Algorithm Selection for Maximum Common Subgraph

Paulius Dilkas

FATA seminar

16th January 2018

Outline

1 Algorithm selection

2 Labelling

3 Algorithms

4 Random forests

5 Results

6 What happens when labelling changes?

Algorithm selection

Definition (Bischl et al. 2016)

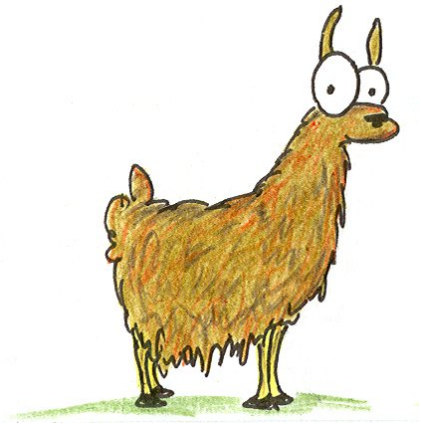
Given a set $\mathcal{I}$ of problem instances, a space of algorithms $\mathcal{A}$, and a performance measure $m\colon \mathcal{I} \times \mathcal{A} \to \mathbb{R}$, the *algorithm selection problem* is to find a mapping $s\colon \mathcal{I} \to \mathcal{A}$ that optimises $\mathbb{E}[m(i, s(i))]$.

Algorithm selection

Definition (Bischl et al. 2016)

Given a set $\mathcal{I}$ of problem instances, a space of algorithms $\mathcal{A}$, and a performance measure $m\colon \mathcal{I} \times \mathcal{A} \to \mathbb{R}$, the *algorithm selection problem* is to find a mapping $s\colon \mathcal{I} \to \mathcal{A}$ that optimises $\mathbb{E}[m(i, s(i))]$.

LLAMA (Kotthoff 2013)



Labelling

Data from Foggia, Sansone and Vento 2001; Santo et al. 2003

Labelling

Data from Foggia, Sansone and Vento 2001; Santo et al. 2003

Definition

A graph $G = (V, E)$ is said to have a $p\%$ *(vertex) labelling* if

$$N = \max\left\{2^n : n \in \mathbb{N},\ 2^n < \left\lfloor \frac{p}{100\%} \times |V| \right\rfloor \right\}.$$

Labelling

Data from Foggia, Sansone and Vento 2001; Santo et al. 2003

Definition

A graph $G = (V, E)$ is said to have a $p\%$ *(vertex) labelling* if

$$N = \max\left\{2^n : n \in \mathbb{N},\ 2^n < \left\lfloor \frac{p}{100\%} \times |V| \right\rfloor \right\}.$$

- 5% labelling - 20 vertices per label (on average)
- 50% labelling - 2 vertices per label (on average)

Labelling

Data from Foggia, Sansone and Vento 2001; Santo et al. 2003

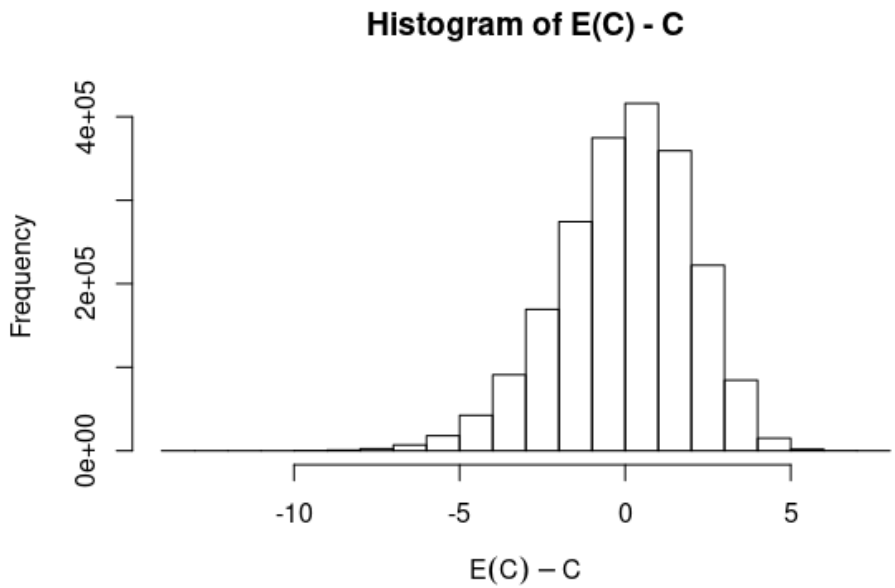
Definition

A graph $G = (V, E)$ is said to have a $p\%$ *(vertex) labelling* if

$$N = \max\left\{2^n : n \in \mathbb{N},\ 2^n < \left\lfloor \frac{p}{100\%} \times |V| \right\rfloor \right\}.$$

- 5% labelling - 20 vertices per label (on average)
- 50% labelling - 2 vertices per label (on average)
- 3 subproblems
  - no labels
  - vertex labels
  - vertex and edge labels

Distribution of vertices per label



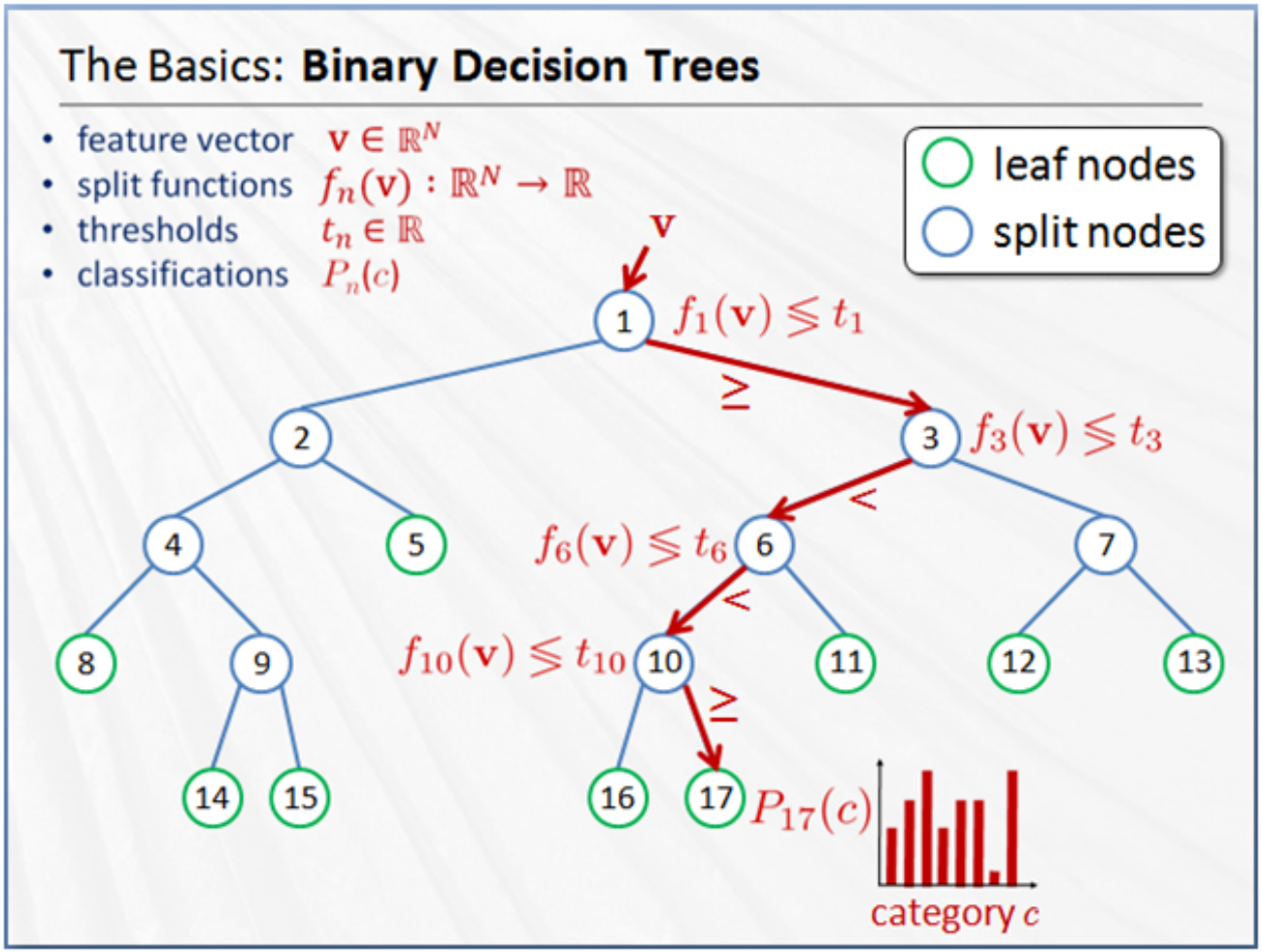
For each graph and label

- C is the number of vertices with that label
- $E(C)$ is the number we would expect from a (discrete) uniform distribution

Algorithms

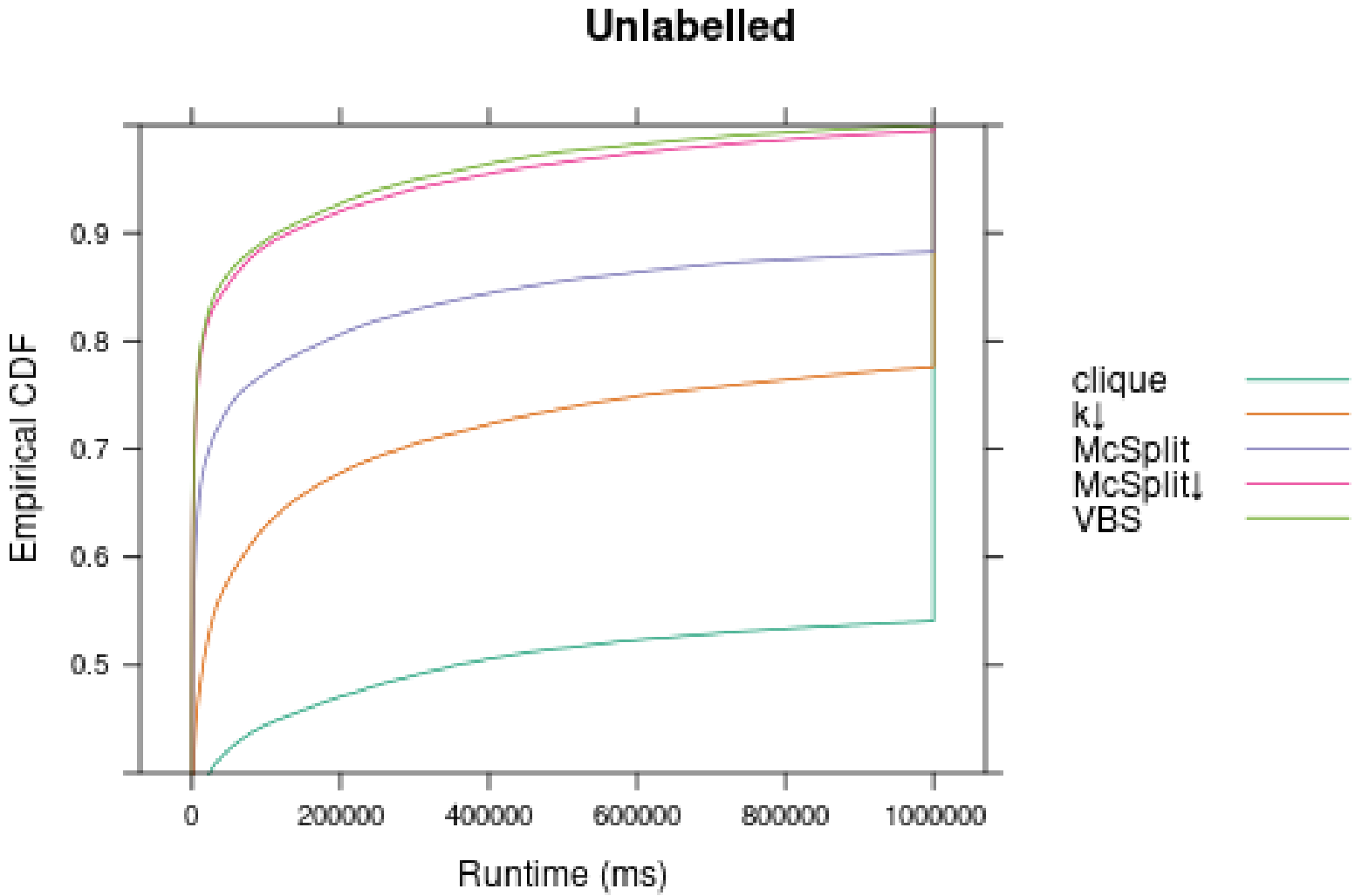
- McSplit, McSplit ↓
  - (McCreesh, Prosser and Trimble 2017)
- clique encoding
  - (McCreesh, Ndiaye et al. 2016)
- $k \downarrow$
  - (Hoffmann, McCreesh and Reilly 2017)

Random forests

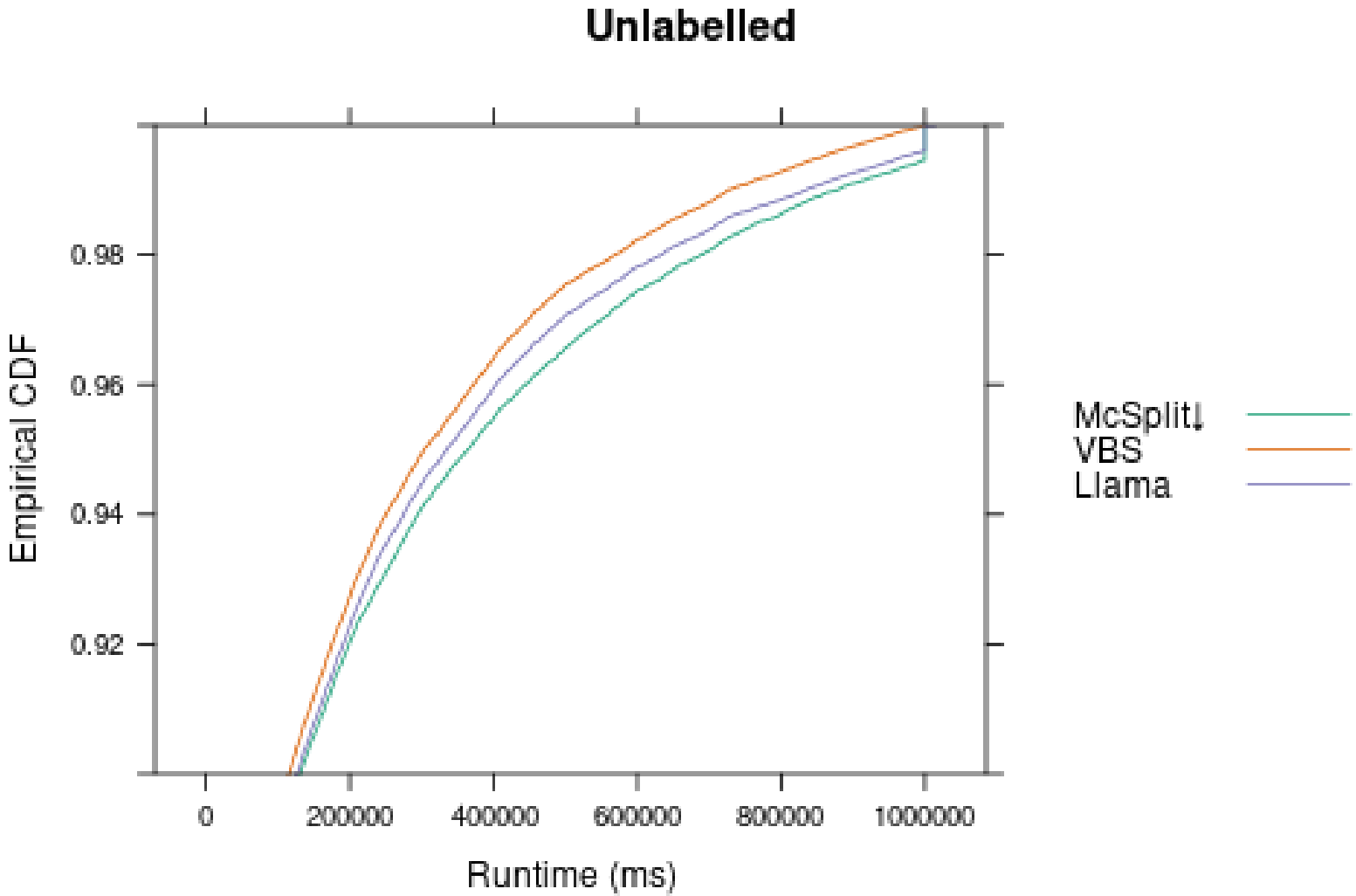


Source: Tae-Kyun Kim & Bjorn Stenger, Intelligent Systems and Networks (ISN) Research Group, Imperial College London

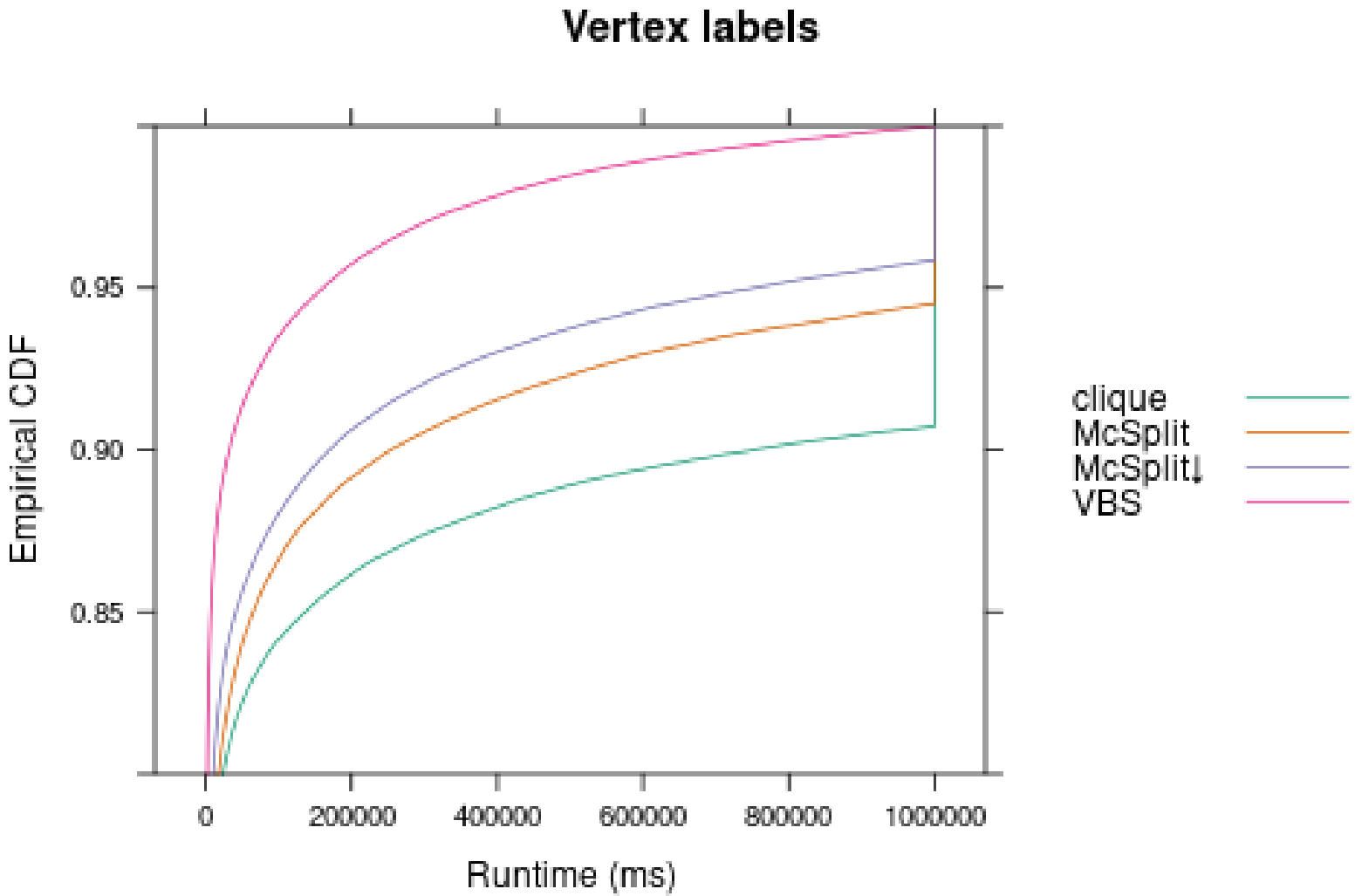
Results



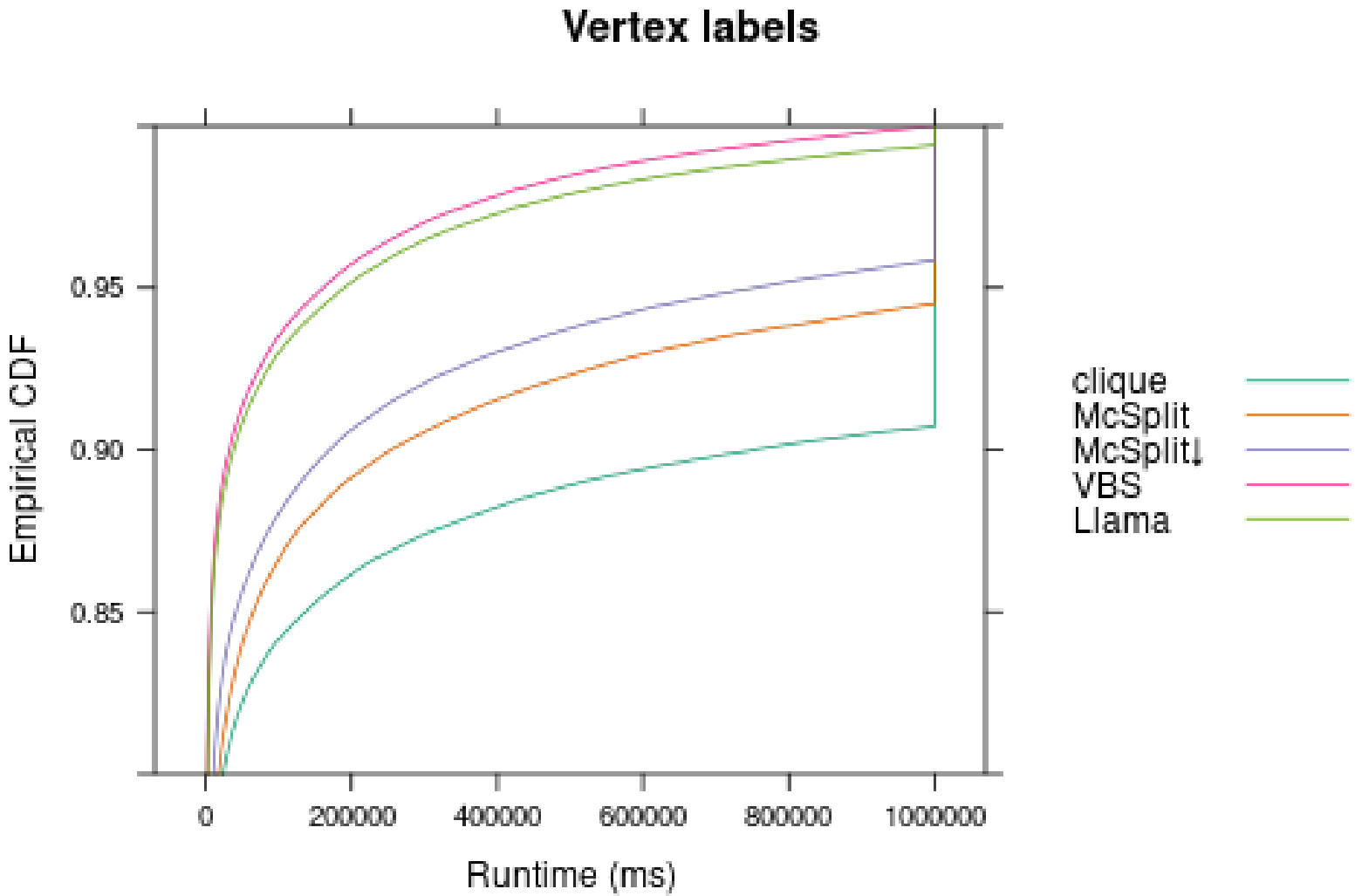
Results (27%)



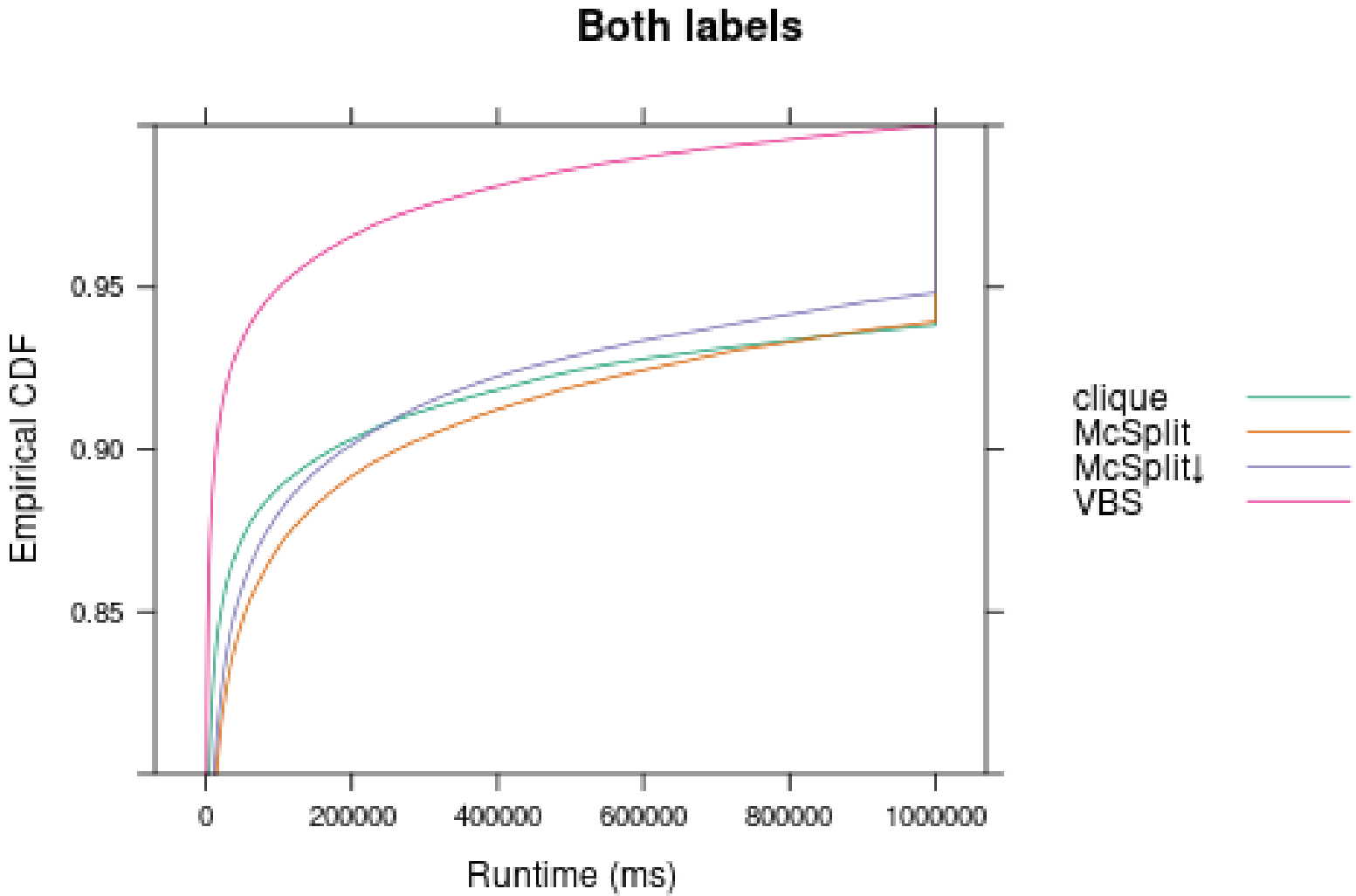
Results



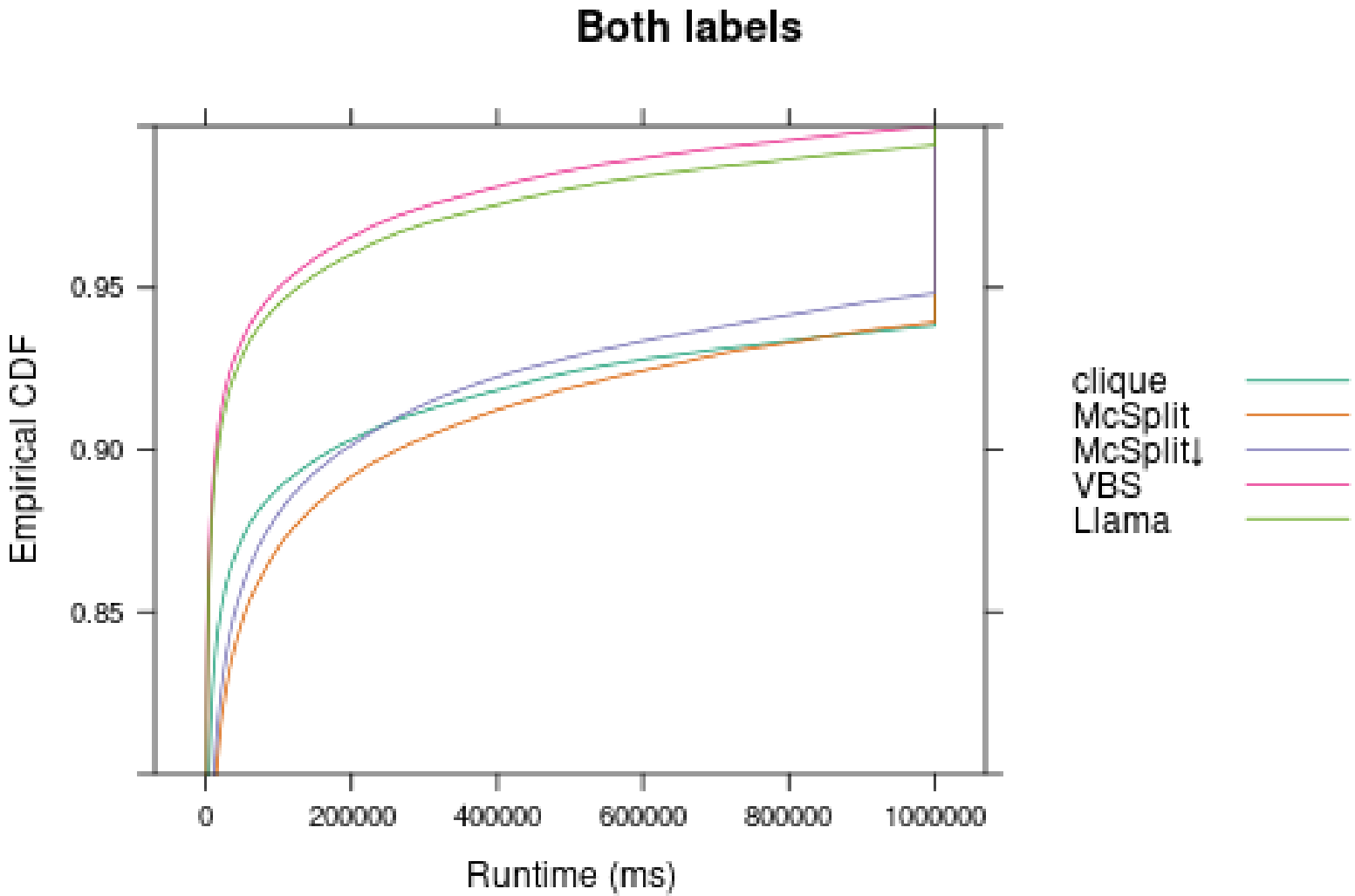
Results (86%)



Results



Results (88%)



Results

- Most important features
  - labelling percentage
  - standard deviation of degrees (for both graphs)

Errors

- Out-of-bag
- (for each algorithm) $1 - \text{recall}$

Definition

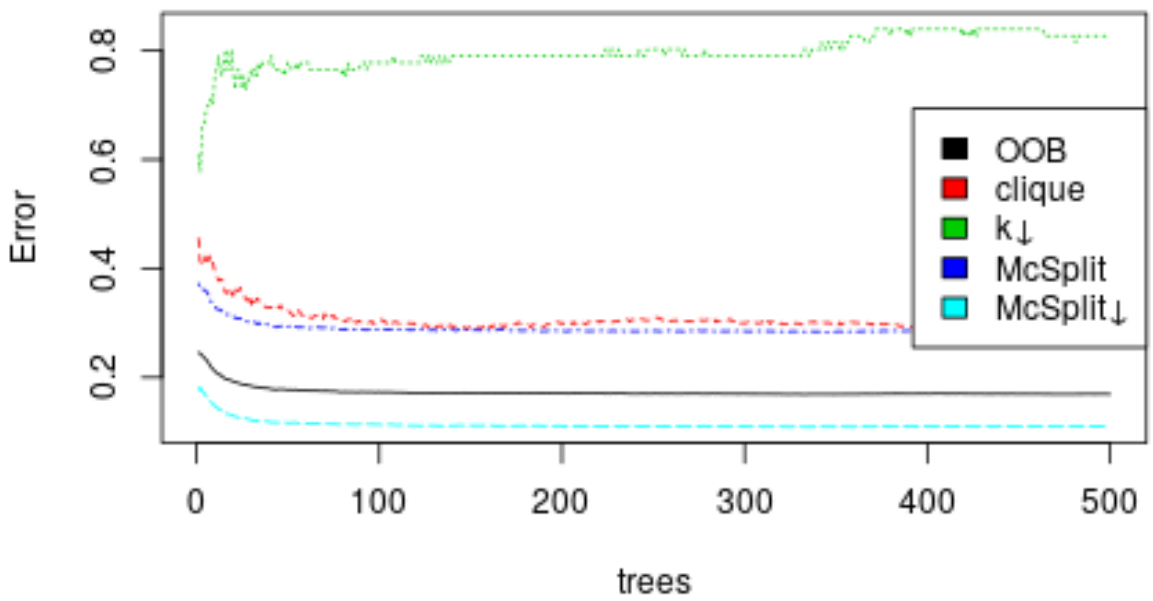
For an algorithm A, *recall* is

$$\frac{\text{the number of instances that were correctly predicted as } A}{\text{the number of instances where } A \text{ is the correct prediction}}.$$

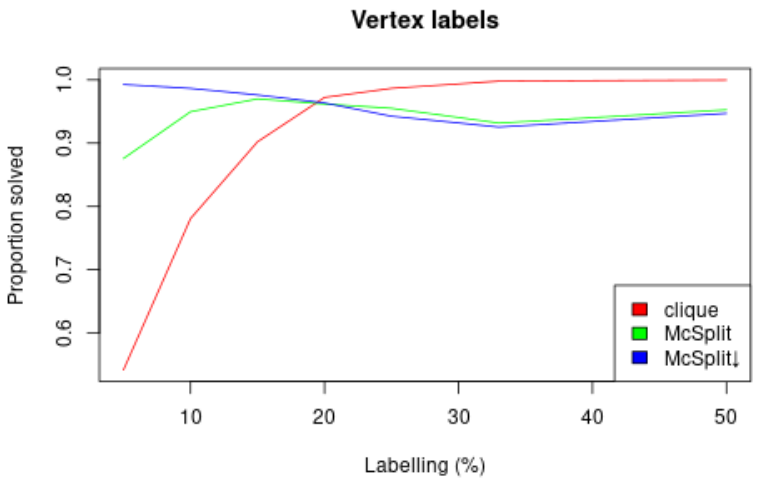
Errors (%)

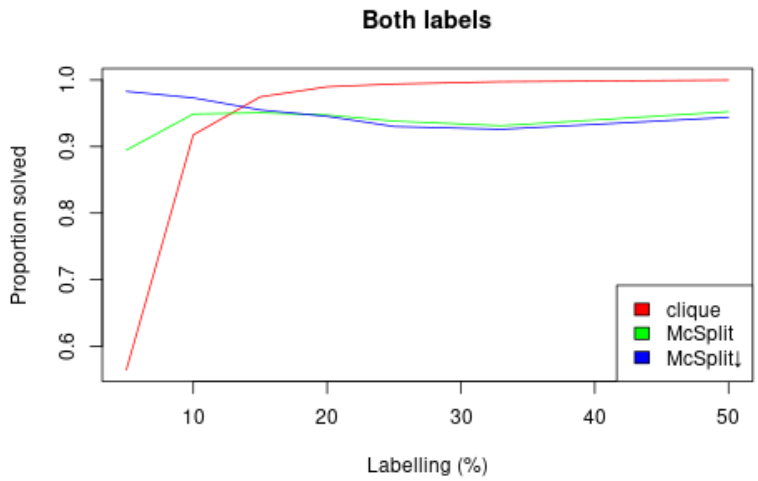
Error	Labelling		
	no	vertex	both
out-of-bag	17	13	14
clique	30	8	7
McSplit	29	22	29
McSplit ↓	11	11	11
$k \downarrow$	80		

Convergence of errors for unlabelled graphs



What happens when labelling changes?





What happens when labelling changes?