

Escalonamento de Atualizações de Páginas Web Usando Programação Genética

Aécio Solano Rodrigues Santos

Orientador: Nivio Ziviani

Departamento de Ciência da Computação

Universidade Federal de Minas Gerais

11 de Março de 2013

Problema

- ▶ Web consiste em +30 trilhões de páginas

Fonte: <https://www.google.com/insidesearch/howsearchworks/thestory/>

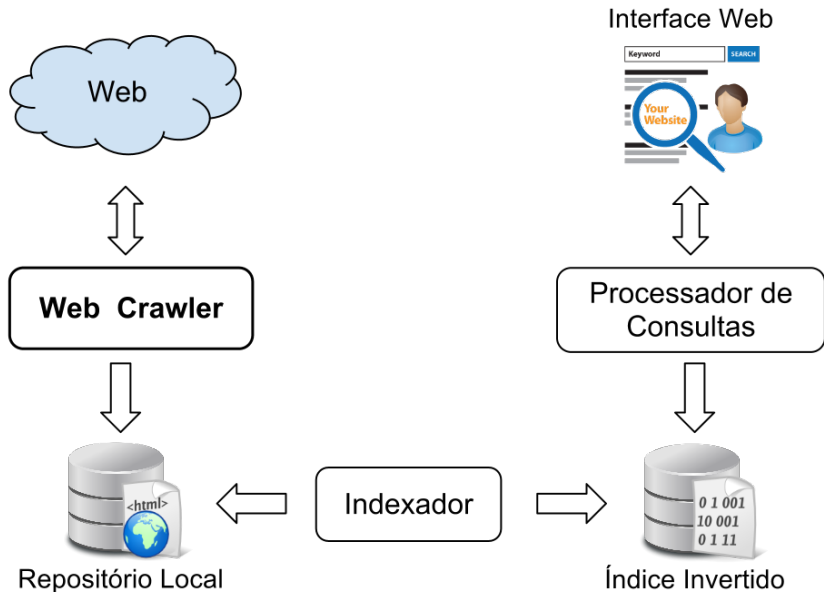
- ▶ Máquinas de busca armazenam cópias da Web
- ▶ Cópias locais devem ser atualizadas
- ▶ Recursos são limitados
- ▶ Coletar páginas que não mudaram é desperdício

Atualizar páginas com maior probabilidade de mudança desde a última coleta

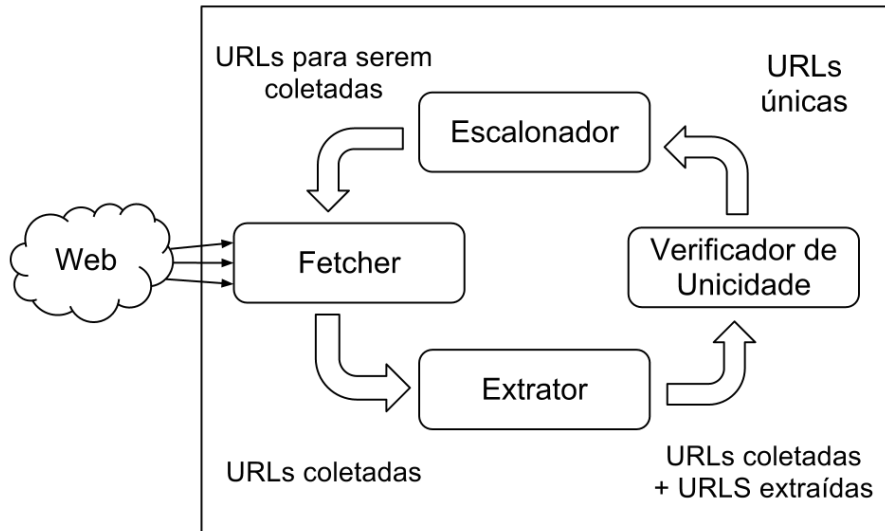
Solução Proposta

- ▶ Arcabouço baseado em aprendizado de máquina
- ▶ Uso de Programação Genética:
 - ▶ Gerar funções que atribuem *scores* para páginas
 - ▶ *Score* de acordo com probabilidade de mudança
- ▶ Aprendizado de máquina já foi usado antes
 - ▶ Classificação em frequência de mudança
 - ▶ Agrupamento por comportamento de mudança
- ▶ Nenhum trabalho usou aprendizado para definir a ordem de coleta diretamente

Máquinas de Busca: Componentes



Web Crawler: Componentes

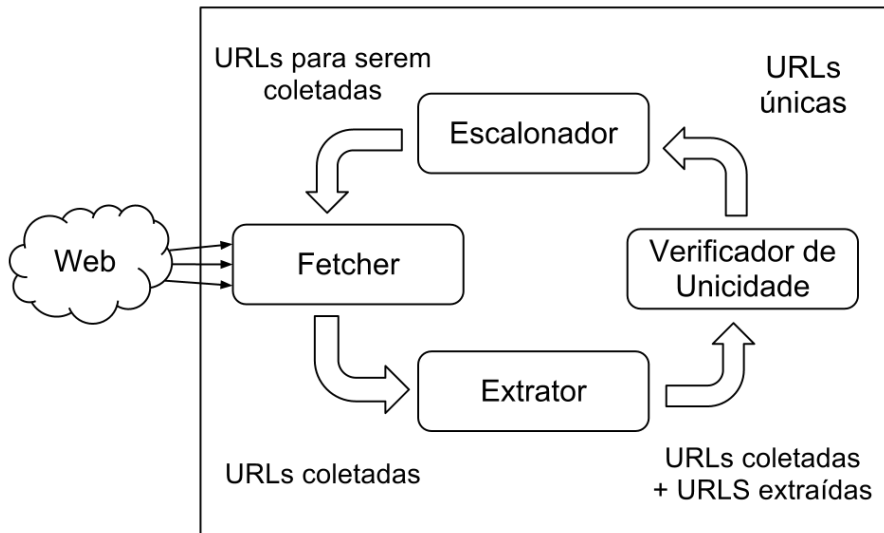


Objetivos e Desafios da Coleta

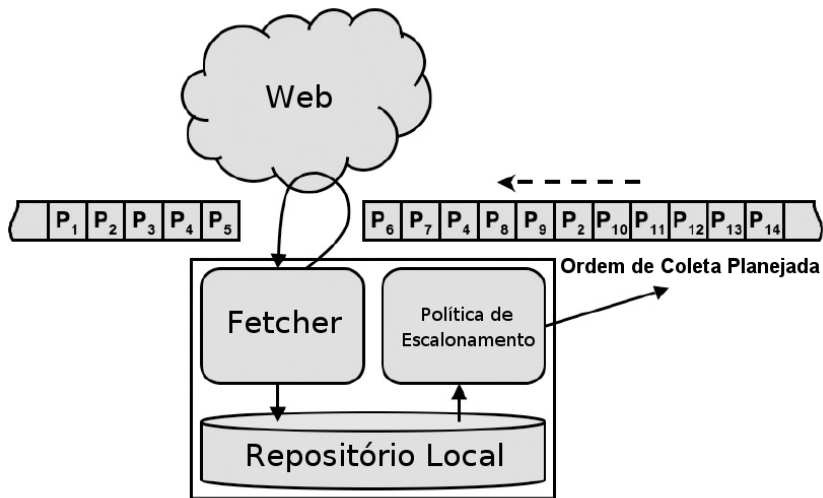
- ▶ *Coverage* – coletar uma grande parte das páginas disponíveis na Web
- ▶ *Freshness* – manter cópias locais atualizadas
- ▶ *Quality* – páginas mais importantes devem ser coletadas mais cedo
- ▶ *User Experience* – evitar mostrar páginas que não existem mais ou são irrelevantes
- ▶ *Discoverability* – descobrir rapidamente páginas novas

Conceitos Básicos & Trabalhos Relacionados

Arquitetura de um Web Crawler



Modelo de Ordenação de Coleta



Fonte: Olston e Najork (2010)

Formas de Coleta

1. Batch Crawling

- ▶ Cada página é coletada somente uma vez
- ▶ Não são necessárias páginas atualizadas

2. Incremental Crawling

- ▶ Páginas são coletadas várias vezes
- ▶ É um processo contínuo
- ▶ Necessidade de páginas atualizadas
- ▶ Utilizado por máquinas de busca comerciais

Batch Crawling

- ▶ O objetivo é coletar páginas mais importantes
- ▶ Weighted Coverage (WC):

$$WC(t) = \sum_{p \in \mathcal{C}(t)} w(p)$$

- ▶ t é o tempo desde início da coleta
 - ▶ $\mathcal{C}(t)$ é o conjunto de páginas coletado até t
 - ▶ $w(p)$ é a importância da página p
- ▶ Políticas: breadth-first, OPIC, larger-sites-first, etc.

Incremental Crawling

A cada ciclo, deve balancear duas ações:

1. Coletar página nova:

1.1 Aumenta cobertura (*Coverage*)

1.2 Provê links novos

- ▶ Melhora estimativa de importância
- ▶ Melhora detecção de spam

2. Atualizar página velha:

2.1 Aumenta *freshness*

2.2 Detecta remoção de páginas

2.3 Provê links novos

- ▶ Mesmos benefícios de 1.2

Weighted Freshness

- ▶ Objetivo é maximizar páginas atualizadas
- ▶ Weighted Freshness (WF):

$$WF(t) = \sum_{p \in \mathcal{C}(t)} w(p) \cdot f(p, t)$$

- ▶ $WF(t)$ – *freshness* do repositório no momento t
- ▶ $\mathcal{C}(t)$ – conjunto de páginas coletado até t
- ▶ $w(p)$ – importância da página p
- ▶ $f(p, t)$ – *freshness* da página p no momento t

Modelos de Freshness

Modelo Binário:

$$f(p, t) = \begin{cases} 1 & \text{se } p \text{ é igual à cópia da Web} \\ 0 & \text{caso contrário.} \end{cases}$$

Modelos Contínuos:

1. Idade

- ▶ Tempo que as cópias divergiram

2. Longevidade de Conteúdo

- ▶ Quantidade de fragmentos atualizados

3. Centrado no usuário

- ▶ Impacto de mudança no resultado de busca

Políticas de Atualização

- ▶ Modelo de Poison para mudança de páginas (Coffman et al., 1998)
- ▶ Estimadores de frequência de mudança (Cho e Garcia-Molina, 2003)
 - ▶ Caso de web crawler: informação incompleta
- ▶ Clustering-based (Tan e Mitra, 2010)
 - ▶ Amostras em nível clusters
 - ▶ Características estáticas e dinâmicas
 - ▶ Probabilidades de mudança adaptativas

Políticas de Atualização

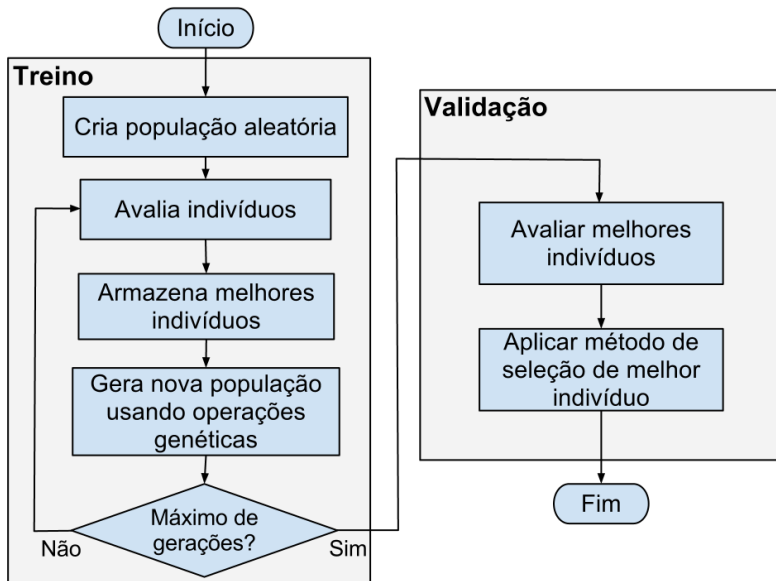
- ▶ *Sampling-based*: amostras em nível de web site (Cho e Ntoulas, 2002)
- ▶ *Embarrassment-level*: minimiza embaraçamento (Wolf et al., 2002)
- ▶ *Classification-based*: primeiros a usar conteúdo (Barbosa et al., 2005)
- ▶ *User-centric*: considera impacto nas buscas (Pandey e Olston, 2005)
- ▶ *Content longevity*: minimiza conteúdo incorreto (Olston e Pandey, 2008)

Programação Genética (GP)

- ▶ Técnica para solução de problemas baseada na teoria da evolução das espécies (Koza, 1992)
 - ▶ Herança biológica
 - ▶ Evolução genética de indivíduos
 - ▶ Mutação
 - ▶ *Cross-over*
- ▶ População é composta de indivíduos
- ▶ Indivíduos são representados em árvores
- ▶ Busca por soluções aproximadas em grandes espaços de soluções candidatas

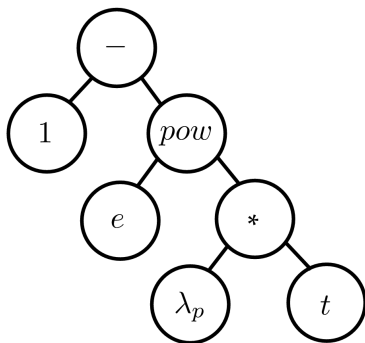
GP2C – Programação Genética para Coleta Incremental

Processo GP2C



Indivíduos

- ▶ Combina informações úteis para estimar de probabilidade de uma página ter mudado
- ▶ Cada indivíduo representa uma função que dá score para cada página



$$1 - e^{\lambda_p t}$$

Terminais e Funções

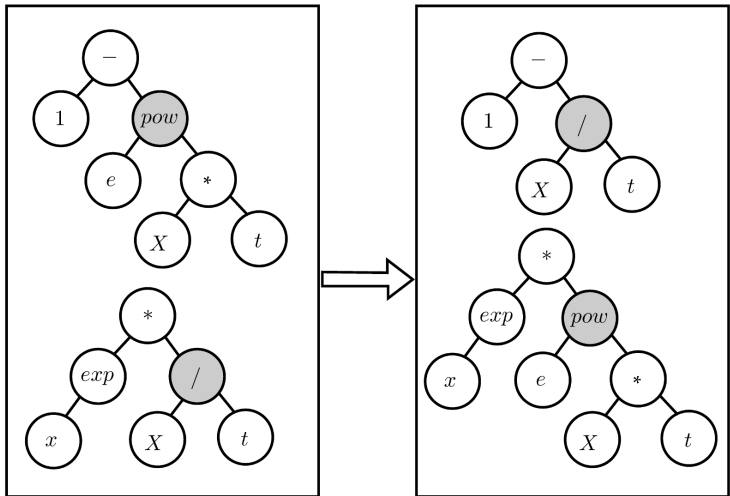
- ▶ Terminais:

n	Número de vezes que a página foi visitada
X	Número de mudanças nas n visitas
t	Número de ciclos desde a última visita

- ▶ Terminais extraídos dos *baselines*
- ▶ Outros terminais podem ser utilizados
- ▶ Funções: $+$, $-$, $/$, $*$, $\log$, pow , exp

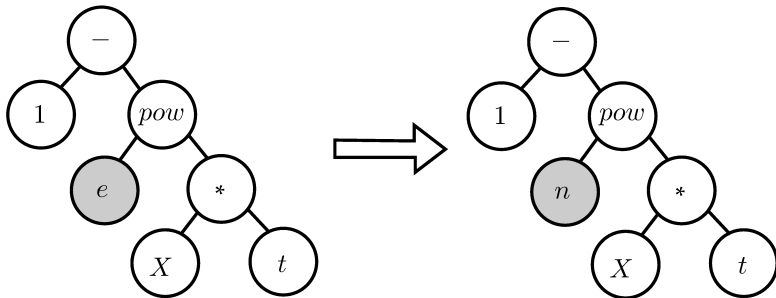
Operações Genéticas: Crossover

- Indivíduos pais têm sub-árvores são trocadas



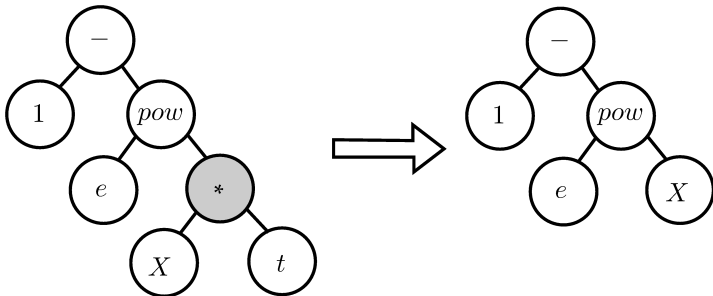
Operações Genéticas: Mutação

- ▶ Mutação *Node replacement*

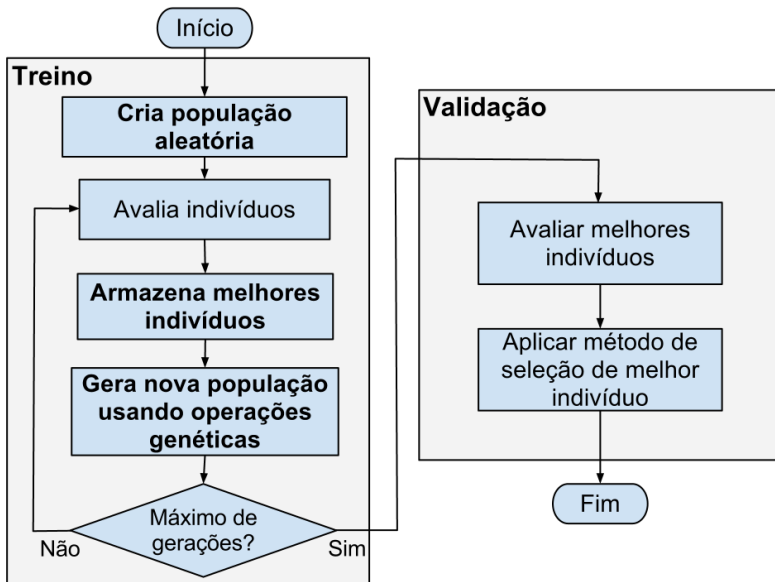


Operações Genéticas: Mutação

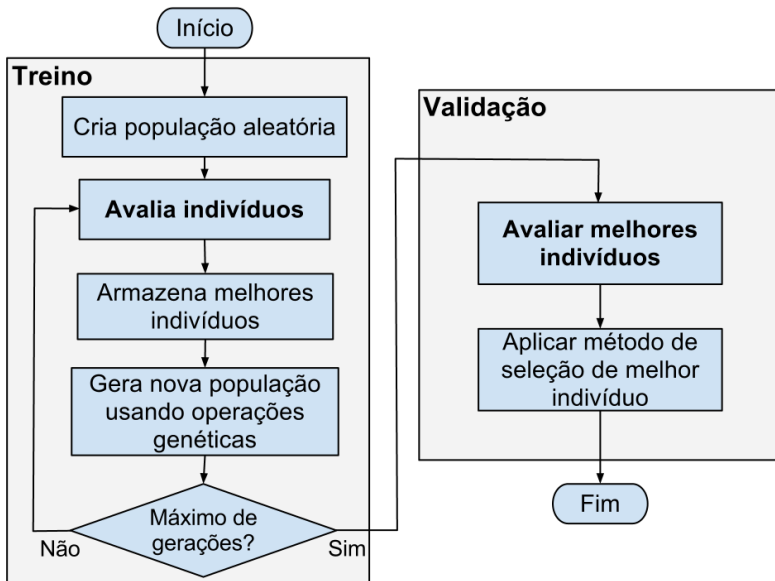
- ▶ Mutação *Shrink*



Processo GP2C



Processo GP2C



Função de Fitness

- ▶ Função de fitness avalia a qualidade da função de escalonamento
- ▶ Adotamos a métrica *Change Ratio*:

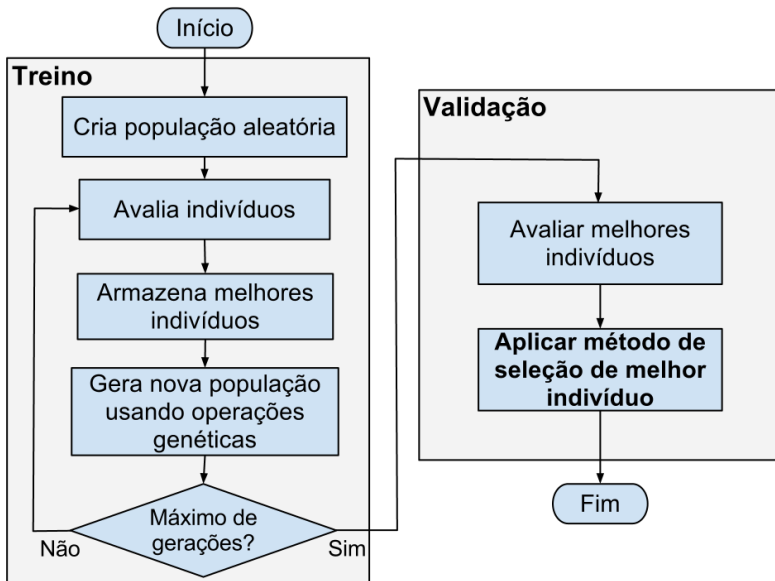
$$C_i = \frac{D_i^c}{D_i}$$

- ▶ D_i – # páginas coletadas no ciclo i
 - ▶ D_i^c – # de páginas coletadas que mudaram
- ▶ Computada realizando *Simulação de Coleta*

Simulação de Coleta

- ▶ Utiliza dados coletados previamente para determinar quando cada página mudou
- ▶ Coleta páginas N_w vezes para obter informações básicas
- ▶ Para cada ciclo disponível:
 1. Computa *score* de cada página usando indivíduo
 2. Seleciona páginas de maior *score*
 3. Usa dados para verificar se página mudou no ciclo
 4. Computa Change Ratio do ciclo
- ▶ Fitness final é a média de todos ciclos

Processo GP2C



Seleção de Melhores Indivíduos

- ▶ $GP2C_{Best}$ – melhor indivíduo conjunto treino
- ▶ $GP2C_{Avg}$ – considera média entre treino e validação, menos desvio padrão
- ▶ $GP2C_{Sum}$ – considera soma entre treino e validação, menos desvio padrão
- ▶ Executado com várias sementes aleatórias para evitar risco de ótimos locais

Avaliação Experimental

Baselines

- ▶ Consideradas melhores funções que exploram mudança de páginas
- ▶ Estimador de frequência de mudança
(Cho e Garcia-Molina, 2003)

$$CG = -\log \left(\frac{n - X + 0.5}{n + 0.5} \right)$$

- ▶ n – quantidade de visitas à página
- ▶ X – quantidade de mudanças nas n visitas

Baselines

- Probabilidade de mudança (Tan e Mitra, 2010)

$$\varphi = Pr\{T \leq t\} = 1 - e^{\lambda_p t}$$

$$\lambda_p = \sum_{i=1}^n w_i \cdot I_i(p)$$

- NAD: $w_1 = w_2 = \dots = w_n = \frac{1}{n}$
- SAD: $w_1 = w_2 = \dots = w_{n-1} = 0, w_n = 1$
- AAD: $w_i = \frac{i}{\sum_{i=1}^n i}$
- GAD: $w_i = \frac{2^{i-1}}{\sum_{i=1}^n 2^{i-1}}$

Dataset

- ▶ BRDC'12 – Brazilian Daily Crawl 2012
- ▶ Coletada usando InWeb Crawler
- ▶ Coleta em largura a partir de 15.000 sites mais populares de acordo com Alexa
- ▶ Seleção de sites usando amostra estratificada
- ▶ Páginas de cada site escolhido aleatoriamente
- ▶ Recoleta diária de Setembro a Novembro/2012
- ▶ Análise de dados e remoção de anomalias
- ▶ <http://homepages.dcc.ufmg.br/~aeciosantos/datasets/brdc12>

Dataset

BRDC'12 Dataset	
Período de monitoramento	57 dias
# de páginas web	417.048
# web sites	7.171
# min páginas web por site	1
# max páginas web por site	2.336
Média de páginas por web site	58,15
% downloads com erros	2,92

Metodologia

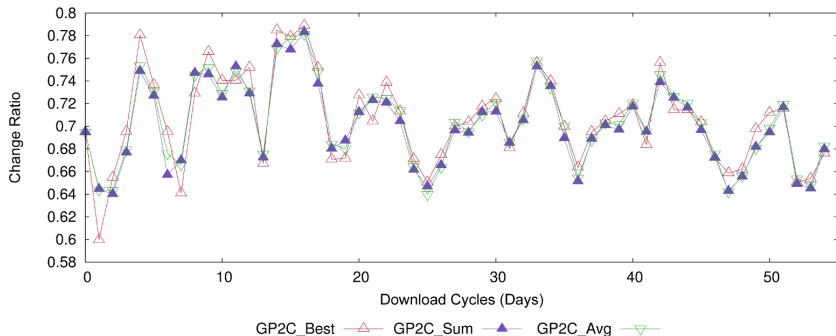
- ▶ Validação cruzada com 5 folds
 - ▶ 4 folds: Divididos em 2 para treino e validação
 - ▶ 1 fold: Teste
- ▶ Métrica de avaliação: Change Ratio
- ▶ Simulação de coleta
- ▶ k igual 5% de páginas do repositório
- ▶ Ciclos de “*warm-up*” $N_w = 2$
- ▶ Teste t pareado com 95% de confiança
- ▶ Teste em cada ciclo para cada par de método

Parâmetros de GP

- ▶ Seleção de valores por “força bruta”

Parâmetro	Valor
Indivíduos	300
Gerações	50
Seleção de indivíduos	Torneio de tamanho 2
Crossover	90%
Reprodução	15%
Mutações	5%
Altura máxima de árvore	10
Melhores indivíduos	50
Sementes aleatórias	5

$GP2C_{Best} \times GP2C_{Avg} \times GP2C_{Sum}$



- ▶ Empate estatístico na maioria dos dias
- ▶ $GP2C_{Best}$ melhor que $GP2C_{Avg}$ em 1 dias
- ▶ $GP2C_{Best}$ melhor que $GP2C_{Sum}$ em 2 dias

Change Ratio das melhores funções $GP2C_{Best}$

Fold	Train set	Validation set	Test set
1	0,7093	0,7080	0,7035
2	0,7049	0,6981	0,7068
3	0,7180	0,7158	0,7118
4	0,7066	0,7088	0,7130
5	0,6960	0,6940	0,6937

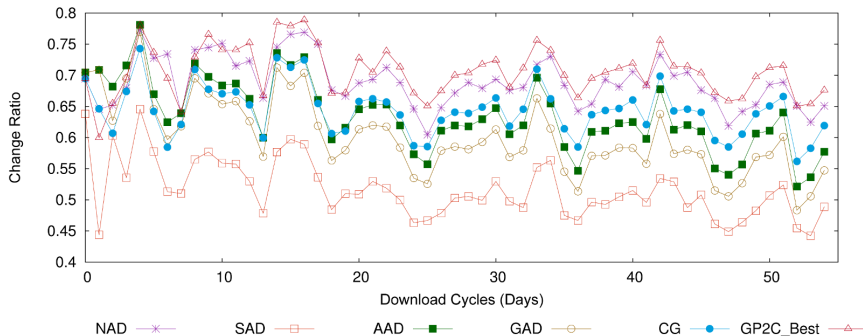
Exemplo de Função Gerada

- ▶ GP2C pode ser usado para entender melhor as *features*
- ▶ Melhores funções geradas seguem um padrão:
 - ▶ Os terminais X e t aparecem frequentemente
 - ▶ n não impacta muito o resultado
- ▶ Exemplo de função simples gerada:

$$X * t$$

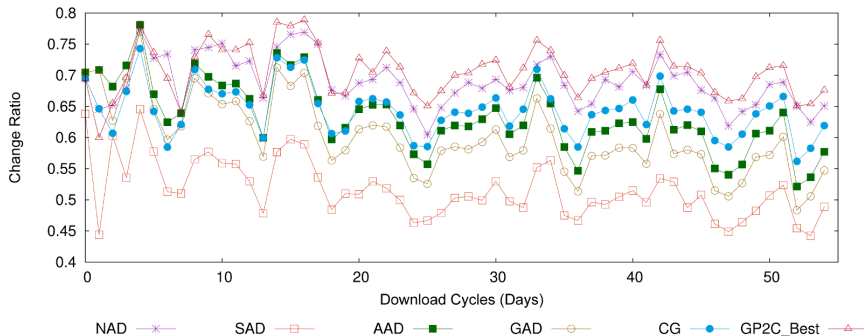
- ▶ São geradas funções que se adaptam aos dados

$GP2C_{Best} \times Baselines$



- ▶ $GP2C_{Best}$ é superior na maioria dos ciclos
- ▶ Empata em alguns ciclos
- ▶ NAD, AAD, GAD, e CG são os mais competitivos

GP2C_{Best} × *Baselines*



	NAD	AAD	GAD	CG
Superior	22	27	49	50
Inferior	0	2	1	0

Change Ratio Médio

Política	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Média
Rand	0.1853	0.1866	0.1855	0.1855	0.1853	0.1856
Age	0.2121	0.2139	0.2127	0.2135	0.2129	0.2130
NAD	0.6837	0.6936	0.6865	0.6941	0.6880	0.6892
SAD	0.5094	0.5189	0.5151	0.5237	0.5156	0.5165
AAD	0.6272	0.6425	0.6280	0.6424	0.6314	0.6343
GAD	0.5974	0.6074	0.5967	0.6057	0.6007	0.6016
CG	0.6399	0.6502	0.6377	0.6481	0.6435	0.6439
$GP2C_{Best}$	0.7035	0.7068	0.7118	0.7130	0.6937	0.7058
$GP2C_{Sum}$	0.7033	0.7052	0.7114	0.7079	0.6760	0.7007
$GP2C_{Avg}$	0.7033	0.7052	0.7114	0.7079	0.6888	0.7033

Uso em Coletor de Larga Escala

- ▶ O Processo GP2C deve ser executado off-line
- ▶ A função aprendida é usada on-line, em conjunto com uma ordenação para pegar as top- k páginas
- ▶ Não é viável realizar acessos aleatórios para atualizar dados das páginas
- ▶ Atualização pode ser realizada em *batches* junto com verificação de unicidade:
 - ▶ Acumular páginas coletadas em um *buffer*
 - ▶ Atualizar em uma única passada no repositório
 - ▶ Quando o crawler percebe que uma página já existe, os dados são atualizados

Custo Computacional

- ▶ On-line:
 - ▶ Custo de computar score: $O(|\mathcal{R}|)$
 - ▶ Ordenação: $O(|\mathcal{R}| \times \log |\mathcal{R}|)$
- ▶ Off-line:
 - ▶ Tarefa mais cara é a simulação de coleta:

$$O(N_o \times |\mathcal{R}| \times \log |\mathcal{R}|)$$

- ▶ Quantidade de simulação de coletas realizadas:

$$O(\mathcal{N}_g \times |\mathcal{P}| + |\mathcal{B}_t|)$$

Desempenho

- ▶ Tempo (s) de 5 folds (5 treinos + 5 validações)
- ▶ Coeficiente de variação entre seeds é 1.5%

	Seeds						
Fase	111	222	333	444	555	Total	%
T	43213	44321	44452	43095	44236	219316	99.624%
V	161	149	196	142	171	819	0.372%
T+V	43374	44470	44647	43237	44407	220136	99.996%
Total	43375	44472	44649	43239	44409	220144	100.00%

Conclusões e Trabalhos Futuros

- ▶ Foi proposto um arcabouço baseado em PG para escalonamento de atualização de páginas
- ▶ GP2C foi capaz de gerar funções tão boas ou melhores que baselines
- ▶ Caso novas características sejam fornecidas, novas funções podem ser geradas facilmente
- ▶ Pode ser adaptado para considerar:
 - ▶ Probabilidade de página ser exibida em buscas
 - ▶ Clique de usuários
 - ▶ Chances de encontrar páginas novas

Conclusões e Trabalhos Futuros

- ▶ Estudar o uso do arcabouço para gerar funções para coletar páginas importantes mais cedo
- ▶ Balancear objetivos de coleta:
Coverage $\times$ Freshness
- ▶ Utilização de técnicas de GP multi-objetivo

Contribuições

- ▶ Proposta do GP2C
 - ▶ Primeiro a usar GP
 - ▶ Oportunidades de trabalhos futuros
 - ▶ Solução flexível e viável
- ▶ Aplicação do GP2C em atualização de páginas
- ▶ Avaliação experimental do arcabouço
- ▶ Integração em um web crawler de larga escala
- ▶ Artigo submetido:
Learning to Schedule Web Page Updates Using Genetic Programming. SIGIR, 2013.

Obrigado

Escalonamento de Atualizações de Páginas Web Usando Programação Genética

Aécio Solano Rodrigues Santos
Orientador: Nivio Ziviani

Departamento de Ciência da Computação
Universidade Federal de Minas Gerais

11 de Março de 2013