales hay 10 en un 10-fold validación cruzada), junto con las etiquetas de clase verdadera de las instancias correspondientes, y generar una única lista rankeada basada en estos datos. Si el esquema de aprendizaje no permite ordenar las instancias, primero puede hacer que sea sensible al costo como se describió anteriormente. Para cada fold de una 10-fold validación cruzada, ponderar las instancias para una selección de diferentes ratios de coste, entrenar el esquema en cada conjunto ponderado, contar los verdaderos positivos y falsos positivos en el conjunto de pruebas y trazar el punto resultante en los ejes ROC .

1.8.5 recall(int classIndex) y precision(int classIndex) - RECALL-PRECISION CURVES: La gente ha lidiado con la compensación fundamental ilustrada por los gráficos de elevación y las curvas ROC en una amplia variedad de dominios. Comparar un sistema que localiza 100 documentos, 40 de los cuales son relevantes, con otro que localiza 400 documentos, 80 de los cuales son relevantes. ¿Cual es mejor? La respuesta ahora debe ser obvia: depende del costo relativo de falsos positivos, documentos devueltos que no sean relevantes, y falsos negativos, documentos que son relevantes pero que no se devuelven. Los investigadores de recuperación de información definen parámetros llamados recall y precisión: RECALL = number of documents retrieved that are relevant / total number of documents that are relevant. PRECISION: number of documents retrieved that are relevant / total number of documents that are retrieved. Los expertos en recuperación de información usan curvas recall-precision que trazan una contra la otra, para diferentes números de documentos recuperados de la misma manera que las curvas ROC y los gráficos de elevación, excepto que debido a que los ejes son diferentes, las curvas son de forma hiperbólica y el punto de operación deseado está hacia la parte superior derecha. Table 5.7 summarizes the three different ways introduced for evaluating the same basic tradeoff; TP, FP, TN, and FN are the numbers of true positives, false positives, true negatives, and false negatives, respectively. Diferentes técnicas dan diferentes compensaciones, y se pueden trazar como líneas diferentes en cualquiera de estas gráficas. TABLA 5.7

1.8.6 areaUnderPRC(int classIndex) - Area Under Precision-Recall Curve:

1.8.7 Tasa de error (error rate): on the testing data. Predecir la tasa de error. Tenfold cross-validation is the standard way of measuring the error rate of a learning scheme on a particular dataset. The basic quality measure offered by the error rate is longer appropriate.