



Mudança de Conceito

Definição, técnicas e ferramentas

Ruivaldo Neto

UFBA - PGCOMP

1. Introdução
2. Formalização
3. Tipos, Padrões e Taxonomia
4. Técnicas de detecção
5. Ferramentas
6. Conclusão

Introdução

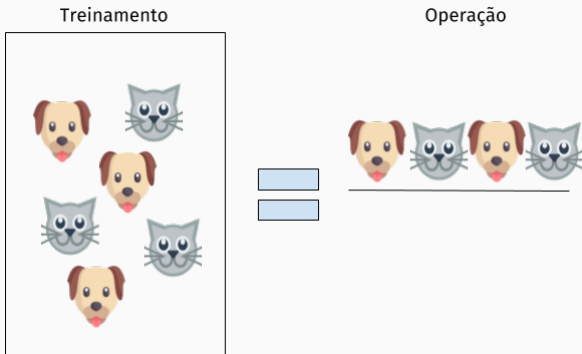
Algoritmos de aprendizagem não melhoram pelo simples passar do tempo.



Introdução

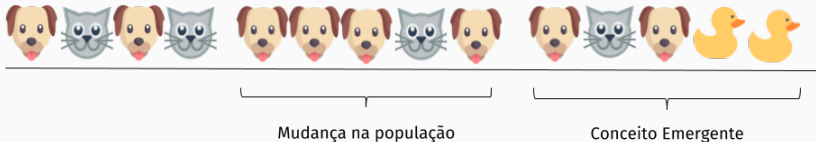
Parte significativa das técnicas foi projetada para cenários **estacionários**, onde se assume que:

distribuição de treinamento = distribuição de operação.



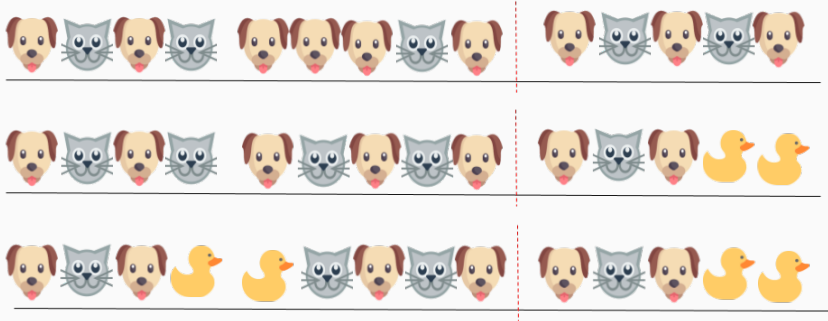
Introdução

As aplicações reais são, em sua maioria, **não-estacionárias**, isto é:
distribuição de treinamento \neq distribuição de operação.



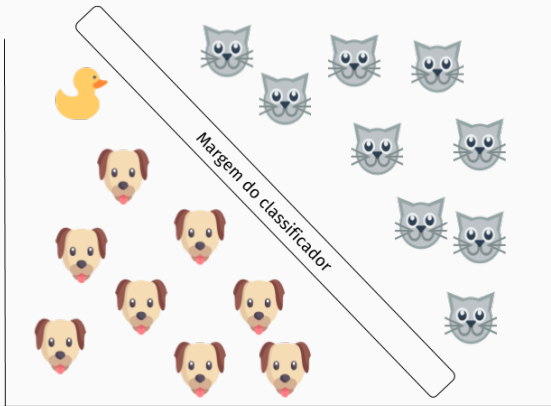
Introdução

Uma solução comum, mas **pobre**, é a execução de treinamentos recorrentes.



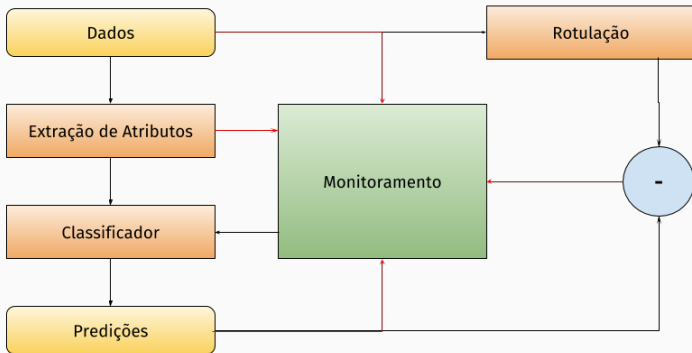
Introdução

Técnicas de aprendizado ativo diminuem o custo de rotulação, mas não são capazes de lidar com mudanças de conceito.



Introdução

Solução: **monitoramento**.



Formalização

A Teoria Bayesiana é comumente aplicada para descrever as tarefas de classificação [10]. A partir dela, também é possível formalizar o problema de mudança de conceito.

Considerando $X \in \mathbb{R}^p$ uma instância em um espaço p -dimensional de atributos e $X \in c_i$ onde c_1, c_2, \dots, c_k é o conjunto de classes, o classificador ótimo para classificar $x \rightarrow c_i$ é determinado pelas probabilidades a priori das classes, $P(c_i)$, e a função de densidade de probabilidade condicionada às classes, $p(X|c_i)$, para $i = 1, \dots, k$.

Dessa forma, um conceito pode ser definido como um conjunto de probabilidades a priori e condicionais das classes (Eq. 1):

$$S = \{(P(c_1), P(X|c_1)), (P(c_2), P(X|c_2)), \dots, (P(c_k), P(X|c_k))\} \quad (1)$$

Ainda segundo a Teoria Bayesiana, a classificação baseada na máxima probabilidade a posteriori de uma instância X pode ser obtida através da Equação 2:

$$p(c_i|X) = \frac{p(c_i) * p(X|c_i)}{p(X)} \quad (2)$$

Sendo possível definir formalmente que há mudança de conceito entre os instantes t_0 e t_1 se:

$$\exists X : p_{t_0}(X, c) \neq p_{t_1}(X, c) \quad (3)$$

onde, p_{t_0} e p_{t_1} denotam as distribuições de probabilidades conjuntas nos instantes t_0 e t_1 , respectivamente, para X e c [13].

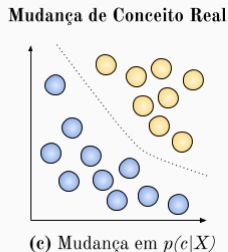
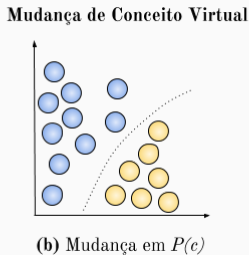
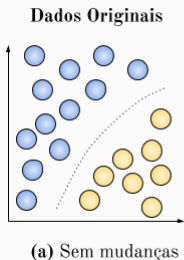
Com base na formalização exposta [29], pode-se definir que mudanças de conceito ocorrem devido a:

1. Alterações na probabilidade a priori das classes $P(c)$;
2. Alterações na distribuição de uma ou mais classes $p(X|c)$;
3. Alterações nas distribuições a posteriori das classes $p(c|X)$.

Tipos, Padrões e Taxonomia

Tipos

- **Virtuais:** indicam mudanças na probabilidade a priori das classes, $P(c)$, e não têm impacto nos conceitos-alvo.
- **Reais:** referem-se a mudanças na probabilidade a posteriori, $p(c|X)$, e afetam os conceitos-alvo.



As mudanças de conceito podem ocorrer de forma abrupta, gradual, incremental ou recorrente [13]:



(a) Abrupta



(b) Gradual



(c) Incremental



(d) Recorrente

Termos correspondentes a **mudança de conceito** em cada área de pesquisa [29]:

Área	Termos
Mineração de Dados	Mudança de Conceito
Aprendizado de Máquina	Mudança de Conceito, Mudança de Covariável
Computação Evolucionária	Ambiente Evolutivo, Ambiente em Mudança
IA e Robótica	Ambiente Dinâmico
Estatísticas, Séries Temporais	Não Estacionário
Recuperação de Informação	Evolução Temporal

Os termos Detecção de *Outliers*, Detecção de Novidade, Detecção de *Change Points* e Detecção de Mudança de Conceito, apesar de próximos, possuem significados diferentes:

- **Detecção de Outliers:** identificam padrões em desacordo com o comportamento esperado [5].
- **Detecção de Novidade:** identificam padrões ainda não observados e incorporam ao modelo [5].
- **Detecção de Change Points:** identificam variações abruptas de valor, que podem representar transições entre estados, em séries temporais unidimensionais estacionárias [1].

Técnicas de detecção

- **Algoritmos Explícitos/Supervisionados:** Dependem da rotulação dos dados, pois são utilizados no cálculo das medidas de performance.
- **Algoritmos Implícitos/Não Supervisionados:** Independem de rotulação. Baseiam-se nas características dos dados.

Os algoritmos supervisionados dividem-se em três grupos principais [13]:

- **Métodos Baseados em Análise Sequencial:** Monitoram os resultados das predições (taxa de erro).
- **Abordagens baseadas em Estatística:** Monitoram os resultados das predições conforme parâmetros estatísticos (média, desvio padrão, etc).
- **Métodos baseados em Janelas:** Mantém o sumário da distribuição de uma janela passada como treinamento. Comparam este sumário com a distribuição da janela atual.

Os algoritmos não supervisionados também foram divididos em três grupos [14]:

- **Detecção de Novidade / Métodos de Agrupamento:** Utilizam a distância e/ou a densidade dos dados para detectar novos padrões.
- **Monitoramento de distribuição multivariada:** Monitoram diretamente a distribuição dos dados, considerando seus atributos.
- **Monitoramento dependente de modelo:** Restringem-se aos classificadores probabilísticos, pois monitoram as probabilidades a posteriori.

Técnicas de detecção - Resumo

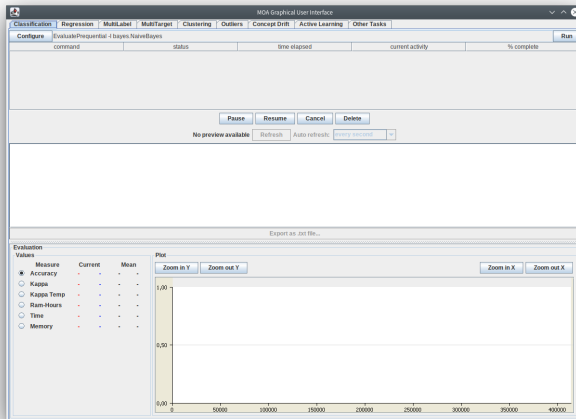
Resumo das categorias, grupos e técnicas:

Algoritmos Explícitos/Supervisionados	Métodos Baseados em Análise Sequencial	Cumulative Sum (CUSUM) PageHinkley (PH) [21] Geometric Moving Average (GMA) [24]
	Abordagens baseadas em Estatística	Drift Detection Method (DDM) [12] Early Drift Detection Method (EDDM) [2] Exponentially Weighted Moving Average (EWMA) [25] Reactive Drift Detection Method (RDDM) [6]
	Métodos baseados em Janelas	Adaptive Windowing (ADWIN) [3] SeqDrift [22] HDDMA/HDDMW [4]
Algoritmos Implícitos/Não Supervisionados	Detecção de Novidade / Métodos de Agrupamento	OLINDDA [28] MINAS [11] Woo [26] DETECTNOD [16] ECSMiner [20] GC3 [27]
	Monitoramento de distribuição multivariada	CoC [18] HDDDM [7] PCA-detect [17]
	Monitoramento dependente de modelo	A-distance [8] CDBD [19] Margin [9]

Ferramentas

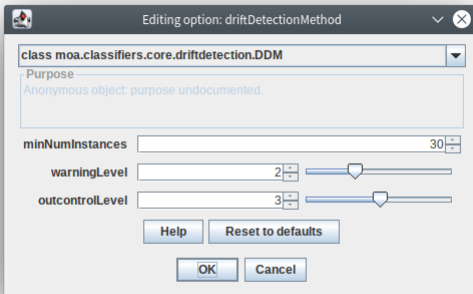
MOA – Massive Online Analysis

- Principal framework para mineração de dados em fluxos contínuos.
- Código-aberto¹ e multi-plataforma (Java).
- Grande base de algoritmos e integração com o WEKA [15].
- Ferramentas para avaliação de desempenho desses algoritmos.



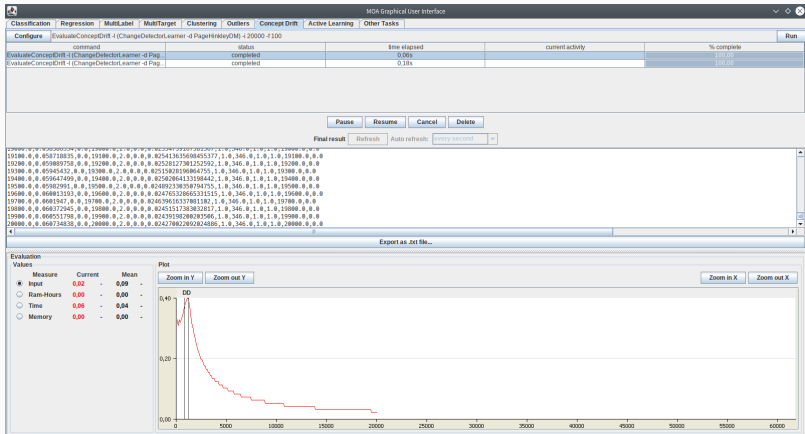
MOA - Massive Online Analysis

- Novos detectores de mudança de conceito podem ser criados estendendo a classe abstrata `AbstractChangeDetector`.
- Interface de configuração é criada dinamicamente, a partir dos atributos da classe.



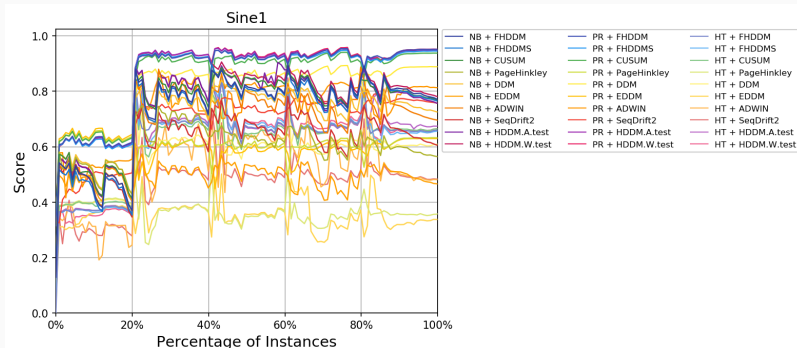
MOA - Massive Online Analysis

- Algoritmos para detecção de mudança de conceito podem ser avaliados pelos métodos DriftDetectionMethodClassifier ou BasicConceptDriftPerformanceEvaluator.



Tornado

- Framework para avaliação de pares de classificador e detector ao longo do tempo [23].
- Código-aberto² e multi-plataforma (Python).
- Grande base de algoritmos implementados.



²<https://github.com/alipsgh/tornado>

Conclusão

Obrigado! Dúvidas ?



S. Aminikhanghahi and D. J. Cook.

A survey of methods for time series change point detection.

Knowl. Inf. Syst., 51(2):339–367, May 2017.



M. Baena-García, J. del Campo-Ávila, R. Fidalgo, A. Bifet,
R. Gavaldá, and R. Morales-Bueno.

Early drift detection method.

In *In Fourth International Workshop on Knowledge Discovery from Data Streams*, 2006.



A. Bifet and R. Gavaldà.

Learning from time-changing data with adaptive windowing.

In *SDM*, pages 443–448. SIAM, 2007.



I. I. F. Blanco, J. del Campo-Ávila, G. Ramos-Jiménez, R. M. Bueno, A. A. O. Díaz, and Y. C. Mota.

Online and non-parametric drift detection methods based on hoeffding's bounds.

IEEE Trans. Knowl. Data Eng., 27(3):810–823, 2015.



V. Chandola, A. Banerjee, and V. Kumar.

Anomaly detection: A survey.

ACM Comput. Surv., 41(3):15:1–15:58, July 2009.



R. S. M. de Barros, D. R. de Lima Cabral, P. M. G. Jr., and S. G. T. de Carvalho Santos.

RDDM: reactive drift detection method.

Expert Syst. Appl., 90:344–355, 2017.



G. Ditzler and R. Polikar.

Hellinger distance based drift detection for nonstationary environments.

In 2011 IEEE Symposium on Computational Intelligence in Dynamic and Uncertain Environments (CIDUE), pages 41–48, April 2011.



M. Dredze, T. Oates, and C. Piatko.

We're not in kansas anymore: Detecting domain changes in streams.

pages 585–595, 2010.

cited By 13.



A. Dries and U. Rückert.

Adaptive concept drift detection.

Statistical Analysis and Data Mining: The ASA Data Science Journal, 2(5-6):311–327, 2009.



R. O. Duda, P. E. Hart, and D. G. Stork.

Pattern Classification (2Nd Edition).

Wiley-Interscience, New York, NY, USA, 2000.



E. R. Faria, J. a. Gama, and A. C. P. L. F. Carvalho.

Novelty detection algorithm for data streams multi-class problems.

In *Proceedings of the 28th Annual ACM Symposium on Applied Computing, SAC '13*, pages 795–800, New York, NY, USA, 2013. ACM.



J. Gama, P. Medas, G. Castillo, and P. P. Rodrigues.

Learning with drift detection.

In A. L. C. Bazzan and S. Labidi, editors, *SBIA*, volume 3171 of *Lecture Notes in Computer Science*, pages 286–295. Springer, 2004.



J. a. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia.

A survey on concept drift adaptation.

ACM Comput. Surv., 46(4):44:1–44:37, Mar. 2014.



P. M. Gonçalves, S. G. de Carvalho Santos, R. S. Barros, and D. C. Vieira.

A comparative study on concept drift detectors.

Expert Systems with Applications, 41(18):8144 – 8156, 2014.



M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten.

The weka data mining software: An update.

SIGKDD Explor. Newsl., 11(1):10–18, Nov. 2009.



M. Z. Hayat and M. R. Hashemi.

A dct based approach for detecting novelty and concept drift in data streams.

In 2010 International Conference of Soft Computing and Pattern Recognition, pages 373–378, Dec 2010.



L. Kuncheva.

Classifier ensembles for detecting concept change in streaming data: Overview and perspectives.

Proc. Eur. Conf. Artif. Intell., pages 5–10, 2008.

cited By 70.



J. Lee and F. Magoulès.

Detection of concept drift for learning from stream data.

In 2012 IEEE 14th International Conference on High Performance Computing and Communication 2012 IEEE 9th International Conference on Embedded Software and Systems, pages 241–245, June 2012.



P. Lindstrom, B. Mac Namee, and S. J. Delany.

Drift detection using uncertainty distribution divergence.

Evolving Systems, 4(1):13–25, Mar 2013.



M. Masud, J. Gao, L. Khan, J. Han, and B. M. Thuraisingham.

Classification and novel class detection in concept-drifting data streams under time constraints.

IEEE Trans. on Knowl. and Data Eng., 23(6):859–874, June 2011.



E. S. Page.

Continuous Inspection Schemes.

Biometrika, 41(1/2):100–115, 1954.



R. Pears, S. Sakthithasan, and Y. S. Koh.

Detecting concept change in dynamic data streams - A sequential approach based on reservoir sampling.

Machine Learning, 97(3):259–293, 2014.



A. Pesaranghader.

A reservoir of adaptive algorithms for online learning from evolving data streams, 2018.



S. W. Roberts.

Control chart tests based on geometric moving averages.

Technometrics, 42(1):97–101, Feb. 2000.



G. J. Ross, N. M. Adams, D. K. Tasoulis, and D. J. Hand.
Exponentially weighted moving average charts for detecting concept drift.

Pattern Recogn. Lett., 33(2):191–198, Jan. 2012.



J. W. Ryu, M. M. Kantardzic, M.-W. Kim, and A. Ra Khil.
An efficient method of building an ensemble of classifiers in streaming data.

In S. Srinivasa and V. Bhatnagar, editors, *Big Data Analytics*, pages 122–133, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.



T. S. Sethi, M. Kantardzic, and H. Hu.
A grid density based framework for classifying streaming data in the presence of concept drift.

Journal of Intelligent Information Systems, 46(1):179–211, Feb 2016.



E. J. Spinosa, A. P. de Leon F. de Carvalho, and J. a. Gama.

Olindda: A cluster-based approach for detecting novelty and concept drift in data streams.

In *Proceedings of the 2007 ACM Symposium on Applied Computing*, SAC '07, pages 448–452, New York, NY, USA, 2007. ACM.



I. Zliobaite.

Learning under concept drift: an overview.

CoRR, abs/1010.4784, 2010.