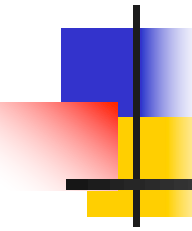
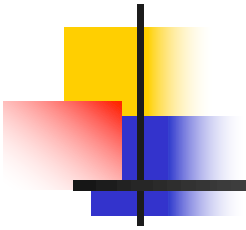


Aprendizaje de Máquina

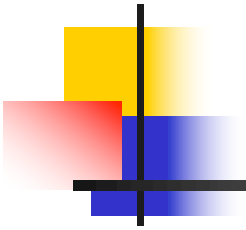
ITAM

agosto-diciembre 2017



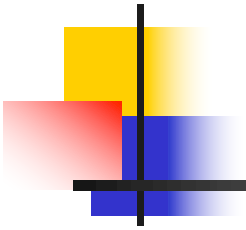
Menú

- Dinámica de Clase
- Evaluación
- Temario
- Minado de Datos
- Un Método Simple



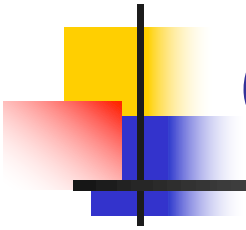
Dinámica de Clase

- Tenemos una clase a la semana de tres horas
- Clases
  - 1.5 horas dedicadas a la presentación del profesor
  - 0.5 hora dedicadas a la discusión de artículos (participación de todos) y proyectos
  - 1 hora para la resolución de ejercicios



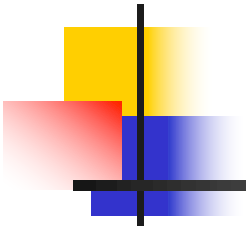
Evaluación

- Dos exámenes de igual valor (20% c/u)
- Ejercicios de clase (30%)
  - Se muestran durante clase al profesor y se considera una entrega a tiempo hasta la siguiente clase. A partir de eso se pierde un punto por semana
- Proyectos y presentaciones (30%)
  - Los detalles se encontrarán en GIT



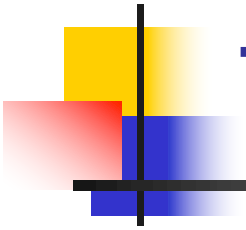
Objetivo

- Obtener un panorama del aprendizaje de máquina
- Entender el funcionamiento de algunas de las técnicas de esta disciplina
- Obtener experiencia en su uso



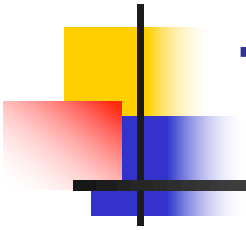
Motivación

- Cada vez más datos digitalizados
  - Medicina, economía, sensores, politica, publicidad…
- Cada vez máquinas más poderosas
- Necesitamos algoritmos para aprovechar las nuevas oportunidades
- Emular y automatizar algunos de los procesos cognitivos del ser humano
  - Las computadoras como extención de la inteligencia
  - Video https://www.youtube.com/watch?v=7Pq-S557XQU



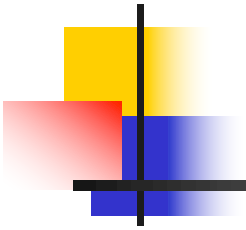
Temario

- Técnicas de aprendizaje supervisado
  - Decisiones Bayesianas
  - Árboles de decisión
  - Métodos clásicos:
    - Regresión lineal
    - Regla del perceptrón
  - Redes Neuronales
  - K-vecinos cercanos
  - Máquinas de Soporte Vectorial (SVM)
  - Ensemble learning



Temario

- Técnicas de aprendizaje No-supervisado
  - Reglas de Asociación
  - K-medias
  - Clustering Jerárquico y otros
- Medidas de Error
- Teoría del Aprendizaje
- Reducción de Dimensionalidad
- Privacidad
- Visitas de profesores
  - Aprendizaje reforzado

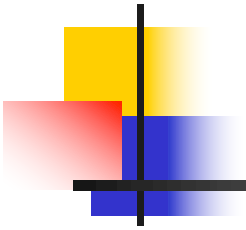


Bibliografía

- *The Elements of Statistical Learning*
  - Hastie, Tibshirani y Friedman
- *Pattern Recognition*
  - Duda, Hart y Stork
- Bishop, C. M., *Pattern Recognition and Machine Learning*
- Marsland, S., *Machine Learning: An Algorithmic Perspective*
- Artículos varios y material en internet

Herramientas

- Python
- R
- Excel
- Datos
  - http://archive.ics.uci.edu/ml/datasets.html
  - http://www.quandl.com/
  - http://catalog.data.gov/dataset
  - http://datos.gob.mx/
  - https://www.kaggle.com/



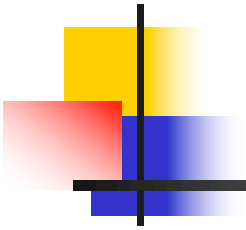
Enfoque

- La mayor parte del curso se avocará a estudiar una variedad de técnicas de aprendizaje
- Nuestra aproximación a estas técnicas será desde el punto de vista del minado de datos
  - También sirven, por ejemplo, para programar robots autónomos

Etapas para el Minado de Datos (1)

- Limpieza
  - Remover datos ruidosos, inconsistentes, etc. Adjudicar valores.
- Integración
  - Combinar las diferentes fuentes de datos
- Selección
  - Seleccionar el subconjunto de los datos relevante para el estudio. Si hay suficientes datos, guardar un subconjunto de estos para probar el modelo resultante
- Transformación
  - Seleccionar atributos, generar atributos agregados, convertir tipos de variables, etc

Etapas para el Minado de Datos (2)



- Minado
  - Utilizar técnicas de clasificación y regresión (una tarea de minado involucra, por lo general, varias técnicas)
- Evaluación
  - Identificar los resultados (patrones) interesantes (¿Qué es interesante?)
- Presentación
  - Usar técnicas de visualización para presentar los resultados obtenidos
  - Generar un sistema o protocolo para repetir el proceso con nuevos datos (de ser necesario)
- Nota el proceso no es necesariamente lineal pues en ocasiones es necesario regresar a etapas anteriores

Etapas para el Minado de Datos: El Caso Netflix

- Descripción de los datos
  - Los datos vienen en 1770 archivos de texto, uno por película
  - Cada archivo tiene en la primer línea el id de la película seguido de dos puntos “:”
  - El resto de la líneas tienen el siguiente formato:
    - Id_cliente, calificación, fecha

Etapas para el Minado de Datos: El Caso Netflix

- Etapa de preproceso
  - Limpieza de datos
    - Los datos en el caso de Netflix están limpios. En otros ejercicios puede ser necesario remover ejemplos con errores, componer errores, llenar campos vacíos, homogeneizar valores (todo en mayúsculas,…), etc.
  - Integración
    - Son 1770 archivos, cómo los cargo todos? ¿Necesito todos?
    - Talvez generar un solo archivo que en cada línea tenga un cliente y, separado por comas, la calificación que da a una película
    - Buscar información adicional en IMDB y integrar
  - Selección
    - Sólo algunos clientes
    - Sólo algunas películas
    - Apartar un subconjuntos de datos para hacer pruebas
    - Remover fechas…
  - Transformación
    - Cambio de escala a calificaciones
    - Generación de variables derivadas

Etapas para el Minado de Datos: El Caso Netflix

- **Etapa de minado**
  - Seleccionar técnicas de aprendizaje eg. C.5 y K-medias
  - Regresar, posiblemente, a la etapa de preproceso para alistar los datos
  - Minar, calcular errores, seleccionar otras técnicas,…

Etapas para el Minado de Datos: El Caso Netflix

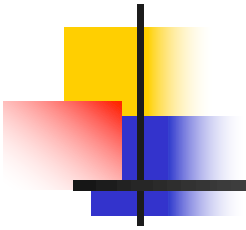
- Etapa de post-proceso
  - Evaluar resultados. En este caso qué técnicas funcionaron, y que tan bien
    - En otros ejercicios de minado: que patrones interesantes se encontraron, por ejemplo
  - Presentación
    - Powerpoint + documento
    - Medidas de error para cada técnica
    - Tener el sistema listo para predecir
      - Netflix proporciona un archivo con clientes y películas para los cuales hay que generar una predicción.

Etapa de Minado
Aprendizaje

- ¿Qué es aprender?
- ¿Para qué hacer que una computadora aprenda?
  - Buscamos entender, encontrar relaciones, similitudes, diferencias, invariantes. Buscamos predecir
  - Buscamos modelar un fenómeno sin tener el conocimiento explícito del proceso subyacente
- Al aprendizaje de máquina consiste de varias técnicas (familias de funciones) para aproximar el proceso que genera lo observado

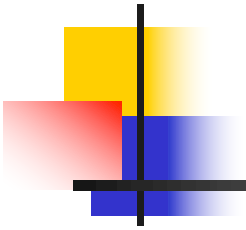
Etapa de Minado
Aprendizaje

- Una definición (Mitchell, Machine Learning)
  - Se dice que un programa de computadora aprende de su experiencia E con respecto a una tarea T y función de evaluación F, si su desempeño en la tarea T (con respecto a la evaluación F) mejora con la experiencia E
- En general un problema de aprendizaje debe precisar:
  - La clase de tareas a las que se refiere
  - La función de evaluación
  - La fuente de experiencia
- Ya definido esto podemos seleccionar un modelo (a.k.a función objetivo) y ajustarlo para maximizar su desempeño



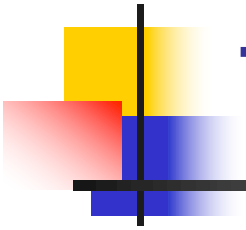
Ejemplos

- Encontrar la tendencia de una acción
  - Tarea: Predecir el precio futuro de una acción
  - F.e: Ganancias
  - Experiencia: Movimientos históricos de un año
- Predecir si un día es bueno para jugar tenis
  - Tarea: Clasificar días como buenos o malos para jugar tenis
  - F.e: Porcentaje de días bien clasificados y porcentaje de días mal clasificados
  - Experiencia: Historia de un mes



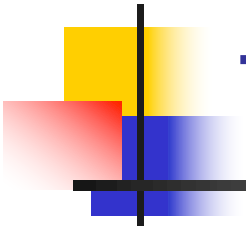
Ejemplos

- Segmentar un grupo de clientes de supermercado
  - Tarea: Crear *n* categorías de clientes. Clasificar a cada cliente como miembro de una categoría
  - F.e: Que los grupos tengan sentido estratégico. Disminuir los costos de publicidad e incrementar ventas. (Evaluación indirecta)
  - Experiencia: Transacciones en el supermercado



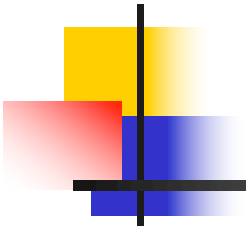
Tipos de Problemas

- Lo que queremos aprender es una función que dado un ejemplo (un dato) nos entregue un valor
  - Si el valor es numérico se conoce como **regresión**
    - El valor de una acción
  - Si el valor es categórico se conoce como **clasificación**
    - Si un día es bueno o no para jugar tenis



Tipos de Técnicas

- Dependido de si tenemos disponible el valor de la función objetivo para los ejemplos de entrenamiento, las tareas de aprendizaje se dividen en:
  - Aprendizaje Supervisado
    - Se utilizan los datos de entrenamiento y el valor correcto para cada uno de ellos de la función objetivo (la función que intentamos aprender)
      - Árboles de decisión, redes neuronales
  - Aprendizaje No-supervisado
    - Sólo se le presentan datos, se desconoce el valor objetivo de los ejemplos de entrenamiento
      - Técnicas de agrupamiento (“clustering”)
      - K-medias, EM, redes neuronales



Lo que hay que hacer

- Ya definido el problema debemos:
  - Determinar los ejemplos con los que vamos a entrenar (fuente de experiencia) y ponerlos a modo
  - Escoger los atributos que utilizaremos de cada ejemplo
  - Normalizar, escalar
  - Escoger el algoritmo de aprendizaje. Qué tan expresiva queremos que sea nuestra representación?
    - Si es muy expresiva necesitaremos muchos ejemplos para poder aprender (y distinguir entre las distintas hipótesis)
    - Si es poco expresiva habrá conceptos que no se pueden representar
  - Evaluación de desempeño
  - Presentar resultados

Detalle Importante Acerca de la Fuente de Experiencia

- En que grado representan los ejemplos utilizados la realidad
  - Puede sesgar mucho el resultado
  - Lo ideal es que la los ejemplos que se usan para entrenar (aprender) sigan la misma distribución que los ejemplos que se encontrará en el mundo. Pero en ocasiones es necesario sesgar
  - e.g. Difícilmente podremos crear una buena segmentación de clientes de supermercado, si entrenamos sólo con datos de un día de la semana

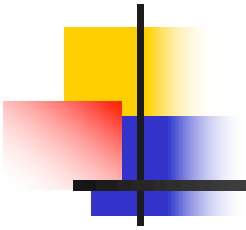
Procedimiento (ideal) para generar un modelo

1. Dividir los datos en un conjunto de entrenamiento y uno de prueba
   - El conjunto de prueba no hay que usarlo para nada!! Ni siquiera para normalizar. Nada, solo para probar al final
   - La regla de dedo es 75% para entrenamiento y 25% para pruebas
2. El conjunto de entrenamiento puede a su vez contener un conjunto de validación
   - El conjunto de entrenamiento (sin el de validación) se usa para ajustar el modelo.
   - El conjunto de validación sirve para comparar diferentes parámetros del modelo (para los que los tienen). Por ejemplo queremos decidir entre si ponerle 4 u 8 neuronas a la capa intermedia de una RN
   - La extracción del conjunto de validación se hace por cada experimento (validación cruzada)
3. Ya elegido el modelo y sus parámetros. Entrenar con todo el conjunto de entrenamiento y reportar desempeño con el conjunto de prueba

Nota: Si no hay suficientes datos para tener un conjunto de prueba intacto entonces se utiliza sólo el paso 2 para seleccionar parámetros. El modelo se reentrena finalmente con todos los datos.

Comentario

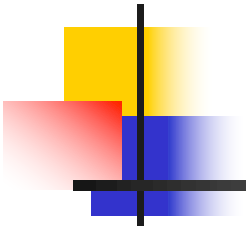
- La mayoría de las fallas en la aplicación del aprendizaje de máquina son aquí, en el procedimiento. Es muuuy fácil equivocarse y entrenar, de alguna forma, con información de los datos de prueba



Un Método Simple

- Uso de conceptos probabilísticos simples
  - Estimar frecuencias
  - Calcular probabilidades
  - Usar regla de Bayes
- Efectivo en una gran cantidad de casos

Un Método Simple



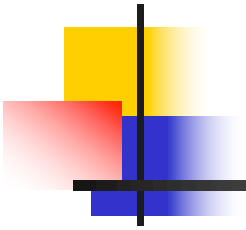
- Tipo de Método
  - Supervisado
- Supuestos
  - Ejemplos de entrenamiento son representativos

Un Método Simple

- Los datos provienen de un proceso que no es totalmente conocido
- Nosotros lo modelamos como un proceso estocástico
  - Por ejemplo: los datos son resultados de volados, los datos son precios de una acción
- Gracias a que no tenemos toda la información necesaria para descubrir el proceso determinista que lo rige, definimos una variable aleatoria X que pude tomar distintos valores
  - Águila o Sol en el caso de los volados
  - Un número real positivo para las acciones
- Deseamos encontrar la probabilidad P(X=valor)
- En el caso de los volados supongamos que P(X=aguila) $=p_o$
  - ¿Qué en P(X=sol)?

Un Método Simple

- Si deseamos predecir qué valor tendrá el siguiente volado
  - Escogemos águila si $p_o > 0.5$
  - Sol de otro modo
  - ¿Porqué?
- Problema: cómo calculamos p_o?
- Lo estimamos de una muestra de tamaño N (ejemplos de aprendizaje)
  - $\hat{p}_o$=número de aguilas/N



Un Método Simple

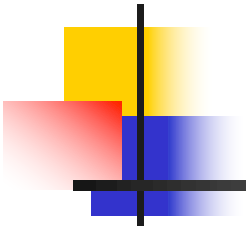
- Supongamos ahora que lo que tenemos es una base de datos de clientes de un banco y deseamos un modelo para determinar si un cliente nuevo es de alto o bajo riesgo para un préstamo
- Supongamos que las variables disponibles son su ingreso mensual y su saldo actual
  - Puede haber muchas más variables. El uso de más variables no es necesariamente mejor (esto lo veremos a detalle más adelante)

Un Método Simple

- Lo que queremos calcular es
  - $P(C|X_1,X_2)$
  - Donde C puede ser alto o bajo riesgo y X_1 es el valor del ingreso mensual y X_2 el saldo actual
- De esta manera podemos definir que un cliente es de alto riesgo si
  - P(C=alto|ingreso_cliente, saldo_cliente)>=0.2

Un Método Simple

- A diferencia del ejemplo de la moneda
  - Si el cliente es de alto o bajo riesgo depende de otras muchas variables observables (dos)
  - Primero observamos las características del cliente y luego decidimos su nivel de riesgo
- En este caso es la probabilidad de que el cliente sea de alto riesgo dado que ya observamos su ingreso y saldo
  - $P(C|X_1,X_2)$ es la probabilidad posterior



Un Ejercicio

- En un concurso existen tres puertas. Detrás de una de ellas hay un premio

1. El conductor te pide que elijas una.
2. Una vez que elijes una puerta el conductor abre una de las otras dos revelando que el premio no esta ahí.
3. Ahora te da la opción de cambiar la puerta que elegiste
4. ¿Qué debes hacer? ¿Importa si cambias? Demuéstralo! Simúlalo en Python
5. Nota: lo que importa es cuál debe ser tu acción si jugaras este juego repetidas veces

Regla de Bayes

- Sea **x** el vector de observaciones (X_1 y X_2 en nuestro ejemplo)
  - P(C|**x**)=P(C)P(**x**|C)/P(**x**)
  - Probabilidad previa (“prior”)
    - P(C=alto), P(C=bajo)
    - Es la probabilidad de que un cliente sea de riesgo alto o bajo independientemente de **x.** P(C=alto)+P(C=bajo)=1
  - Probabilidad de clase
    - P(**x**|C)
    - La probabilidad condicional de que un ejemplo perteneciente a C tenga **x** asociado
    - P(I,S|C=alto) es la probabilidad de que un cliente riesgoso tenga como ingreso I y saldo S

Regla de Bayes

- Sea **x** el vector de observaciones (X_1 y X_2 en nuestro ejemplo)
  - $P(C|\mathbf{x})=P(C)P(\mathbf{x}|C)/P(\mathbf{x})$
  - Evidencia
    - $P(\mathbf{x})$
    - Es la probabilidad de observar los datos **x**, independientemente del riesgo del cliente (normaliza)
    - $P(\mathbf{x})=P(\mathbf{x}|C=\text{alto})p(C=\text{alto})+P(\mathbf{x}|C=\text{bajo})p(C=\text{bajo})$
- Las suma de las probabilidades posteriores es 1
  - $P(C=\text{alto}|\mathbf{x}) + P(C=\text{bajo}|\mathbf{x})=1$

Ejemplo del uso de Bayes para Predicción

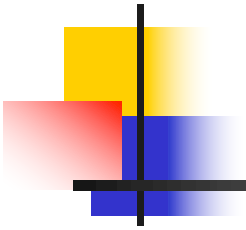
- Dados

Ingreso	Saldo	Clasificación
20	10	alto
30	40	bajo
30	10	alto
10	30	bajo
20	10	alto
20	10	bajo
20	5	alto

Ejemplo
Regla de Bayes

- Calcular la probabilidad posterior de cada clase para los siguientes datos

Ingreso	Saldo
20	10
30	40
20	5



Dificultades con el Método

- El número de ejemplos necesarios crece muy rápido con el número de atributos
- No esta definido para atributos que no se encuentran en el conjunto de entrenamiento.
- La estimación de probabilidad que se mencionó no sirve para variables contínuas

Bayes Ingenuo (Naive) (mitiga el primer punto)

- Supuesto: Las probabilidades de clase son independientes, i.e.,
  - P(X,Y,Z|C)=P(X|C) * P(Y|C) * P(Z|C)
  - ¿Qué ventajas y desventajas tiene esto?
- Calcular la probabilidad posterior de cada clase para los siguientes datos

Ingreso	Saldo
30	5
20	10

Datos nuevos (mitiga el segundo punto)

- Laplace smoothing
  - Sumar un término en el numerador y en el denominador
  - P’(C)=(|C| +1) / (numdatos+ numCat)
    - Donde C es una categoria y |C| es el número de datos con categoría C
  - P’(x $\wedge$ C)=(1+|x $\wedge$ C|) / (|C| + unique(C))
    - Donde |x $\wedge$ C| es el número de veces que x tiene la categoría C y unique(C) es el número de datos diferentes con categoría C

Bayes Ingenuo Continuo Variable continuas (mitiga el tercer punto)

- Para este método debemos suponer que los datos se distribuyen de acuerdo a alguna distribución y en base a esto calcular la las probabilidades de clase.
- Lo más común es suponer normalidad

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}},$$

- Calcule la probabilidad posterior de cada clase para los siguientes datos (entrene con los datos antes dados)

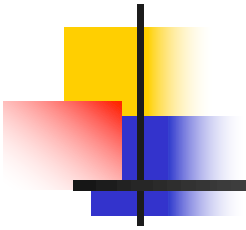
Ingreso	Saldo
9.5	35
300	5
25	15.7

Bayes Ingenuo Continuo

- Nota:
  - La función que dimos es la función de densidad.
    - No da una probabilidad (la probabilidad de una variable continua es cero por definición)
    - Lo que nos da es la “altura” de la función en un punto, para calcular la probabilidad necesitaríamos definir una base (para aproximar usando base x altura)
    - Si estandarizamos los datos comparar las alturas nos da el resultado deseado
  - Si al final el valor de utilizar la fórmula es igual a cero, se recomienda en su lugar poner un valor muy pequeño como 10^{-6}

Una simplificación para predictor binario

- Predecir clase C si y sólo si $P(C|\mathbf{X})/P(\sim C|\mathbf{X}) \geq 1$
- $\log(P(C|\mathbf{X})/P(\sim C|\mathbf{X})) \geq 0$
- $\log(P(C)\Pi_{x_i \varepsilon \mathbf{X}} P(x_i|C)) \geq \log(P(\sim C)\Pi_{x_i \varepsilon \mathbf{X}} P(x_i|\sim C))$
- $\log(P(C))+\Sigma_{x_i \varepsilon \mathbf{X}} \log(P(x_i|C)) \geq \log(P(\sim C))+ \Sigma_{x_i \varepsilon \mathbf{X}} \log(P(x_i|\sim C))$
- Esto es evita calcular el término de la evidencia $P(\mathbf{X})$
  - Se elimina en la división y
- Evita los problemas numéricos de multiplicar números muy pequeños
  - El logaritmo de la multiplicación se convierte en la suma de logaritmos



Ejemplo

- Escriba un filtro de spam en python usando la técnica de Bayes ingenuo
- Utilice los datos de spam de
- https://archive.ics.uci.edu/ml/datasets/Spambase

Tarea 1

- Leer artículos
  - “Imitation Game” de Alan Turing
  - Funes el memorioso de José Luis Borges