Sentiment Knowledge Discovery in Twitter Streaming Data

Albert Bifet and Eibe Frank (2010)

Angermüller Christof

LMU, Lehr- und Forschungseinheit für Datenbanksysteme

November 14, 2012

1. Introduction
2. Methods
3. Results
4. Discussion

Twitter

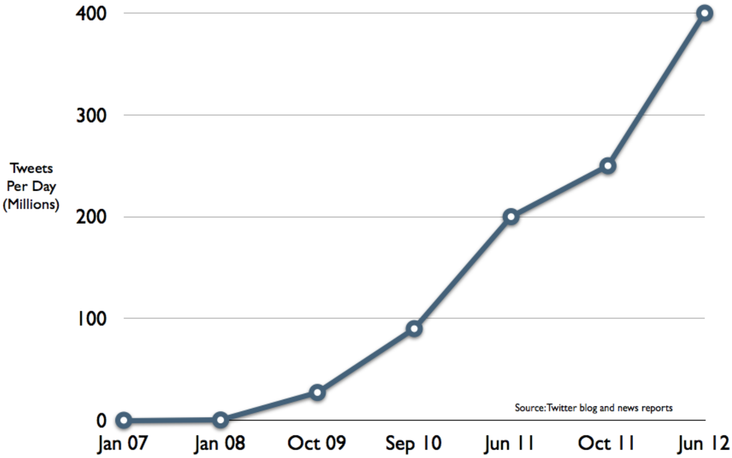
iPhoneAppCafe @iPhoneAppCafe 19m
iOS 6.0.1 jailbreak misery after iOS 6 on **iPhone** 5, 4S - Product Reviews: Product ReviewsiOS 6.0.1 jailbreak mis... bit.ly/Z2gozs
Öffnen

Andreas Heuer @DubaiHeuer 19m
hat ein @runtastic Livetracking gestartet. Schau dir an wo ich bin und feuere mich an. - bit.ly/Uf511r #runtastic #iphone
Öffnen

iStore Balikpapan @iStoreBPN 23m
#iPhone5 iOS 6.0.1 jailbreak misery after iOS 6 on **iPhone** 5, 4S - Product Reviews ow.ly/2t8tJq

- Social networking - microblogging service
- Tweets
  - up to 140 characters long text message
  - **@runtastic**: user name
  - **#iPhone**: subject or category
  - **RT** *I like my iPhone*: retweet - repetition of tweet

Twitter: Tweets per day



Source: Twitter blog and news reports

Twitter: Statistics

Statistics

- More than 140 mil users
- More than 400 mil tweets per day
- Exponential growth

Makes Twitter attractive for

- Companies
- Politicians
- **Data mining**

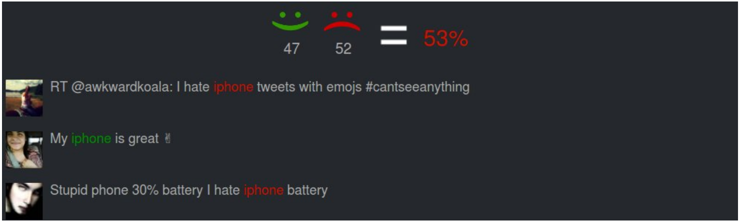
Sentiment analysis

Definition (Sentiment analysis)

- Given a tweet
- Predict its sentiment
  - positive
  - (neutral)
  - negative

Twitter sentiment analysis

- sentiment140.com
- tweetfeel.com
- tweettone.com



RT @awkwardkoala: I hate iphone tweets with emojs #cantseeanything

My iphone is great ✌

Stupid phone 30% battery I hate iphone battery

Challenges learning from Twitter streams

- Twitter API provides access to all tweets
- Incremental learning instead of Batch learning required

1. **Time efficiency**: samples arrive with high rate
2. **Memory efficiency**: infinite training set
3. **Concept drift**: sentiments change dynamically
4. **Evaluation**: concept drift and unbalanced classes

1. Introduction
2. **Methods**
3. Results
4. Discussion

Classification

Problem description

- **Given**: training set $\left\{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\right\}$
  - $x^{(i)}$: data (unigram, bigram)
  - $y^{(i)}$: class ($y^{(i)} = +1$: positive, $y^{(i)} = -1$: negative)
- **Wanted**: hypothesis $h_\theta(x) \rightarrow y$
  - minimizes some cost function $J(\theta)$

Classification

Problem description

- **Given**: training set $\{(x^{(1)}, y^{(1)}), ..., (x^{(m)}, y^{(m)})\}$
    - $x^{(i)}$: data (unigram, bigram)
    - $y^{(i)}$: class ($y^{(i)} = +1$: positive, $y^{(i)} = -1$: negative)
- **Wanted**: hypothesis $h_\theta(x) \rightarrow y$
    - minimizes some cost function $J(\theta)$

Models for sentiment classification

- Naïve Bayes
- SVM
- Hoeffding tree
- Logistic regression
- ...

Classification

Problem description

- **Given**: training set $\left\{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\right\}$
    - $x^{(i)}$: data (unigram, bigram)
    - $y^{(i)}$: class ($y^{(i)} = +1$: positive, $y^{(i)} = -1$: negative)
- **Wanted**: hypothesis $h_\theta(x) \rightarrow y$
    - minimizes some cost function $J(\theta)$

Models for sentiment classification

- Naïve Bayes
- SVM
- Hoeffding tree
- Logistic regression
- ...

Naïve Bayes

Prediction

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)} \propto \prod_{j=1}^{n} P(w_j|y)^{x_j} P(y)$$

$$h_\theta(x) = \text{argmax}_{y'} P(y'|x)$$

Training

$P(w_j\|y)$	Prob. of word w_j in tweet with sentiment y
$P(y)$	Prior prob. of sentiment y

- Can be easily approximated by the relative frequency
- Count word w_j sentiment y
- ⇒ Simple incremental implementation

Batch Gradient Descent

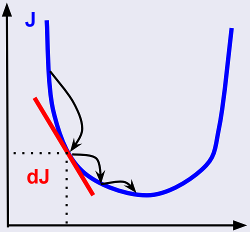
Idea

- Minimize cost function $J(\theta) = \sum_{i=1}^{m} l(x^{(i)}, y^{(i)}, \theta)$
- Go into direction of steepest descent $\nabla J(\theta)$

Algorithm

- Initialize θ^0 randomly
- Loop until convergence:

$$\theta^{t+1} = \theta^t - \alpha \nabla J(\theta^t)$$
$$= \theta^t - \alpha \sum_{i=1}^{m} \nabla l(x^{(i)}, y^{(i)}, \theta^t)$$



Stochastic Gradient Descent

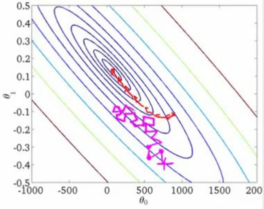
Idea

- Compute gradient of a single random sample
- More efficient than Batch Gradient Descent

Algorithm

- Initialize θ^0 randomly
- Loop until convergence:
  - Shuffle training set randomly
  - for $i = 1$ to m

$$\theta^{t+1} = \theta^t - \alpha \nabla l(x^{(i)}, y^{(i)}, \theta^t)$$



Stochastic Gradient Descent

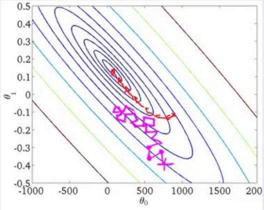
Idea

- Compute gradient of a single random sample
- More efficient than Batch Gradient Descent

Algorithm

- Initialize θ^0 randomly
- Loop until convergence:
    - Shuffle training set randomly
    - for $i = 1$ to m

$$\theta^{t+1} = \theta^t - \alpha \nabla l(x^{(i)}, y^{(i)}, \theta^t)$$



$\Rightarrow$ Simple incremental update for new $(x^{(i)}, y^{(i)})$

SVM: Support Vector Machine

Prediction

Given weight vector θ

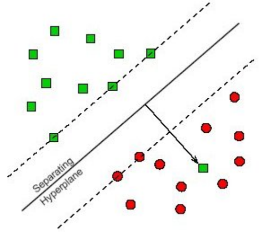
$$h_\theta(x) = \begin{cases} +1 & \theta^T x \geq 0 \\ -1 & \theta^T x < 0 \end{cases}$$

Training

- Hinge loss function:

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} max \left\{ 1 - y^{(i)} \theta^T x^{(i)}, 0 \right\}$$

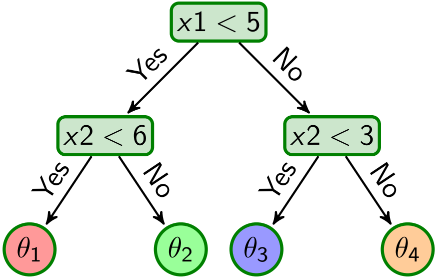
- θ can be updated incrementally via SGD

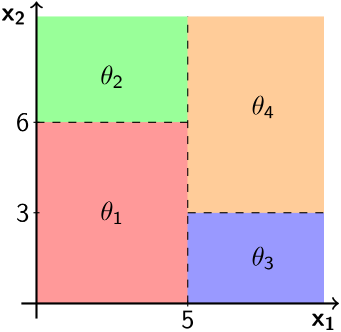


Hoeffding Tree

Prediction

- Each node tests an attribute
- Each edge refers to an attribute value
- $h_\theta(x) =$ *'label y of leaf to which x is assigned to'*

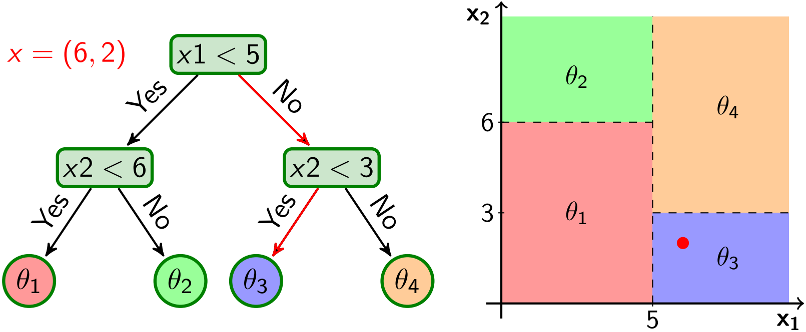




Hoeffding Tree

Prediction

- Each node tests an attribute
- Each edge refers to an attribute value
- $h_\theta(x) =$ *'label* y *of leaf to which* x *is assigned to'*

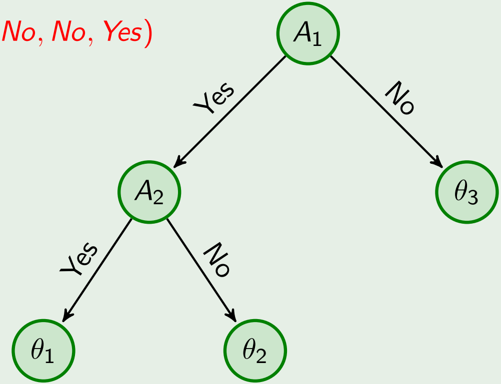


Hoeffding Tree

Incremental tree induction

Top-down model tree induction

$x = (No, No, Yes)$



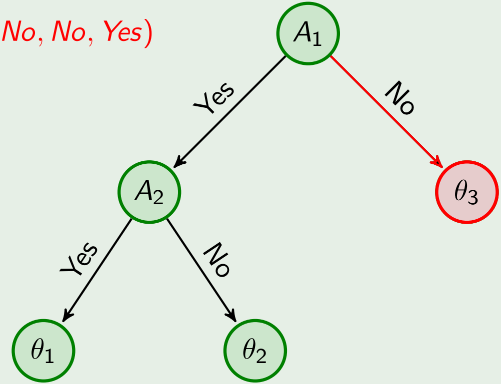
Hoeffding Tree

Incremental tree induction

1. Assign (x, y) to leaf

Top-down model tree induction

$x = (No, No, Yes)$



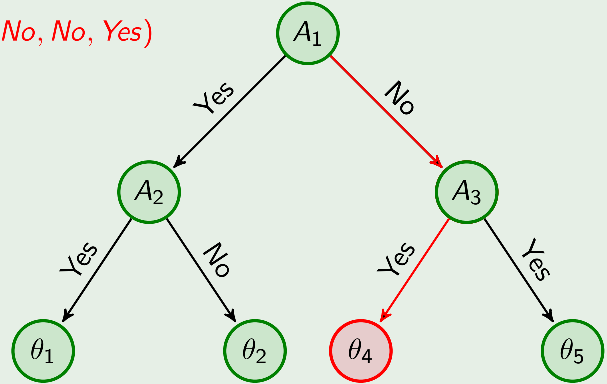
Hoeffding Tree

Incremental tree induction

1. Assign (x, y) to leaf
2. Use Hoeffding bound as split criterion

Top-down model tree induction

$x = (No, No, Yes)$



Hoeffding Tree

Incremental tree induction

1. Assign (x, y) to leaf
2. Use Hoeffding bound as split criterion

$$\Delta G = G(X_a) - G(X_b) > \epsilon$$

$$\epsilon = \ln \sqrt{\frac{R^2 ln(1/\alpha)}{2n}}$$

$\Rightarrow$ **Hypothesis testing with significance level** α

Labeling tweets

Question

How to label millions of tweets?

Labeling tweets

Question

How to label millions of tweets?

Answer

Emoticons :-)

Positive sentiment

- $x =$ '*I love my iPhone more than my girlfriend* **:-)**'
- $y = +1$

Negative sentiment

- $x =$ '*My iPhone is broken* **:-(**'
- $y = -1$

Data preparation (Go et al., sentiment140.com)

1. Retrieve tweets via Twitter API
2. Discard tweets without emoticons
3. Define $y = +1/-1$ with positive/negative emoticon
4. Feature reduction
    - Remove emoticons
    - @john → USER
    - www.google.com → URL
    - heeeello → hello
5. Define x with unigrams/bigrams

Training and Test set

Holdout validation

Prequential / Interleaved test-then-train

Training and Test set

Holdout validation

1. Divide data into training set (60%) and test set (40%)
2. Train model on training set
3. Measure performance on test set

⇒ Static test set

Prequential / Interleaved test-then-train

Training and Test set

Holdout validation

1. Divide data into training set (60%) and test set (40%)
2. Train model on training set
3. Measure performance on test set

⇒ Static test set

Prequential / Interleaved test-then-train

New sample (x, y) arrives...

1. Test $h_\theta(x) \overset{?}{=} y$ and update performance metric
2. Update θ with (x, y)

⇒ Dynamic test set

Accuracy

$$\begin{array}{cc|c|c|c|c}
 & & \multicolumn{3}{c}{f(x)} & \\
 & & 1 & \cdots & L & \\
\hline
 & 1 & o_{1,1} & \cdots & o_{1,L} & r_1 \\
\hline
h_\theta(x) & \vdots & \vdots & \ddots & \vdots & \vdots \\
\hline
 & L & o_{L,1} & \cdots & o_{L,L} & r_L \\
\hline
 & & c_1 & \cdots & c_L & m
\end{array}$$

Definition (Accuracy)

$$p_0 = \frac{1}{m} \sum_{i=1}^{L} o_{i,i}$$

$\Rightarrow$ Fraction of correct predictions

Accuracy

Unbalanced classes

- Unbalanced (skewed) class distribution
- Can lead to high accuracy

Accuracy

Unbalanced classes

- Unbalanced (skewed) class distribution
- Can lead to high accuracy

Cancer prediction

- $y = -1$: person has no cancer (99%)
- $y = +1$: person has cancer (1%)

- $h_\theta(x) = -1$, i.e. predict always *no cancer*
- $\Rightarrow p_0 = 99\%$

κ statistic (Cohen, 1960)

Definition (*kappa* statistic)

$$\frac{r_i}{m} = \text{predicted frequency of class } i$$

$$\frac{c_i}{m} = \text{actual frequency of class } i$$

$$p_c = \sum_{i=1}^{L} \frac{r_i}{m} \frac{c_i}{m} = \text{probability of being correct by chance}$$

$$\kappa = \frac{p_0 - p_c}{1 - p_c}$$

κ: accuracy relative to change prediction

- $p_o = p_c \to \kappa = 0$
- $p_o = 1 \to \kappa = 1$

κ statistic

Cancer prediction

- $y = -1$: person has no cancer (99%)
- $y = +1$: person has cancer (1%)

- $h_\theta(x) = -1$, i.e. predict always *no cancer*
- $p_0 = 0.99$
- $p_c = 1.00 * 0.99 + 0.00 * 0.01 = 0.99$
- $\Rightarrow \kappa = \frac{p_0 - p_c}{1 - p_c} = 0.0$

κ statistic

Incremental update

Variables

Only $2L + 1$ variables

- r_i, c_i for $i = 1, ..., L$: row, column sums
- p_0: accuracy

Can be updated incrementally for a new (x, y)

Only consider most recent samples

- **Sliding window**: store last w samples
- **Fading factors**: down weight older samples

1 Introduction

2 Methods

3 Results

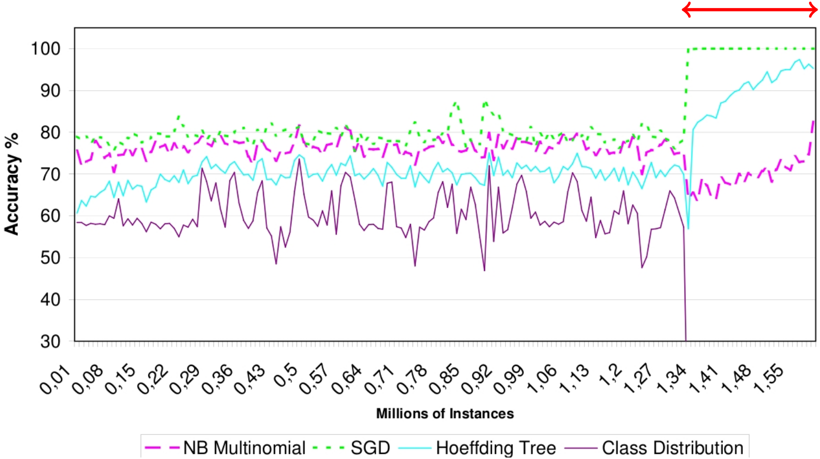
4 Discussion

Training sets

- sentiment140.com (Go et al.)
    - 800.000 positive tweets
    - 800.000 negative tweets
    - Balanced classes - not representative
- Edinburgh corpus
    - 1.800.000 positive tweets
    - 325.000 negative tweets
    - Unbalanced classes - representative

sentiment140.com

Accuracy

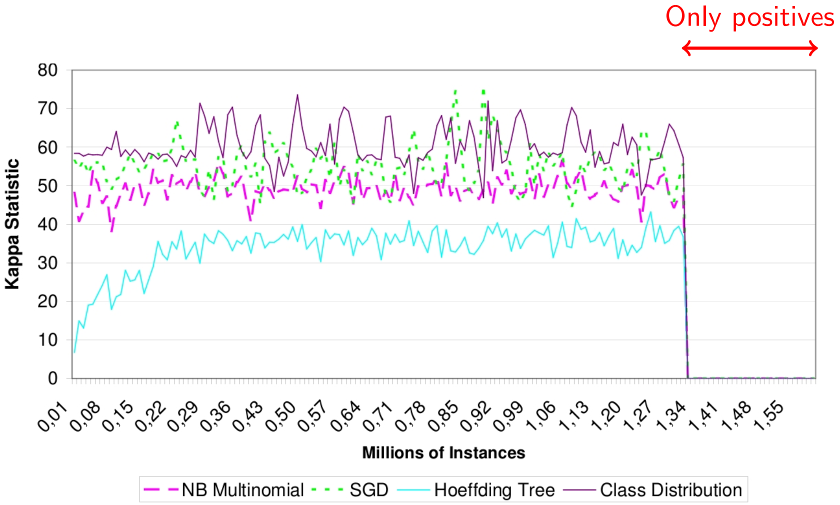


Only positives

⇒ **High accuracy for unbalances classes**

sentiment140.com

κ statistic



$\Rightarrow$ κ **statistic detects unbalanced classes**

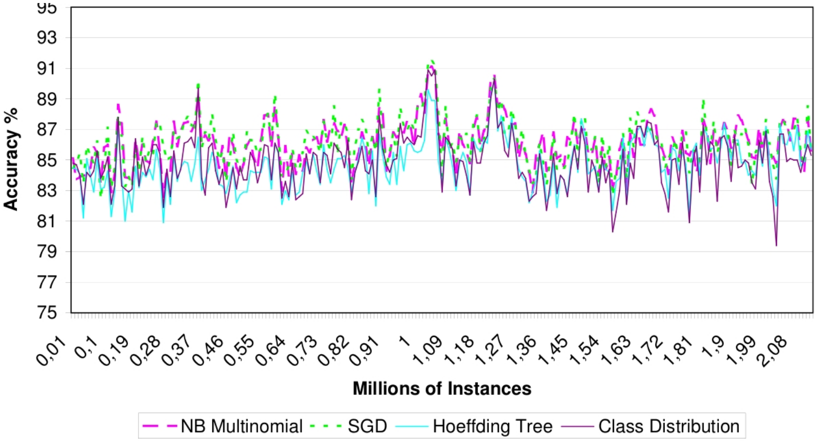
sentiment140.com

		Accuracy	κ	Time
1	SGD	86.80%	62.60%	219.54 sec.
2	Naïve Bayes	75.05%	50.10%	116.62, sec.
3	Hoeffding Tree	73.11%	46.23%	5525.51 sec.

- SGD scores best (Accuracy, κ)
- Hoeffding tree scores worst (Accuracy, κ, Time)

Edinburgh corpus

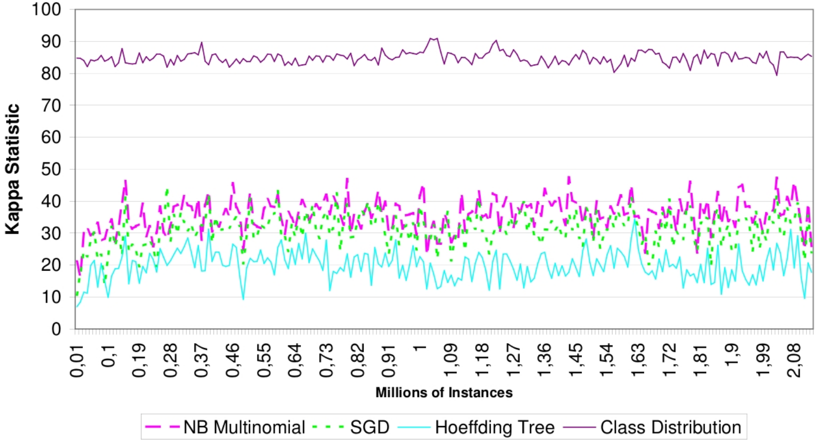
Accuracy



⇒ **Similar accuracy**

Edinburgh corpus

κ statistic



$\Rightarrow$ κ **discriminates better**

Edinburgh corpus

		Accuracy	κ	Time
1	Naïve Bayes	86.11%	36.15%	173.28, sec.
2	SGD	86.26%	31.88%	293.98 sec.
3	Hoeffding Tree	84.76%	20.40%	6151.51 sec.

- Naïve Bayes scores best (κ, Time)
- Hoeffding tree scores worst (Accuracy, κ, Time)
- κ lower than sentiment140.com due to unbalanced classes

Changing feature weights

Tag	Middle	End	Variance
apple	0.3	0.7	0.4
microsoft	-0.4	-0.1	0.3
facebook	-0.3	0.4	0.7
mcdonalds	0.5	0.1	-0.4
google	0.3	0.6	0.3
disney	0.0	0.0	0.0
bmw	0.0	-0.2	-0.2
pepsi	0.1	-0.6	-0.7
dell	0.2	0.0	-0.2
gucci	-0.4	0.6	1.0
amazon	-0.1	-0.4	-0.3

- High weight: more frequent in positive tweets
- Low weight: more frequent in negative tweets

1. Introduction
2. Methods
3. Results
4. **Discussion**

κ statistic

Pros

- Suitable for unbalanced classes
- Simple computation
- Suitable for incremental learning

Cons

- Independence assumption for computing p_c often invalid
- Conservative estimate

Furture improvements

- Additional features
    - Emoticons
    - Number of friends/followers
- Different languages
- Neural sentiments (three classes)
- Semantics
    - Foo beats Bar
    - Bar beats Foo

Sources

A. Bifet and E. Frank (2010)
Sentiment knowledge discovery in twitter streaming data.

A. Go et al. (2009)
Twitter Sentiment Classification using Distant Supervision.

J. Gama et al. (2009)
Issues in evaluation of stream learning algorithms.

J. Cohen (1960)
A coefficient of agreement for nominal scales.