

<b>Nombre: Luis Alfredo León Villapún</b>	
<b>Matrícula: A01322275</b>	<b>Calificación:</b>

**Apegándome al Código de Ética de los Estudiantes del Tecnológico de Monterrey, me comprometo a que mi actuación en este examen esté regida por la honestidad académica. Este examen equivale al 20% del examen final.**

1. Definan de manera breve y concisa ¿Qué es machine learning? ¿Cuál es la diferencia entre aprendizaje supervisado y no supervisado? **(20 puntos)**

Machine Learning, o aprendizaje automático, podría definirse como lo cita Samuel: “El área de estudio que provee a las computadoras la habilidad de aprender sin ser explícitamente programadas”(Samuel, 1959).

El aprendizaje supervisado es aquel en el cual los elementos de la muestra tienen una etiqueta, por ende, se conoce la clase a la cual pertenece. Algunos algoritmos que se aplican con aprendizaje supervisado son algoritmos de regresión, árboles de decisión, KNN y Redes de Bayes.

El aprendizaje no supervisado es en el que el algoritmo no tiene conocimiento ni del número de clases ni del tipo de clase que tiene la muestra. Para estos casos tipo de aprendizaje son implementados algoritmos como el KMeans, SVD, y PCA.

2. ¿Cuál es el objetivo de inicializar de manera aleatoria las medias en el algoritmo de “K-means”? ¿De qué manera sabemos cuándo detener el algoritmo? **(20 puntos).**

Dado que el algoritmo K-Means no siempre converge a una solución óptima, es decir, tiende a caer en mínimos locales, toma importancia iniciar de manera aleatoria las medias en cada iteración, ya que de esta manera los clusters generados van a ser diferentes en cada ocasión, de manera que no siempre se caerá en mínimos locales, sino que da la oportunidad de caer en otros mínimos locales, o como se espera, llegar a una solución óptima.

El algoritmo puede detenerse cuando sus asignaciones de centroide en cada iteración ya no cambian.

3. Suponga que su algoritmo ha sido entrenado y prueban el error de entrenamiento contra el de validación (sobre un subconjunto de prueba) y obtienen la siguientes gráficas de la Fig. 1: **(20 puntos)**

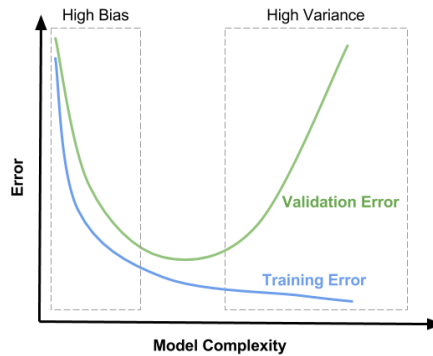


Fig. 1 Error de validación (verde) vs error de entrenamiento (azul).

- a) Suponga que su algoritmo se encuentra en la región de alta varianza (rectángulo punteado derecho en Fig. 1). Lo que sugiere una alta complejidad del mismo ¿Qué realizaría para reducir este “sobreajuste”?

Tener una alta varianza significa que al cambiar diferentes sets de entrenamiento, la función objetivo de nuestro algoritmo va a cambiar radicalmente, de modo que lo que se busca es reducir este “ruido” en el algoritmo para que la función objetivo no cambie tanto.

Para reducir el overfitting del algoritmo, generaría más ejemplos de entrenamiento, de modo que el algoritmo tuviera más “casos de uso” de los cuales aprender y reducir la alta varianza que existe en este algoritmo.

- b) ¿Qué significa encontrarse en la región de algo sesgo (rectángulo punteado izquierdo en Fig. 1)?

La región de alto sesgo (high bias) es un indicador de que al momento de construir el algoritmo se asumieron muchas cosas, lo que conlleva a tener un algoritmo no muy flexible. En cambio, el bajo sesgo, indicaría una “generalización” al momento de construir la función objetivo.

4. ¿En PCA qué por qué se seleccionan los eigenvectores correspondientes a los eigenvalores más grandes ( $\Sigma \mathbf{v} = \lambda \mathbf{v}$ ) en una distribución de datos? ¿Y cuál es el objetivo de utilizarlos con la matriz de covarianza? **(20 puntos)**

Porque las clases representadas por estos eigenvectores usualmente son los que proveen la información más significativa en nuestro set de datos. Por ejemplo, si en un set con tres clases diferentes se obtiene que el eigenvalor de una de las tres es muy pequeño con respecto a los otros, se sabe que la información que provee ese dato es poco relevante, por ende se puede descartar y reducir el problema a uno de dos clases. El objetivo de utilizarlos en la matriz de covarianza es porque en esta matriz, el eigenvector más grande siempre va a apuntar en dirección a la clase con la mayor varianza, por ende, a una de las categorías de datos más importantes en el set. Asimismo, el segundo eigenvector más grande será ortogonal al primero, y apunta a la clase con la segunda mejor varianza.

5. Suponga que tiene un grafo obtenido de una imagen aérea, planeado por PRM y explorado por un método avaro, pero tiene las siguientes complicaciones vistas en la Fig. 2. **(20 puntos)**

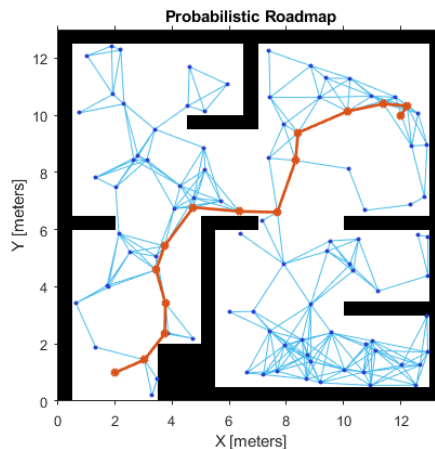


Fig. 2 Mapa planeado y generado (obtenido de Mathworks).

- a) El agente se mueve en los nodos intermedios de manera cercana a los obstáculos, a pesar de tener espacio libre ¿Qué modificaría del algoritmo para evitar esta decisión y correr riesgo innecesario?

Le agregaría una propiedad muy común en grafos: peso. Agregarle peso a cada nodo reduciría en gran medida esta problemática. Para colocarle peso a los nodos, habría que evaluar, dado un radio conveniente (como cuando se genera un PRM) cuál es el punto más cercano a un obstáculo dado un nodo. Si la distancia es muy pequeña, el nodo tendría mayor peso, y si la distancia es más grande, el nodo tendría menor peso. Así, el algoritmo de búsqueda preferiría caminos donde no se acerca tanto a un obstáculo, a menos que no tenga otra opción.

- b) Supongamos que el método de exploración avaro se cicla en un conjunto de nodos que de manera local presentan la mejor solución (mínimos locales) ¿Qué añadiría al algoritmo para evitar estos mínimos locales?

Le agregaría a cada nodo una variable de visitado, de manera que los nodos no puedan volver a otro donde ya han estado antes. De hecho, esta propiedad de la variable “visitado” es muy común para evitar ciclos en grafos, como por ejemplo al hacer BFS o DFS en un grafo. Esto, por supuesto, puede tener algunas desventajas porque de alguna manera descarta nodos que podrían llevar a una solución más óptima al momento de realizar la búsqueda.

## REFERENCIAS

Huss, M.. (2012). *Practical advice for machine learning: bias, variance and what to do next*. 4/Dic/2017, de Wordpress Sitio web: <https://followthedata.wordpress.com/2012/06/02/practical-advice-for-machine-learning-bias-variance/>

Brownlee, J.. (2016). *Gentle Introduction to the Bias-Variance Trade-Off in Machine Learning*. 4/Dic/2017, de Machine Learning Mastery Sitio web: <https://machinelearningmastery.com/gentle-introduction-to-the-bias-variance-trade-off-in-machine-learning/>

Spruyt, V.. (2014). *A geometric interpretation of the covariance matrix*. 4/Dic/2017, de Vision Dummy Sitio web: <http://www.visiondummy.com/2014/04/geometric-interpretation-covariance-matrix/>

Dallas, G.. (2013). *Principal Component Analysis 4 Dummies: Eigenvectors, Eigenvalues and Dimension Reduction*. 4/Dic/2017, de Wordpress Sitio web:

<https://georgemdallas.wordpress.com/2013/10/30/principal-component-analysis-4-dummies-eigenvectors-eigenvalues-and-dimension-reduction/>

Basurto, A. Cañete, P. León, L. Ororio, D. (2017). *Práctica 7: Clasificadores y reconocimiento de patrones*. 4/Dic/2017. En página 1. Puebla: ITESM.