Concept Drift

Eduardo Morales, Hugo Jair Escalante

INAOE

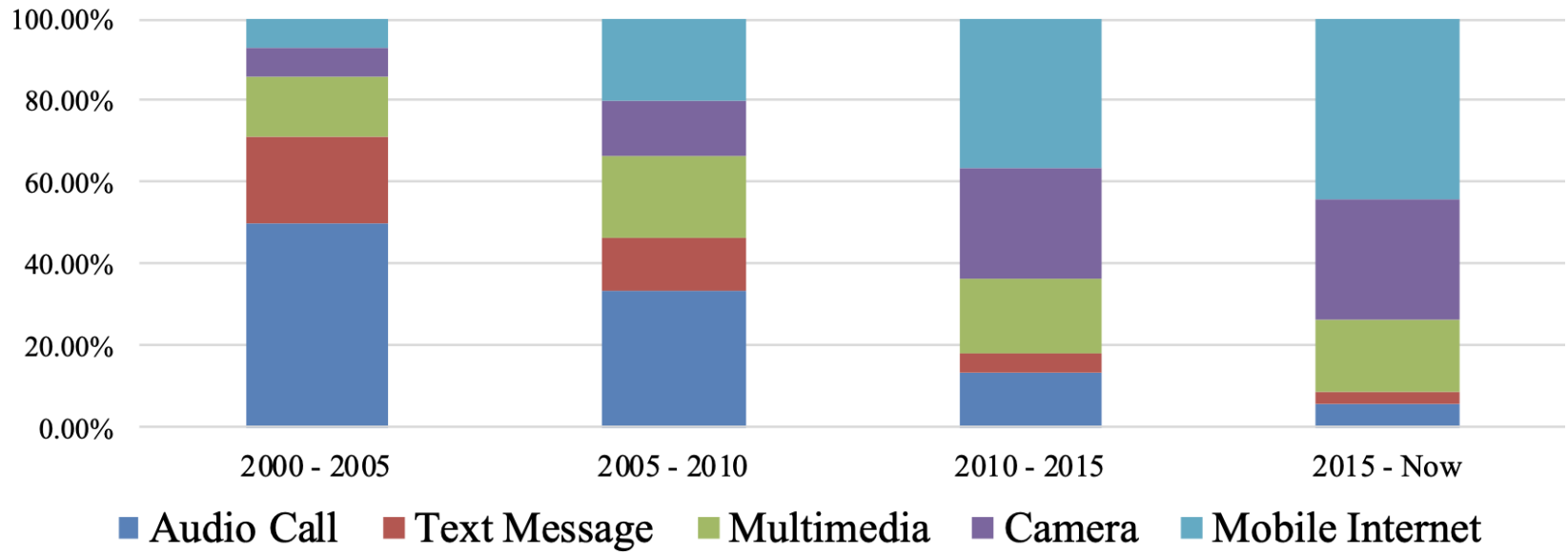
Concept Drift

- *Concept Drift* (CD) describe cambios no anticipados en la distribución de los datos con el tiempo
- Busca desarrollar metodologías y técnicas para detectar los cambios, entenderlos y adaptar los modelos
- Un sistema de aprendizaje en una situación de cambio (*drift*) se degrada y pierde su efectividad

Concept Drift

- El fenómeno se presenta en ambientes cambiantes, que pueden involucrar nuevos productos, mercados, clientes, etc.
- Lo que sucede con CD es que las propiedades estadísticas de la variable objetivo (que es la que quiere predecir el modelo) cambia con el tiempo de manera no anticipada
- Si ocurre CD el modelo inducido con los datos anteriores deja de ser relevante con los nuevos datos, resultando en malas predicciones

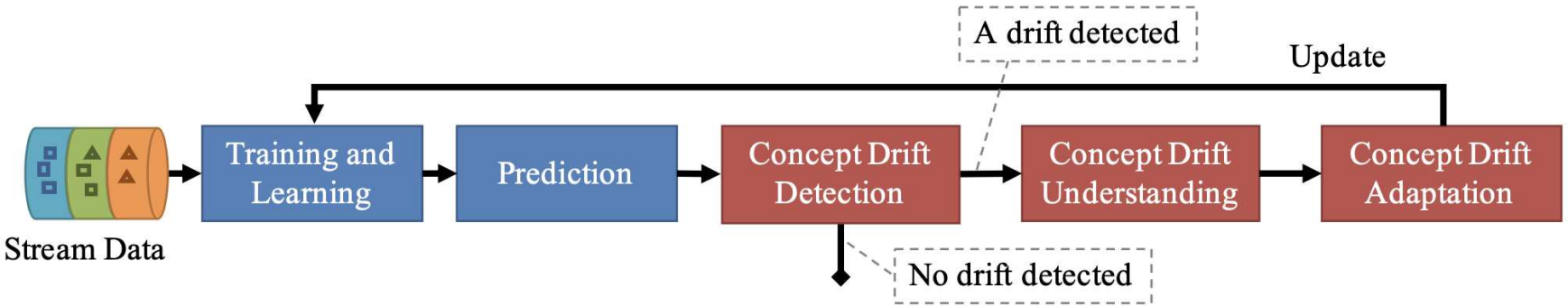
Ejemplo: Uso de celulares



Retos

- ¿Cómo detectar de forma precisa CD en bases de datos no estructuradas y con ruido?
- ¿Cómo entender cuantitativamente el CD de forma que se pueda explicar?
- ¿Cómo reaccionar de manera efectiva a CD adaptando el conocimiento relacionado?

Esquema de manejo de CD



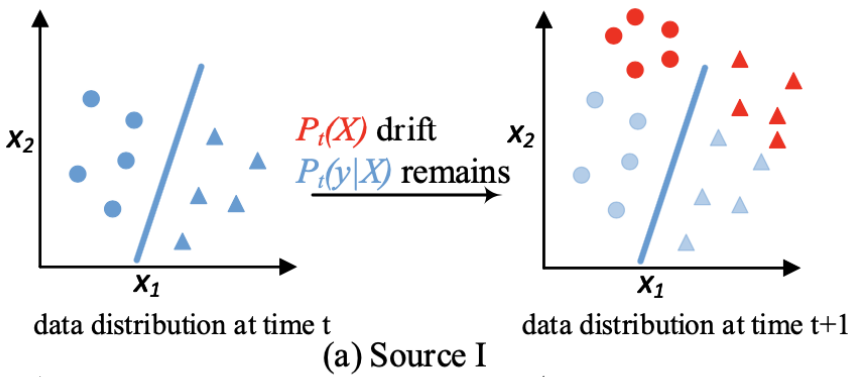
Concept Drift

Dado un periodo de tiempo $[0, t]$, un conjunto de ejemplos $S_{0,t} = \{d_0, \ldots, d_t\}$, donde $d_i = (X_i, y_i)$ es una observación, X_i es el vector de atributos, y_i es la etiqueta, y $S_{0,t}$ sigue una cierta distribución $F_{0,t}(X, y)$
CD ocurre en $t + 1$, si $F_{0,t}(X, y) \neq F_{t+1,\infty}(X, y)$, que se puede denotar como: $\exists t : P_t(X, y) \neq P_{t+1}(X, y)$

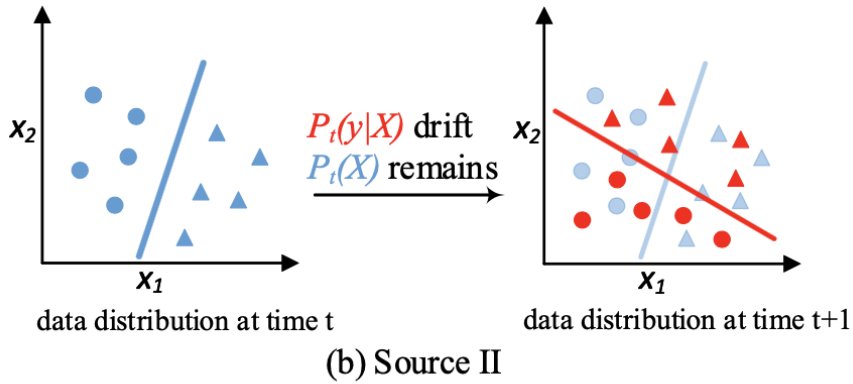
Diferentes fuentes de Concept Drift

1. $P_t(X) \neq P_{t+1}(X)$ mientras que $P_t(y|X) = P_{t+1}(y|X)$, como no se cambia la clasificación, algunos se refieren a este tipo como *drift* virtual
2. $P_t(y|X) \neq P_{t+1}(y|X)$ mientras que $P_t(X) = P_{t+1}(X)$
3. $P_t(X) \neq P_{t+1}(X)$ y $P_t(y|X) \neq P_{t+1}(y|X)$

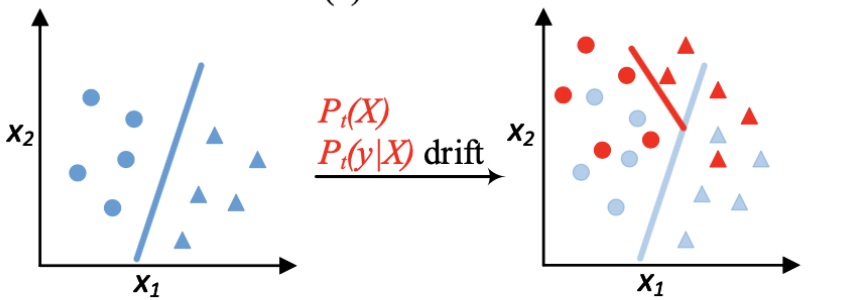
Fuentes de Concept Drift



(a) Source I



(b) Source II



(c) Source III

Two dimensional
data $X=\{x_1, x_2\}$
with two class label
$y=\{y_0, y_1\}$

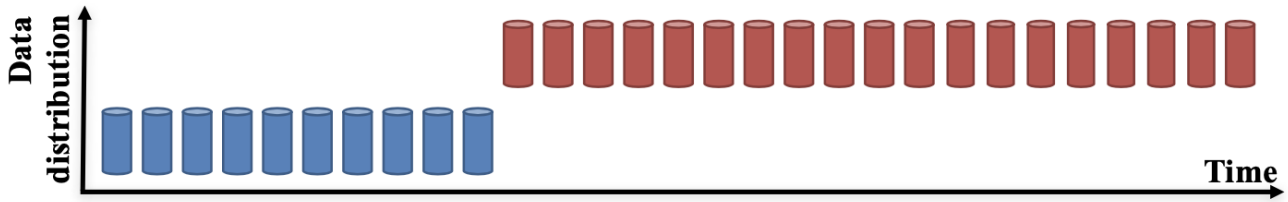
- label y_0 at time t
- label y_1 at time t
- label y_0 at time t
- label y_1 at time t

Tipos de Concept Drift

- Cambio abrupto
- Cambio gradual
- Cambio incremental
- Cambio recurrente

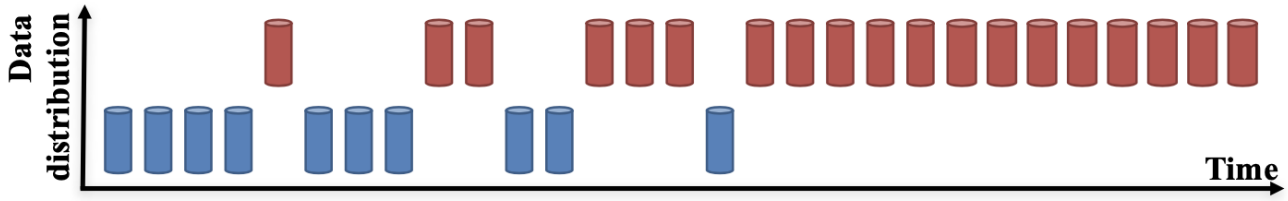
Tipos de Concept Drift

Sudden Drift:



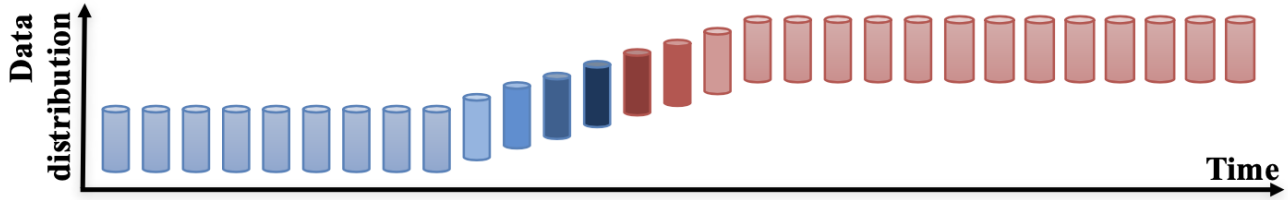
A new concept occurs within a short time.

Gradual Drift:



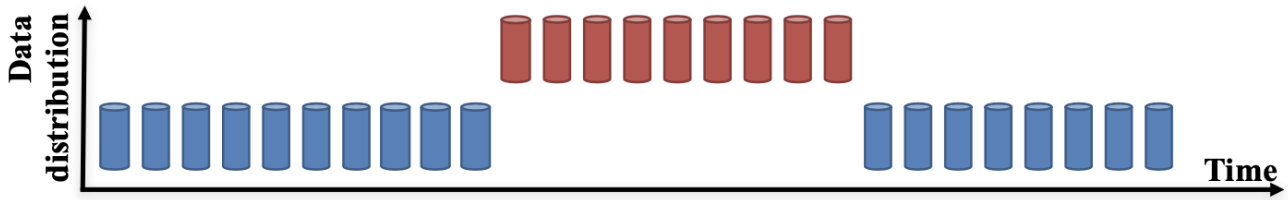
A new concept gradually replaces an old one over a period of time.

Incremental Drift:



An old concept incrementally changes to a new concept over a period of time.

Reoccurring Concepts:



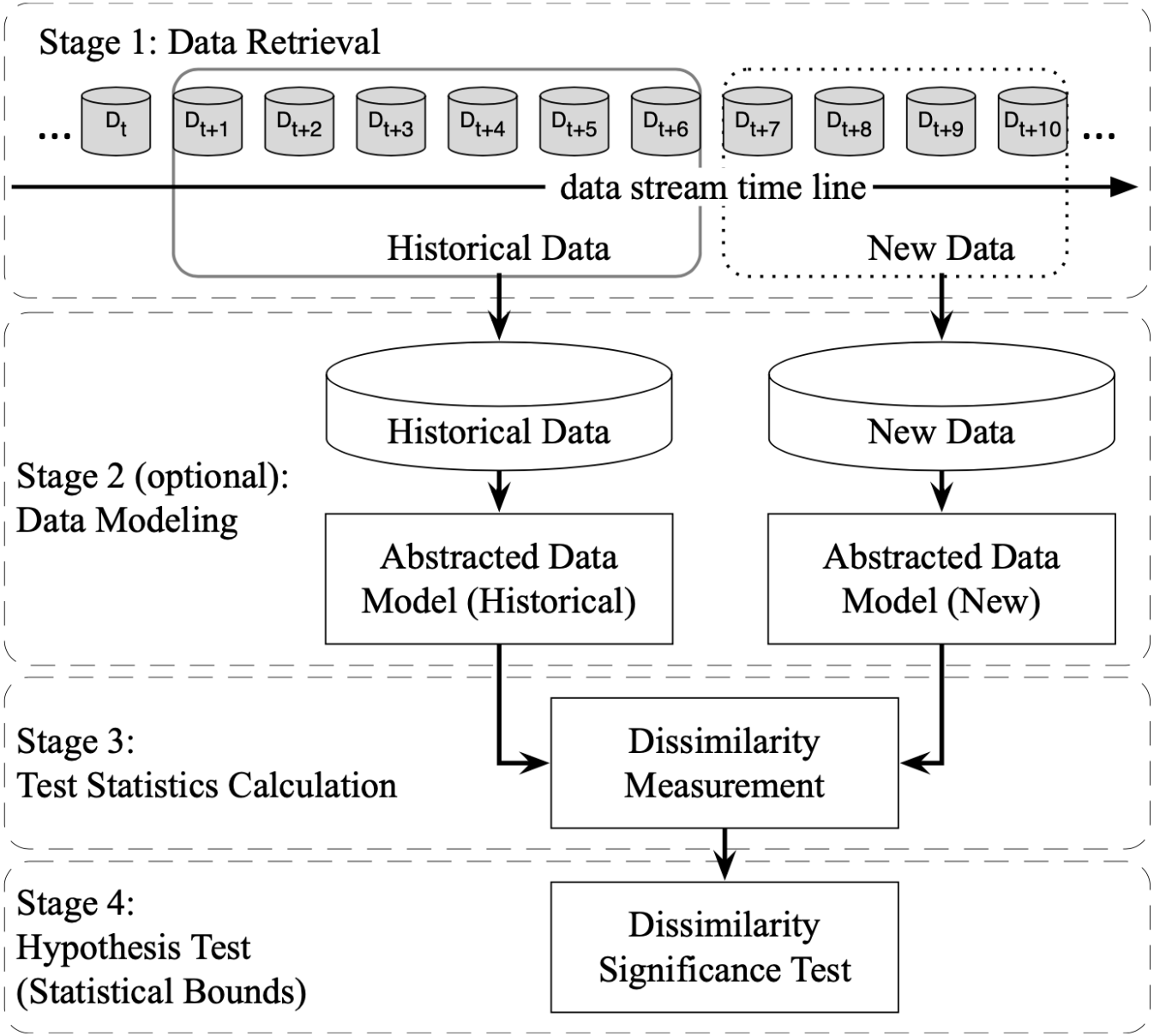
An old concept may reoccur after some time.

Detección de Concept Drift

En general, inviolucra cuatro pasos:

1. Colectar un subconjunto de datos con suficiente información para estimar una distribución
2. Preprocesar los datos (opcional) para seleccionar o transformar los datos originales
3. Medir las diferencias usando una medida precisa y robusta (sigue siendo tema de investigación)
4. Probar si la diferencia es significativa

Detección de Concept Drift



Algoritmos de detección de Concept Drift

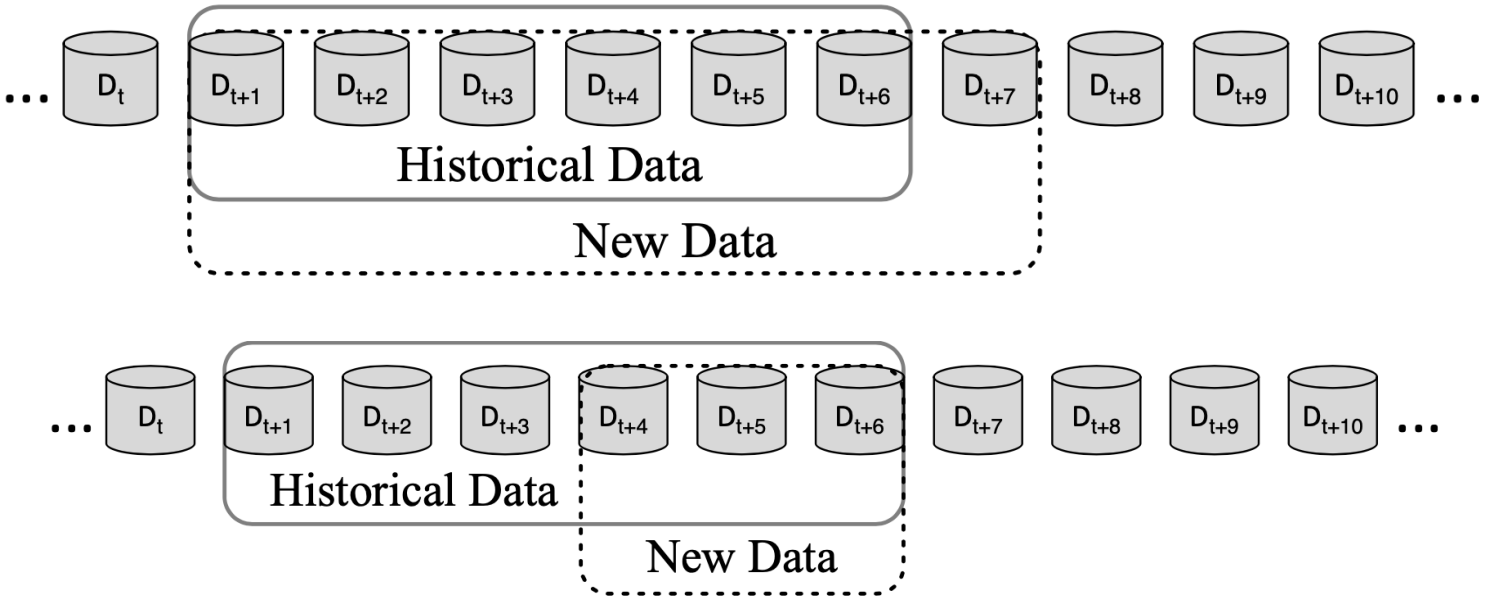
Se pueden dividir en tres categorías de algoritmos basados en:

- Error de predicción
- Distribución de los datos
- Múltiples hipótesis

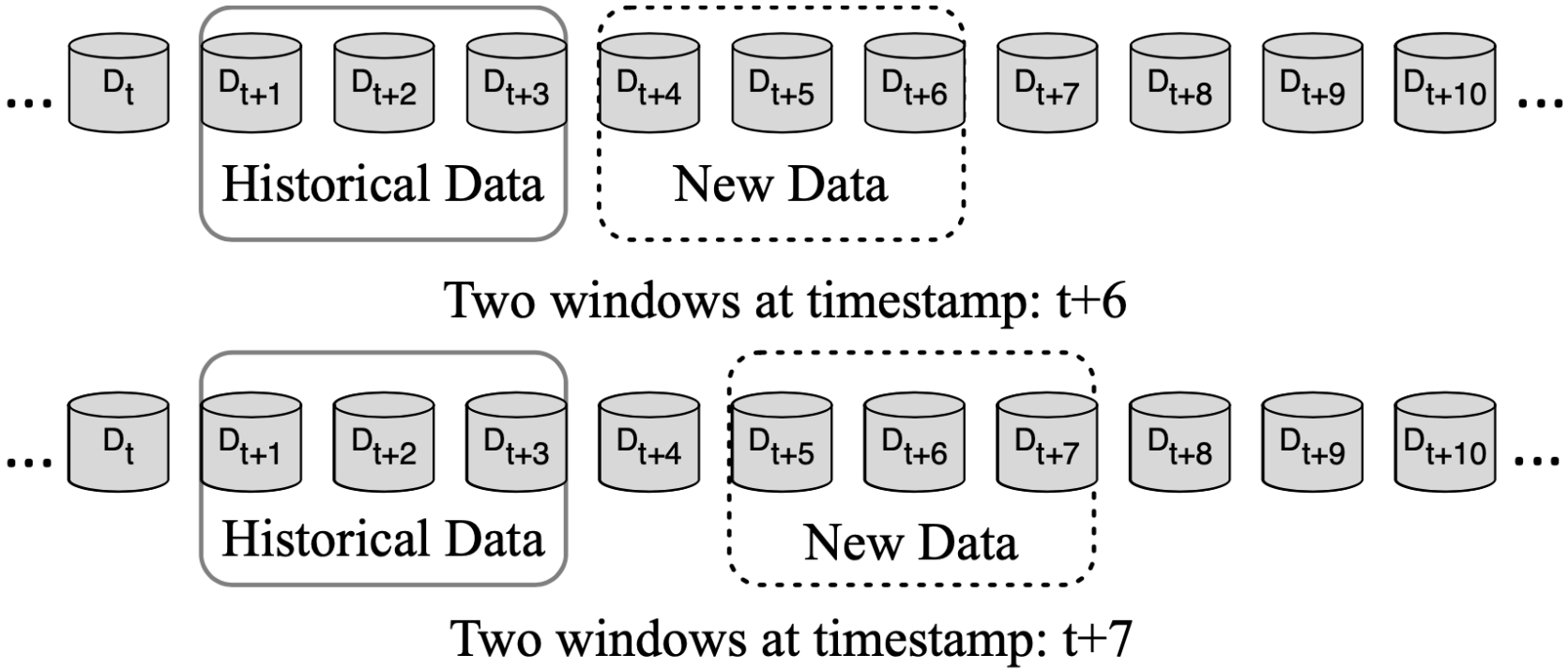
Algoritmos basados en detección de error

- Monitorean el error de los clasificadores hasta que encuentran una desviación significativa
- Si se empiezan a observar errores se empieza a contruir un nuevo modelo
- Si se rebasa el umbral permitido se reemplaza el modelo actual por el nuevo
- Algunos variantes consideran diferentes formas de considerar los datos históricos y los datos recientes

Esquemas de manejo de los datos para detectar Concept Drift



Esquemas de manejo de los datos para detectar Concept Drift

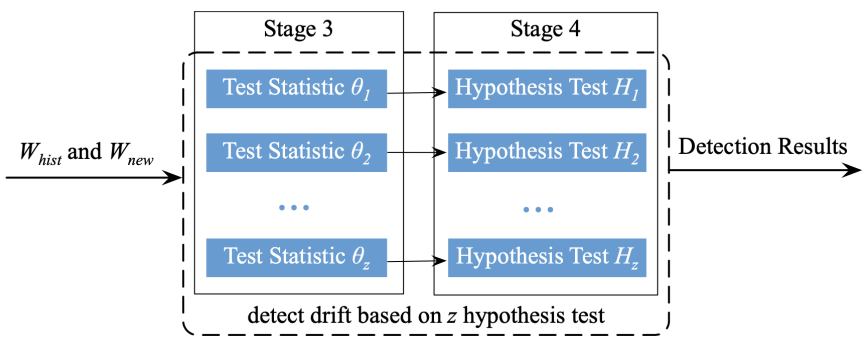


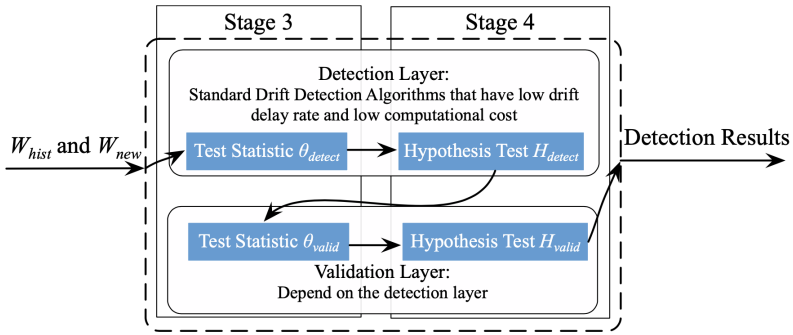
Algoritmos basados en distribuciones

- La idea es usar una distancia/métrica que mida la diferencia entre distribuciones de los datos históricosy los recientes
- De nuevo si se encuentra una diferencia significativa, se busca hacer algo
- Total Variation: $TV(P, Q) = 2sup_{x \in X}|P(x) - Q(x)|$
- Kullback-Liebler: $D_{KL}(P||Q) = \sum_{x \in X} P(x) \log(\frac{P(x)}{Q(x)})$

Algoritmos basados en ensambles

- La idea es usar múltiples pruebas para detectar Concept Drift
- Algunos usan varias pruebas en paralelo y otros de forma jerárquica





Entendimiento de Concept Drift

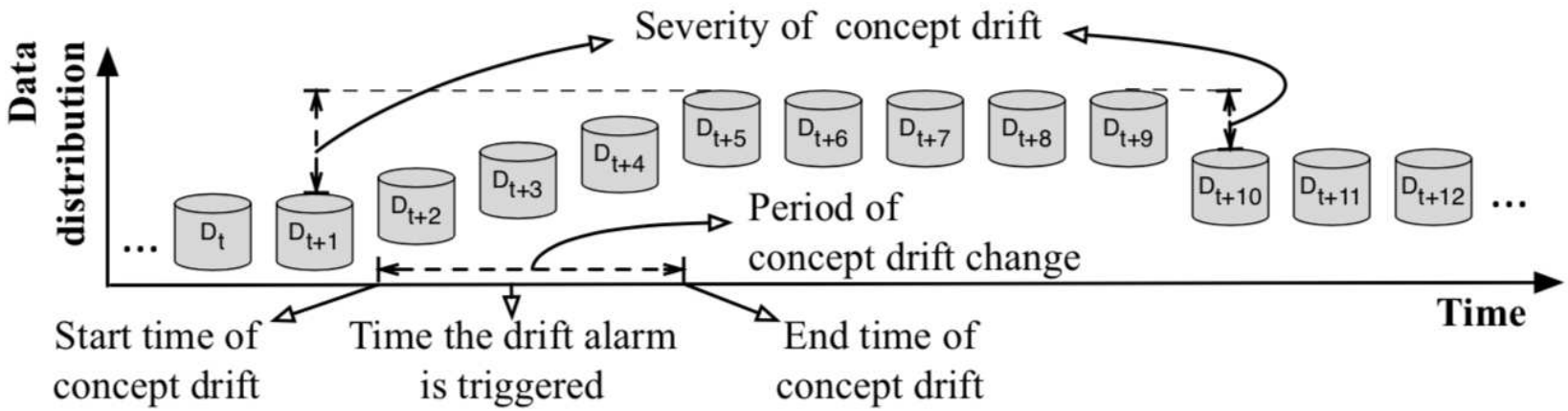
Involucra al menos tres aspectos, entender:

- Cuándo ocurre: Entender cuándo se da y por cuánto tiempo
- Cómo ocurre: Evaluar la severidad del cambio
- Dónde ocurre: Localizar las regiones donde ocurre

Cúando ocurre

- Normalmente se usa una alarma que también indica que hay que adaptarse
- Generalmente la alarma está defasada, ya que se requiere cierta cantidad de datos para identificarla
- Algunos usan advertencias (alarmas “relajadas”) y se esperan a confirmarlas, pero guardan los datos desde la advertencia para actualizar los modelos
- Todos los algoritmos de *concept drift* tienen un sistema de detección

Cúando ocurre



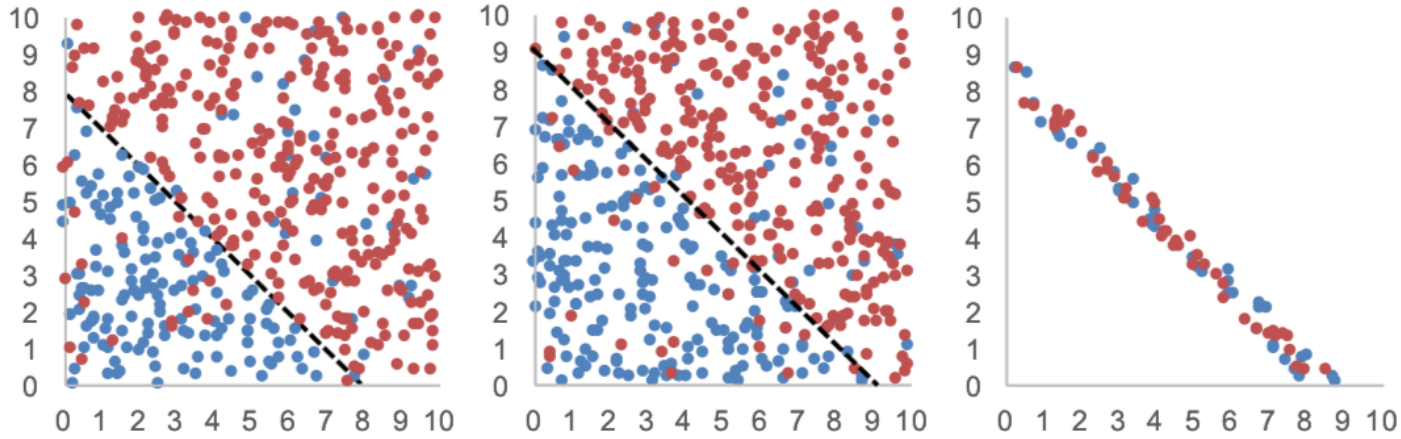
Cómo ocurre (la gravedad del cambio)

- La gravedad mide que tanto se parece el nuevo concepto con el anterior
- Se puede usar el error en clasificación de manera indirecta
- Se puede usar el cambio en las distribuciones de manera directa (variantes de Total Variance, Kullback-Leibler)
- La severidad nos puede indicar la estrategia a seguir: Leve $=>$ cambio incremental, Grave $=>$ reentrenar al modelo
- La mayoria de los algoritmos pueden dar un indicador de la gravedad del cambio

Dónde ocurre

- Encontrar regiones en el espacio de datos de los atributos que tienen diferencias importantes
- Aunque exista un cambio global en la distribución, pueden existir regiones “estables”
- En esas regiones el modelo anterior sigue siendo válido y las regiones que cambiaron se pueden usar para el nuevo modelo
- Muy pocos algoritmos detectan regiones

Dónde ocurre



(a) Data at time t (b) Data at time $t+1$ (c) Data in drift regions

Estrategias de adaptación

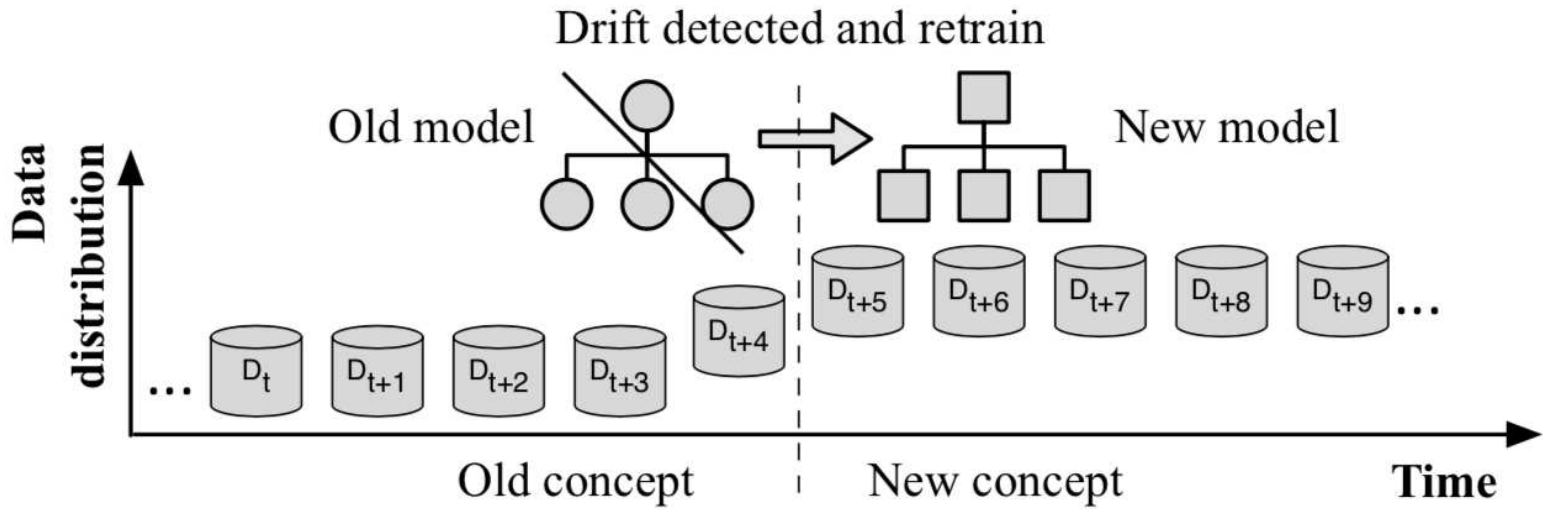
Se pueden agrupar en:

1. Reentrenar
2. Unsando ensambles
3. Ajustar

Reentrenar

- Lo más fácil es volver a entrenar un nuevo modelo usando los últimos datos
- *Paired Learners* mantiene dos modelos: Si el modelo *estable* se equivoca en donde el modelo *reactivo* no, reemplazalos
- Existe un balance para decidir el tamaño de la ventana a utilizar

Reemplazo


Drift detected and retrain
Old model
New model
Data distribution
D_t
D_{t+1}
D_{t+2}
D_{t+3}
D_{t+4}
D_{t+5}
D_{t+6}
D_{t+7}
D_{t+8}
D_{t+9}
Old concept
New concept
Time

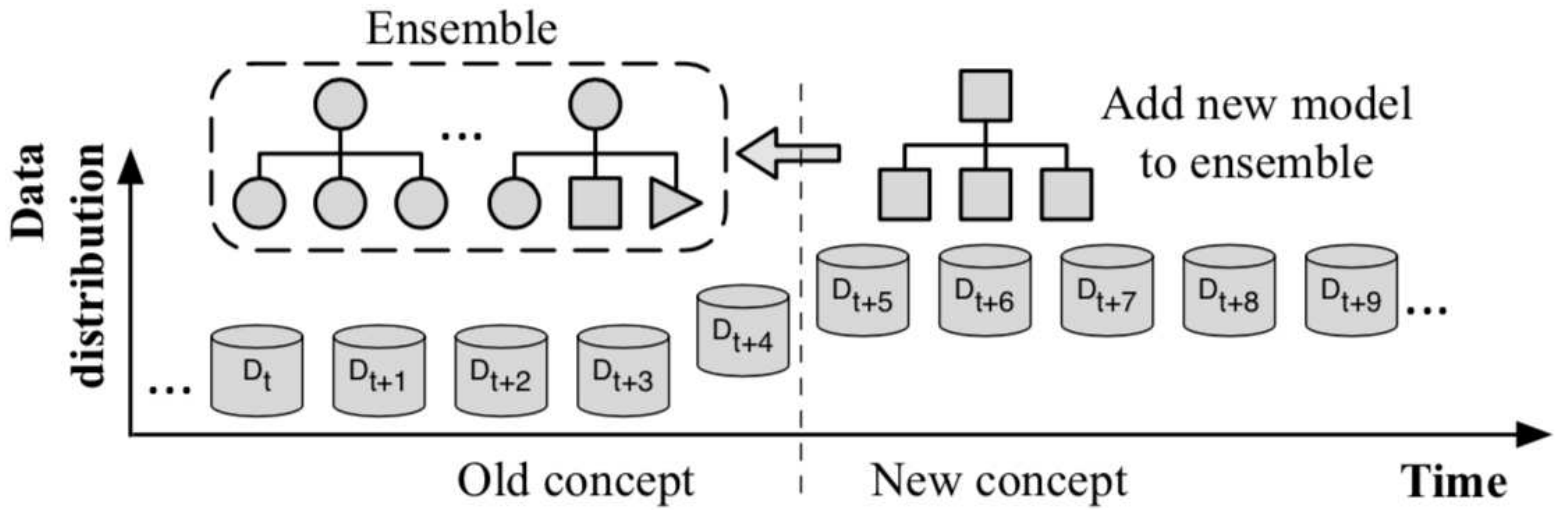
Reentrenar

- Algunos integran la detección del cambio en los algoritmos
- *DELM* ajusta el número de nodos ocultos de una ELM
- *Just-in-Time* elimina instancias viejas dentro de un k-NN
- ...

Ensambles

- En casos de cambios recurrentes, el mantener y reutilizar modelos viejos se vuelve importante
- Los ensambles guardan varios modelos y se han utilizado para lidear con *concept drift*, añadiendo nuevos modelos o cambiando el esquema de decisión
- *Leveraging Bagging* y *Adaptive Random Forest* entrenan un nuevo modelo y eliminan el peor modelo
- *Dynamic Weigthed Majority* entrena un nuevo modelo y reduce los pesos de los modelos que clasifican mal. Si el peso de un modelo se reduce más que un cierto umbral, se elimina

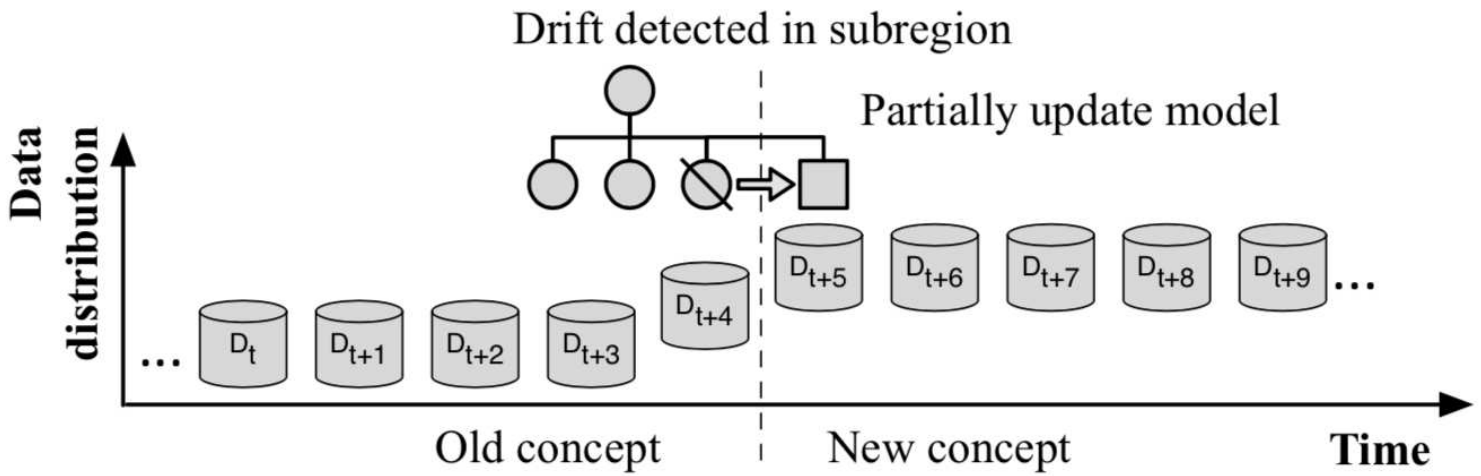
Ensambles


Ensemble
Add new model
to ensemble
Data
distribution
D_t
D_{t+1}
D_{t+2}
D_{t+3}
D_{t+4}
D_{t+5}
D_{t+6}
D_{t+7}
D_{t+8}
D_{t+9}
Old concept
New concept
Time

Ajustes

- Una alternativa es ajustar un modelo existente
- Muchos están basado en árboles de decisión
- Se basan en ventanas deslizantes de datos con la que se entrenan sub-árboles alternos monitoreando su desempeño

Ajustes


Drift detected in subregion
Partially update model
Data distribution
D_t
D_{t+1}
D_{t+2}
D_{t+3}
D_{t+4}
D_{t+5}
D_{t+6}
D_{t+7}
D_{t+8}
D_{t+9}
Old concept
New concept
Time

Evaluación

- Además de las medidas de evaluación tradicionales, se usan las de *data streams* (*prequential error*, cada instancia se prueba y se usa para construir un nuevo modelo)
- También se evalua qué tan bien se detectó el cambio (*true/false/miss detection rate*) y el retraso en la detección
- Las pruebas estadísticas para ver si un algoritmo es mejor que otro se basan en: (i) NcNemar test, (ii) Sign test, y (iii) Wilcoxon sign-rank test

Otros desarrollos

Concept Drift también se ha desarrollado en:

- Clases desbalanceadas
- Grandes bases de datos
- Active learning
- Aprendizaje semi-supervisado
- Reglas de clasificación