

Fluxo de dados

1001513 – Aprendizado de Máquina 2
Turma A – 2023/2
Prof. Murilo Naldi



naldi@ufscar.br



Agradecimentos

- Pessoas que colaboraram com a produção deste material: Elaine Ribeiro, Diego Silva
- Gama, J. A survey on learning from data streams: current and future trends. Prog Artif Intell 1, 45–55 (2012). <https://doi.org/10.1007/s13748-011-0002-6>
- Data stream clustering: A survey. JA Silva, ER Faria, RC Barros, ER Hruschka, AC Carvalho, J Gama ACM Computing Surveys (CSUR) 46 (1), 1-31

Fluxo Contínuo de Dados (FCD)

Dados “chegam” a todo **tempo**

- Em vez de conjuntos de dados, processamos em **fluxo**
 - Possível grande volume e alta velocidade
- Batch vs. fluxo



Fluxo Contínuo de Dados (FCD)

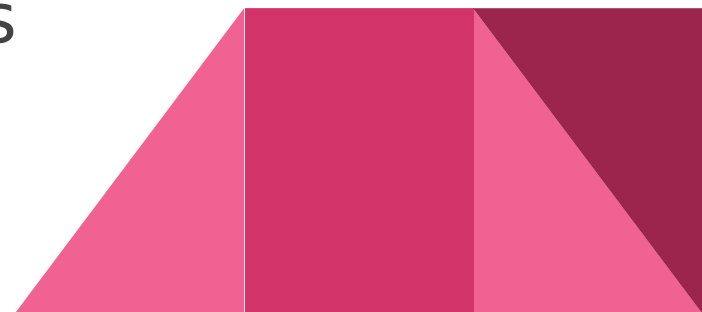
Formalmente temos:

- Um fluxo de dados S é uma sequência massiva de objetos $S = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n, \dots\}$, que é potencialmente ilimitado ($n \rightarrow \infty$), em uma sequência de *timestamps* $T = \{t_1, t_2, \dots, t_n, \dots\}$
 - Que chega continuamente
 - Ordem implícita e sem controle
 - Dado não pode ser armazenado indefinitivamente
 - Processado e descartado

Fluxo Contínuo de Dados (FCD)

Aplicações de fluxo de dados

- Todas :)
 - O que muda é a natureza dos dados
 - E, portanto, o cenário de aplicação
 - Classificação, agrupamento, regras de associação, redução de dimensionalidade, e outras...
 - Contudo é preciso entender as características do fluxo



Fluxo Contínuo de Dados (FCD)

Assumimos que:

- dados são gerados em distribuição **não estacionária**
 - Significa que a distribuição pode mudar com o fluxo!
 - Métodos tradicionais para *batch* podem não funcionar!



Fluxo Contínuo de Dados (FCD)

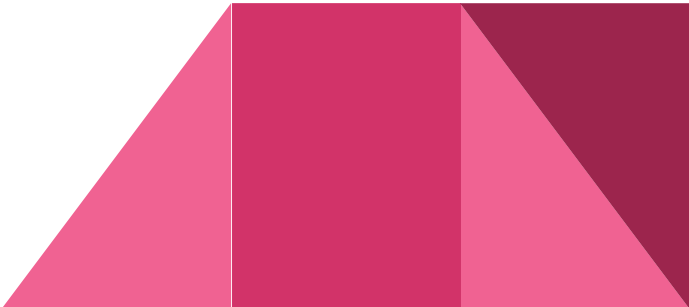
Assumimos que:

- dados são gerados em distribuição **não estacionária**
- precisamos manter um modelo preciso e coerência com o estado atual
 - Ou seja, o modelo precisa refletir um determinado momento do fluxo



Fluxo Contínuo de Dados (FCD)

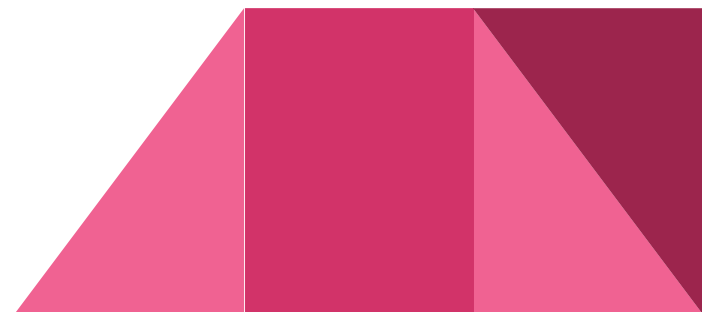
Assumimos que:

- dados são gerados em distribuição **não estacionária**
 - precisamos manter um modelo preciso e coerência com o estado atual
 - devemos manter as seguintes propriedades:
 - Incrementalidade
 - Aprendizado em tempo real (*online*)
 - Memória limitada
 - Acesso limitado ao “passado”
 - Capacidade de adaptação (a mudanças de conceito)
- 

Incrementalidade

Tarefas de aprendizado de máquina geram um modelo

- Modelo precisa ser incrementado com o fluxo
- Exemplos:
 - Classificação e Agrupamento
 - Novos objetos podem não se ajustar ao modelo
 - Novas classes/grupos (???)
 - Padronização Min/Max
 - Novos limites (???)



Aprendizado em tempo real (*online*)

Fluxo possui um tempo inerente entre um objeto e outro

- Não é possível armazenar objetos indefinitivamente
- “Algo” tem que ser feito nesse tempo restrito
 - Dado é analisado
 - Modelo atualizado
- Nem tudo precisa ser *online*
 - Exemplo: usar o modelo para tarefa de AM



Memória limitada

Recursos computacionais são limitados

- Não é possível armazenar objetos indefinitivamente (de forma incremental)
 - Dados precisam ser refletidos no modelo
 - Que tem que ser atualizado
 - Mas o modelo também não pode crescer indefinidamente
 - Algum mecanismo de descarte



Acesso limitado ao “passado”

Como os dados possuem ordem, a informação sobre o histórico dos dados pode ser a base de “padrões” para o aprendizado

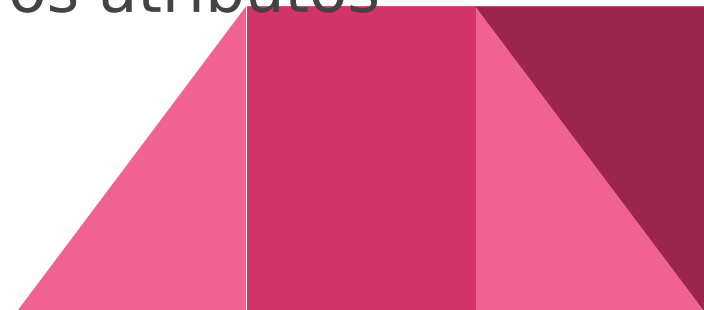
- Mas não é possível armazenar os dados indefinidamente
 - Apenas uma pequena parte, temporariamente
- Outra opção consiste em usar o modelo
 - Mas o modelo apenas reflete parte das características dos dados



Capacidade de adaptação

Natureza não estacionária da distribuição dos dados pode chegar a gerar um fenômeno chamado “mudança de conceito” (*Concept Drift*)

- Não se trata apenas de mudanças “pontuais”
 - Mas também de mudanças radicais no cenário (*Open World*)
 - Detecção de novos padrões
 - Inclusão e exclusão de classes/grupos
 - Valores nunca apresentados para os atributos
 - Inclusive categóricos

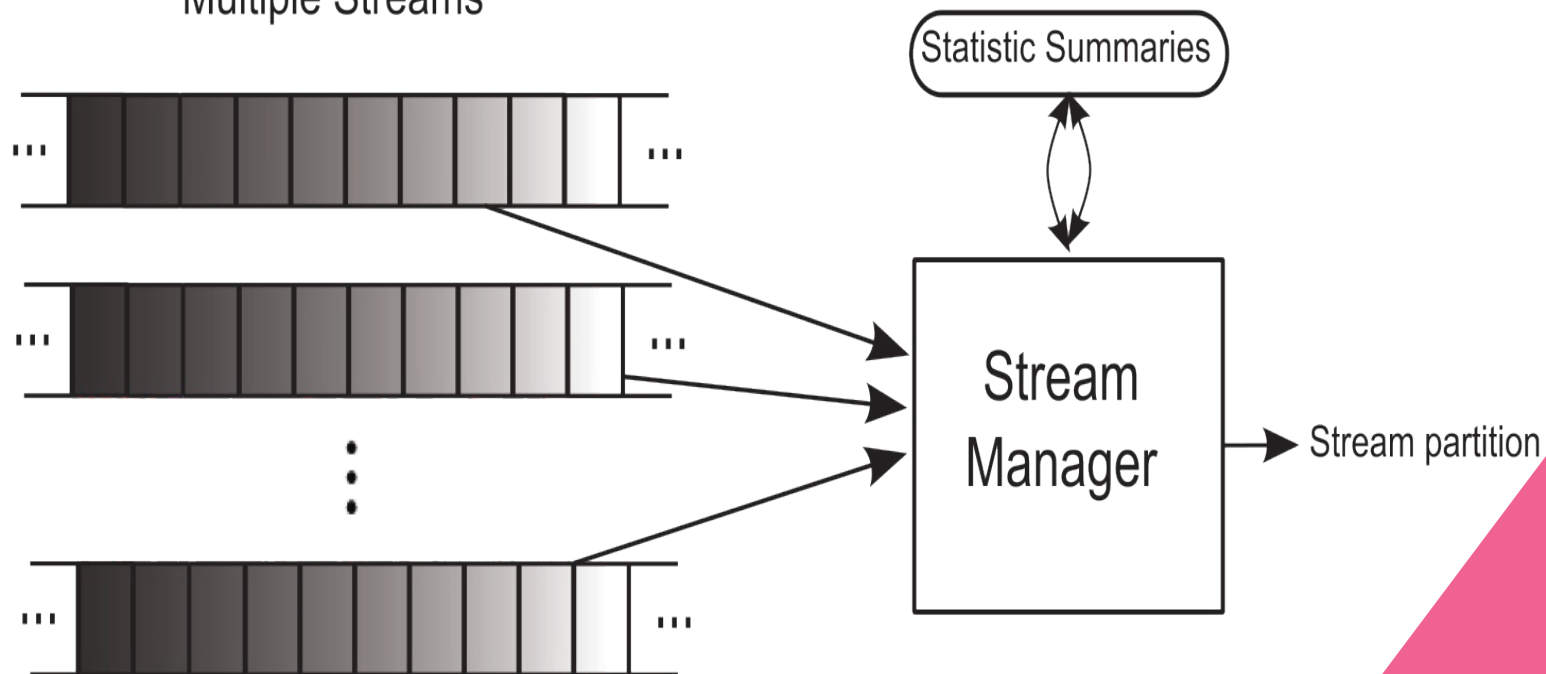


Leitura e armazenamento

Leitura pode ser um-a-um (mais comum) ou por *micro-batches*

- Sistema de *micro-batch* existe para evitar custo computacional excessivo
- Sistemas distribuídos, paralelos

Multiple Streams



Leitura e armazenamento

Depois da leitura, temos que considerar que o armazenamento é limitado

- Tanto para os dados quanto para um modelo
 - Algoritmos diferentes usam estratégias distintas
- É preciso ter um mecanismo de “esquecimento”
 - Não só porque a memória é limitada
 - Mas porque o cenário pode mudar



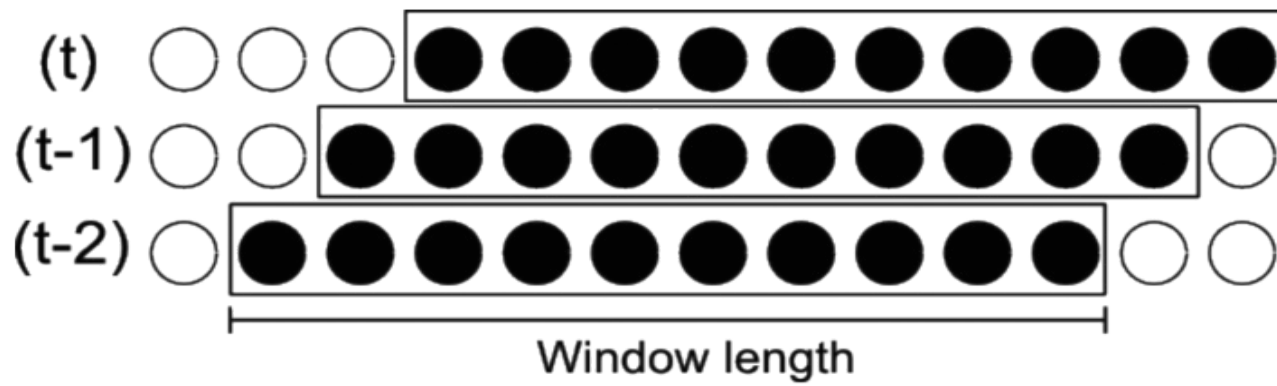


Fig. 6. Sliding-window model.

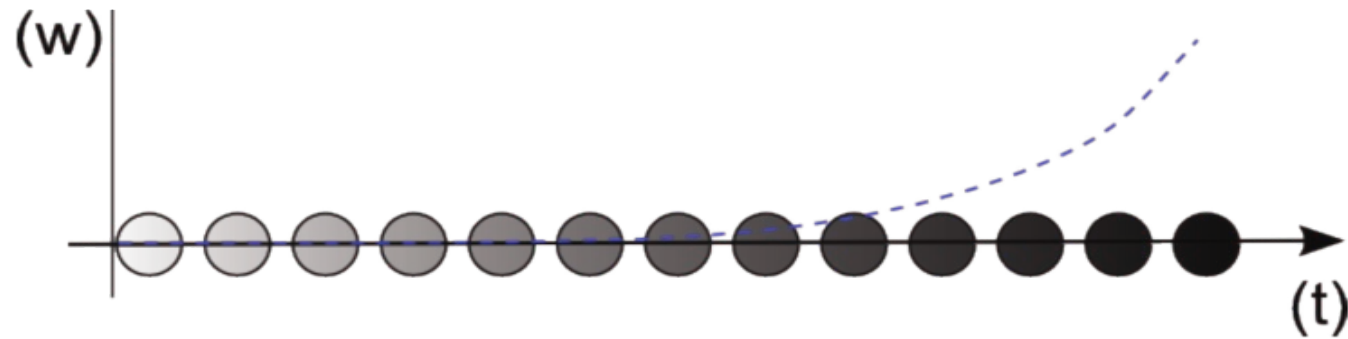


Fig. 7. Damped window model.

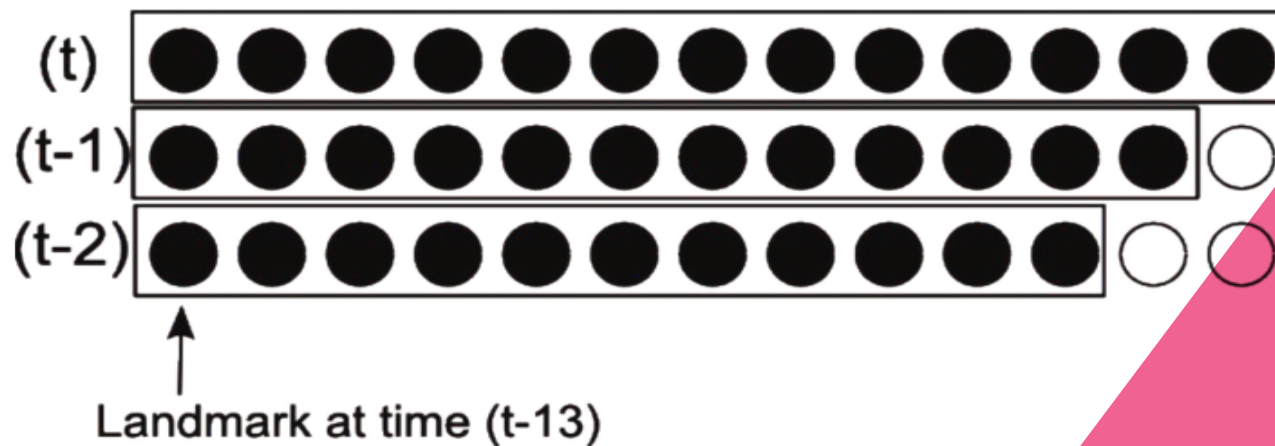


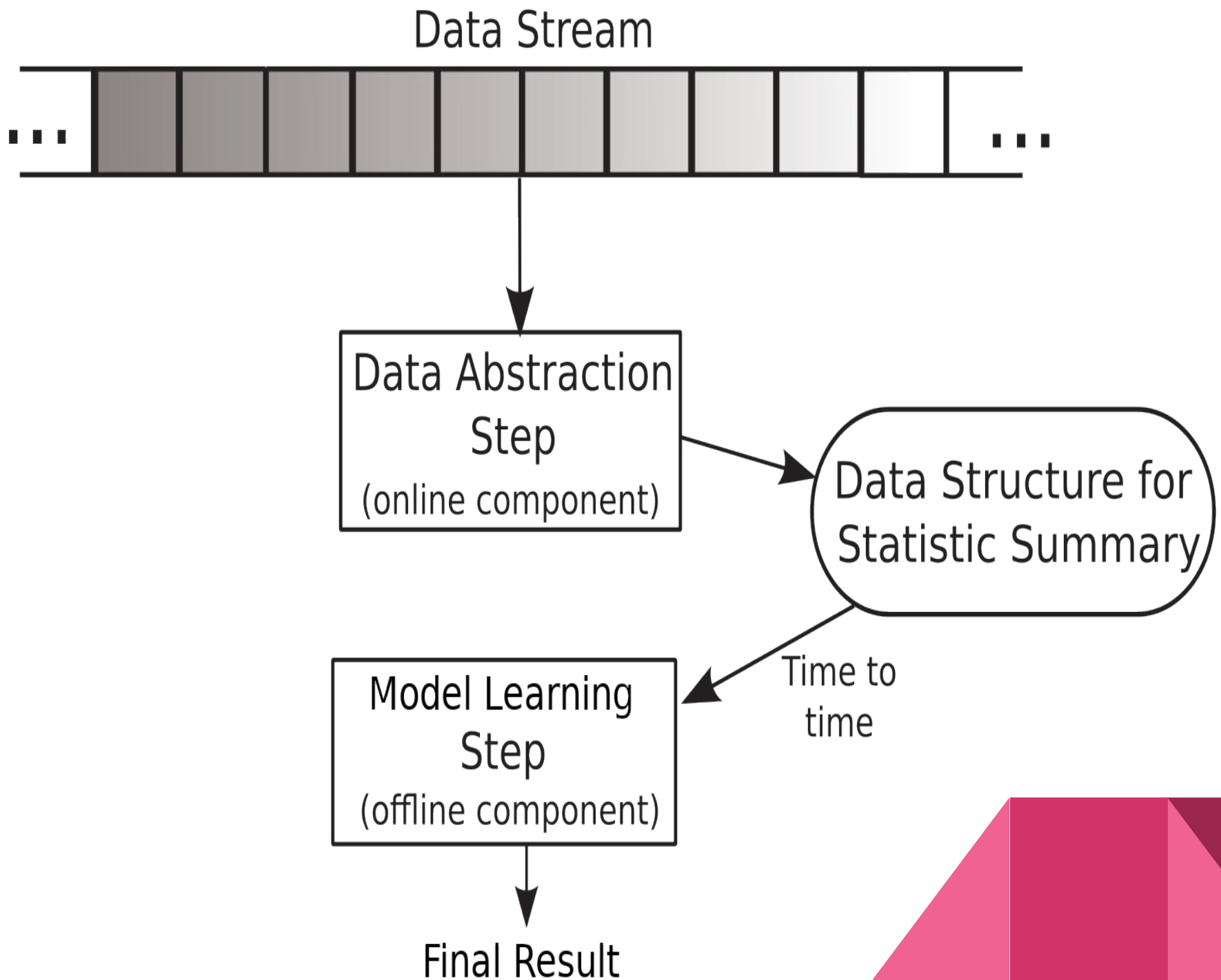
Fig. 8. Landmark window for a time interval of size 13.

Abstração dos dados

Em algumas aplicações, uma pequena e limitada quantidade de dados não é suficiente para gerar modelos

- Exemplos: classificadores, agrupamento
- Nesses casos, é preciso abstrair uma quantidade maior de dados em estruturas que possuam as informações necessárias para a execução bem sucedida do algoritmo



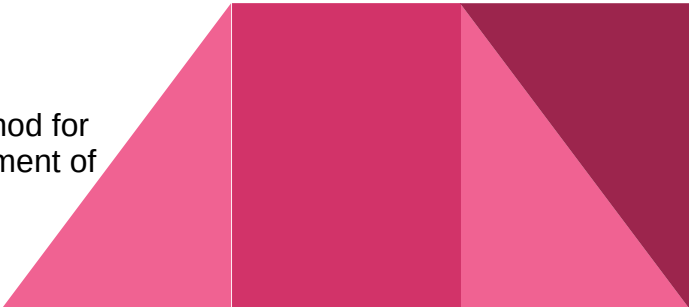


Exemplo abstração: *feature vector*

Vetor de características que resumem um subconjunto dos dados

- Exemplo: CF (*Cluster Feature Vector*) do BIRCH*
 - Contém: número de objetos N , soma linear dos objetos, LS , soma quadrada dos objetos, SS
- São informações incrementais ou adicionados, podendo ser utilizados para calcular informações importantes sobre os dados abstraídos

ZHANG, T., RAMAKRISHNAN, R., AND LIVNY, M. 1996. BIRCH: An efficient data clustering method for very large databases. In Proceedings of the ACM SIGMOD International Conference on Management of Data. ACM Press, New York, 103–114.



$$\textit{centroid} = \frac{LS}{N}$$

$$\textit{radius} = \sqrt{\left(\frac{SS}{N} - \left(\frac{LS}{N} \right)^2 \right)}$$

$$\textit{diameter} = \sqrt{\left(\frac{2N * SS - 2 * LS^2}{N(N - 1)} \right)}$$

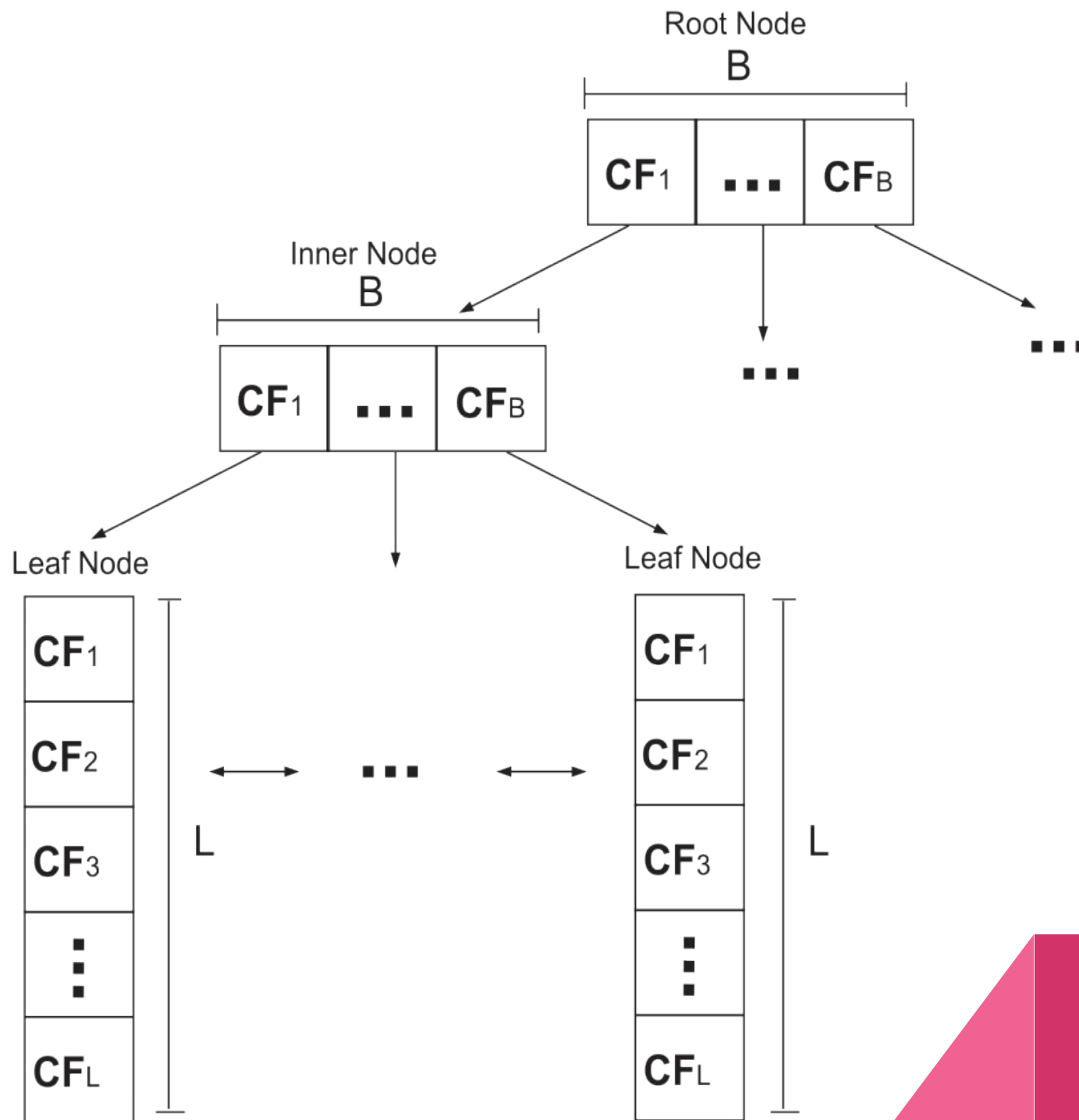
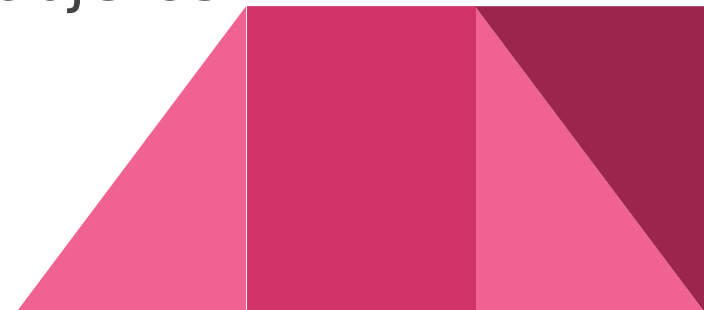


Fig. 3. CF tree structure.

Geração do modelo

Iniciar o modelo varia com o algoritmo e a tarefa:

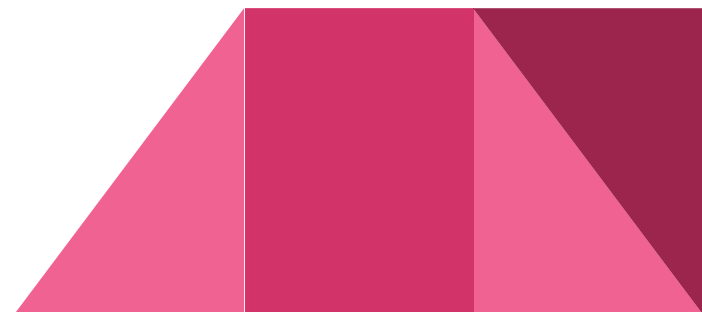
- Alguns algoritmos precisam de um conjunto inicial
 - Modelo inicial é gerado como em *batch*
- Outros são gerados em cima de abstrações
 - Que por sua vez precisam de um conjunto inicial
- Outros são completamente incrementais
 - Ainda assim necessitam de alguns objetos



Utilização do modelo

Pode acontecer de duas formas:

- *Online* : a tarefa necessita de resultados em tempo real
 - Modelo é utilizado a cada ciclo de tempo
 - Custo computacional VS tempo disponível
- *Offline* : a tarefa necessita de resultados sobre demanda
 - O modelo é gerado/atualizado quando requisitado
 - “Cópias” das abstrações são armazenadas



Classificação em FCDs

Com o modelo inicial gerado sobre um conjunto é precioso:

- Incorporar novas informações na velocidade que os dados chegam
- Esquecer a informação desatualizada
- **Detectar mudanças e se adaptar à informação mais recente**



Classificação em FCDs

Principal questão: Atualização do modelo

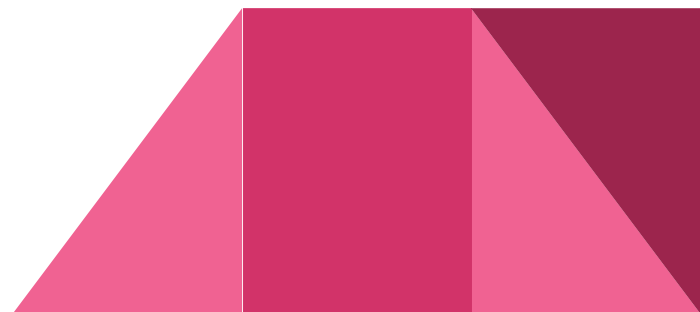
- Como atualizar o modelo de decisão?
 - Com *feedback*?
 - Sem *feedback*?
 - *Feedback* parcial?
- Quando atualizar o modelo de decisão?



Classificação em FCDs

Atualização do modelo: Primeiras propostas

- Propostas Supervisionadas: Supõe que o rótulo de todas as instâncias está disponível imediatamente após sua classificação para atualização do modelo
 - Problemas reais?



Classificação em FCDs

Atualização do modelo: Propostas mais recentes

- Propostas Não-supervisionadas: Supõem que nenhuma instância rotulada estará disponível para atualização
 - Assumir conceito de grupo ou outro conceito preestabelecido para as classes
 - Novos grupos são novas classes?



Classificação em FCDs

Atualização do modelo: Propostas mais recentes

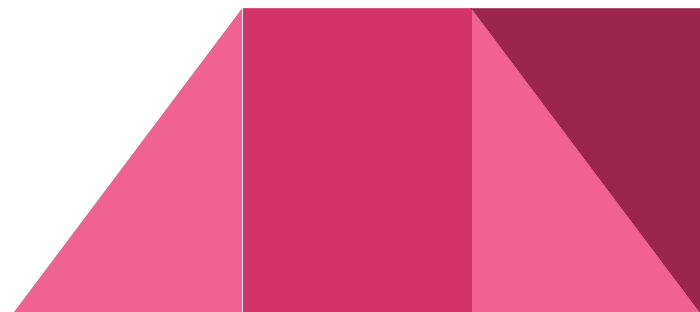
- Propostas Semi-Supervisão:
 - Usar instâncias rotuladas e não rotuladas para atualizar o modelo
 - Também assume estrutura de grupos
 - Porém é possível identificar o surgimento de novas classes ou mudança das classes já existentes



Classificação em FCDs

Atualização do modelo: Propostas mais recentes

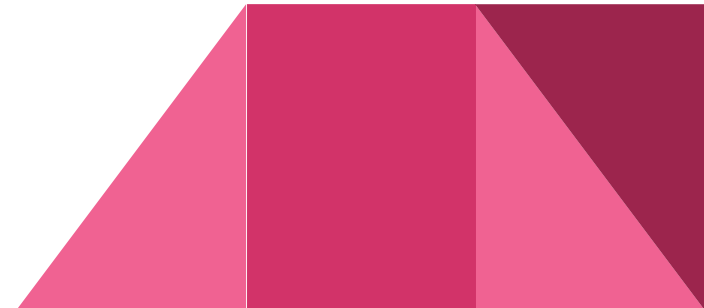
- Latência
 - Atraso na entrega das instâncias rotuladas
 - Caso extremo: latência infinita
 - Atraso na classificação (classificação com *delay*)



Classificação em FCDs

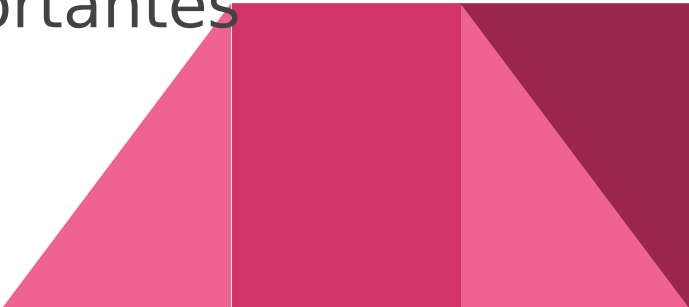
Atualização do modelo: Propostas mais recentes

- Aprendizado ativo (aula futura...)
 - Escolha do melhor conjunto de instâncias a serem rotuladas
 - Escolha aleatória
 - Escolha com base na confiança da classificação
 - Escolher os mais antigos
 - Escolher os mais recentes



Atualização de modelos

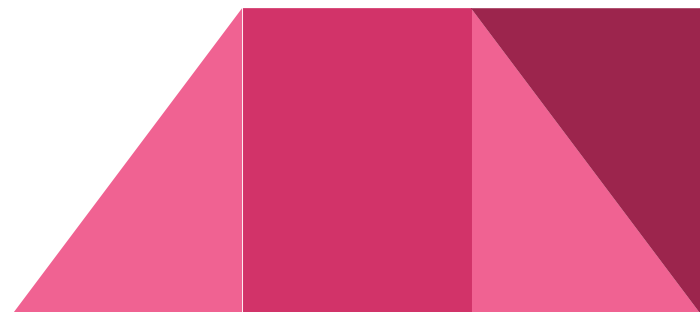
Desafios em desenvolver propostas que considerem:

- Que a tarefa de rotular tem um custo
 - Atraso na entrega de instâncias rotuladas
 - Que o especialista tem um limite de instâncias que ele consegue rotular
 - Rótulo para parte das instâncias
 - Que o modelo deve evoluir com ou sem *feedback*
 - Que algumas instâncias são mais importantes
- 

Exemplo ilustrativo - VFDT

Very Fast Decision Tree

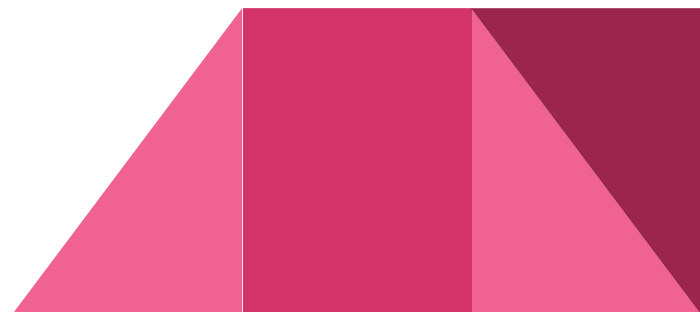
- Uma árvore de decisão (ah vá!)
- Se adapta dinamicamente
- Decide expandir cada nó de acordo com o teste de *Hoeffding* (por isso conhecida como *Hoeffding Tree*)
 - Substitui uma folha por um nó de decisão
 - Melhor a pureza, evitar divisão excessivas
- Somente para latência 0
- Assume que a distribuição não muda!
 - Não se adapta a *concept drift*
 - “Resolvido” pelo CVFDT



Exemplo ilustrativo - SCARGC

Stream Classification Algorithm Guided by Clustering – SCARGC

- Lida com latência infinita (extrema) e mudança incremental
- Cria modelos para o *batch* inicial
- “Segue” a distribuição dos dados com k-means
 - Associa os novos exemplos a um grupo
 - Assume mesmo rótulo
- Atualiza o modelo quando há grande discordância
 - não admite novas classes



Agrupamento em FCDs

Agrupamento geralmente é dividido em duas fases:

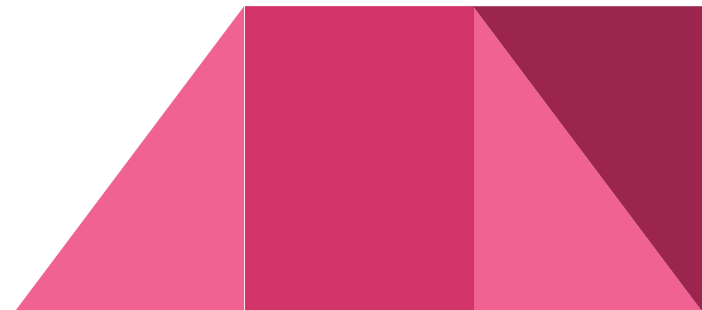
- *Online*: abstração dos dados em micro-grupos (*CFs*)
 - Tipo depende do conceito de grupo (similaridade, densidade)
- *Offline*:
 - Aplicação do algoritmo de agrupamento e obtenção do resultado
- Exemplos:
 - CluStream, DenStream...



Detecção de anomalias em FCDs

Detecção da anomalias necessita de modelo atualizado *online*

- Durante a atualização o objeto é rotulado como *outlier*
 - Pode ser mudança de conceito
 - Tem que ser detectado e atualizado
 - *Outliers* devem ficar em um *buffer*
- Modelo de abstração ajuda
 - Relação entre o conceito de grupos e anomalias



Do nada...mudou...



Referências

- Gama, J. A survey on learning from data streams: current and future trends. Prog Artif Intell 1, 45–55 (2012). <https://doi.org/10.1007/s13748-011-0002-6>
- Data stream clustering: A survey. JA Silva, ER Faria, RC Barros, ER Hruschka, AC Carvalho, J Gama ACM Computing Surveys (CSUR) 46 (1), 1-31
- Geoff Hulten, Laurie Spencer, and Pedro Domingos. 2001. Mining time-changing data streams. In Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '01). Association for Computing Machinery, New York, NY, USA, 97–106. <https://doi-org.ez31.periodicos.capes.gov.br/10.1145/502512.502529>
- SOUZA, Vinícius MA et al. Data stream classification guided by clustering on nonstationary environments and extreme verification latency. In: Proceedings of the 2015 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics, 2015. p. 873-881.