# Probability

Jacopo Notarstefano
jacopo.notarstefano [at] gmail.com

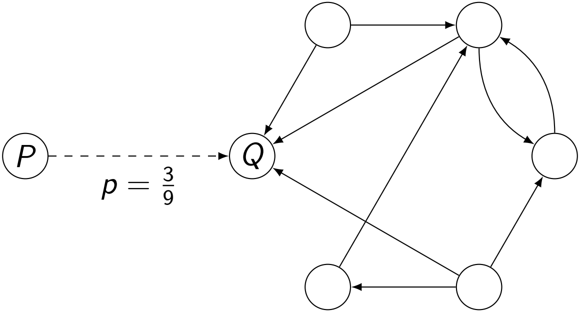
11 February 2014

# Main ideas

1. Given two pages, we want to estimate how many pages would be linked by both if links were created randomly.
2. If the actual number is smaller, then we conclude that those pages are not related. If it's bigger, we assign a score between 0 and 1.
3. This estimate depends on how we model random link creation between pages.

# The Barabási–Albert model

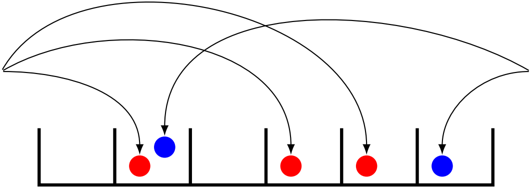
The probability that a new page $P$ links an existing page $Q$ is proportional to $\text{indeg}(Q)$: "The rich get richer".



# Balls and bins, 1/2

## Problem

*Suppose that we have $W$ bins, $n_1$ red balls and $n_2$ blue balls. When we throw a ball it falls in bin $i$ with probability $p_i$. When we are done throwing all the balls, what's the expected number of bins with both a blue and a red ball?*



# Balls and bins, 2/2

**Solution**

*If* **all throws are independent***, then, by linearity of expectation, we have*

$$\mathrm{E}[|N_1 \cap N_2|] = \sum_{i,j=1}^{n_1,n_2} \mathrm{E}[I_{ij}] = n_1 n_2 \sum_{i=1}^{W} p_i^2 = n_1 n_2 \mathbf{P}$$

*where $I_{ij}$ is random indicator variable denoting that red ball $i$ and blue ball $j$ landed in the same bin.*

# The algorithm

**Algorithm 1** Probability

```
// Preprocessing step
Scan Wikipedia and compute P
// For each pair of pages P_1 and P_2
N_1 ← outLinks(P_1); n_1 ← N_1.length
N_2 ← outLinks(P_2); n_2 ← N_2.length
actualValue ← |N_1 ∩ N_2|
expectedValue ← n_1 n_2 · P
if actualValue < expectedValue then
  return 0
else
  return normalize(actualValue − expectedValue)
end if
```

# Results

The algorithm manages to retain the recall baseline while improving on its precision, thus achieving a better F1 score.

| | | | | |
|---|---|---|---|---|
| | TagMe baseline | 0.562 | 0.586 | 0.540 |
|  | group3 | 0.594 | 0.660 | 0.539 |

The algorithm is also fast: a run against the entire AIDA/CoNLL dataset takes less than 2 minutes.