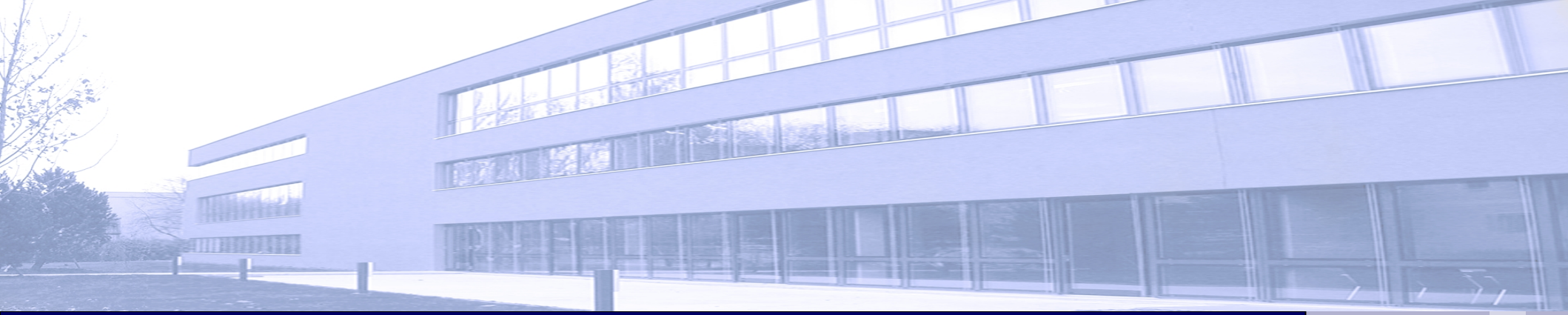


Automatic Cluster Number Selection using a Split and Merge K-Means Approach

Markus Muhr and Michael Granitzer

31st August 2009

Agenda

- Introduction
- Contribution
- Split & Merge Spherical K-Means
- Internal Validity Indices
- Experimental Setup
- Results
- Discussion
- Conclusion

Introduction

- Text is high-dimensional, very sparse, large-scale
- Partitional methods especially Spherical K-Means
  - Bag-of-word based Vector Space Model
  - Cosine similarity
  - Unit sphere
- Cluster number must be known a-priori!
- Dynamic K-Means → X-Means
  - Bisecting K-Means
  - Bayesian Information Criterion
  - Assess quality of bisecting splits

Contribution

- Extending X-Means
  - Merge steps (reduce error propagation)
  - Online updates
  - Spherical optimization criteria
- Evaluation
  - Standard text data sets (Cluto)
  - Several different internal validity indices
  - Batch and Online Updates
- Adapt internal validity indices

Spherical K-Means Basics

- Maximize cosine similarity as cost function
  - Unit-length normalized centroids and data samples
  - Calculate inner product between
  - Data samples and their nearest cluster centroid

$$L = \sum_{i=1}^{N} x_i^T c_{y_i}$$

Batch Spherical K-Means

- Separate update and assignment step
- Samples are assigned to nearest centroid

$$\textbf{for } n = 1 \text{ to } N \textbf{ do}$$
$$y_n = \arg\max_{1 \leq k \leq K} x_n^T c_k$$

- Normalized sum of assigned samples

$$\textbf{for } k = 1 \text{ to } K \textbf{ do}$$
$$c_k = \sum_{x_i \in \mathcal{X}_k} x_i \text{ where } \mathcal{X}_k = \{x_n | y_n = k\}$$
$$c_k = \frac{c_k}{\|c_k\|}$$

Online Spherical K-Means

- Update step of winning cluster after assignment
- Competitive learning technique
- Winner-Take-All approach

$$\textbf{for all } n = 1 \text{ to } N \textbf{ do}$$
$$y_n = \arg\max_{1 \leq k \leq K} x_n^T c_k$$
$$c_{y_n} = \frac{c_{y_n} + \eta(x_n - c_{y_n})}{||c_{y_n} + \eta(x_n - c_{y_n})||}$$

Split & Merge Spherical K-Means

- Initial K-Means partitioning with given k (e.g. 2)
- Consecutive split and merge steps
- Criteria to select cluster candidates
  - → largest cluster (radius or size) for split step
  - → nearest centroid pair for merge step
- Assess quality of cluster result by internal validity index
  - Bayesian Information Criterion, etc.
- No quality improvements → Terminate
- Optional K-Means step for refinement

Split & Merge Spherical K-Means

Algorithm 3 Split and Merge K-Means

Require: $\mathcal{X}$,K, $s(\mathcal{C})$, $m(\mathcal{C})$, $v(\mathcal{C})$
Ensure: $\mathcal{C}$, $\mathcal{Y}$

1: $\mathcal{C}$ = k-means $(\mathcal{X}_t, K)$
2: **repeat**
3: $c_s = s(\mathcal{C})$, $\mathcal{X}_s = \{x_n | y_n = s\}$
4: $\{c_i, c_j\}$ = k-means $(\mathcal{X}_s, K = 2)$
5: **if** $v(\mathcal{C}) > v(\mathcal{C}/c_s \cup \{c_i, c_j\})$ **then**
6: $\mathcal{C} = \mathcal{C}/c_s \cup \{c_i, c_j\}$
7: **until** $|\mathcal{C}|$ is not changing
8: **repeat**
9: $c_i, c_j = m(\mathcal{C})$
10: $Y_j = Y_i$, $\mathcal{C} = \mathcal{C}/c_j$
11: **if** $v(\mathcal{C}) > v_(\mathcal{C}/c_j)$ **then**
12: $\mathcal{C} = \mathcal{C}/c_j$
13: **until** $|\mathcal{C}|$ is not changing
14: $\mathcal{C}$ = k-means $(\mathcal{X}_t, \mathcal{C})$

Internal Validity Indices

- Bayesian Information Criterion (X-Means)

$$-\frac{n_i}{2}\log 2\pi - \frac{n_i m}{2}\log\sigma^2 - \frac{n_i - k}{2} + n_i \log\frac{n_i}{n} - \frac{k}{2}\log n$$

- Bayesian Information Criterion (heuristic)

$$BIC_h = -\frac{n_i m}{2}\log\sigma^2 - \frac{k}{2}\log n$$

- Calinski-Harabasz Index

$$CH_k = \frac{tr(B_k)/(k-1)}{tr(W_k)/(n-k)}$$

- Hartigan Index

$$H_k = \left(\frac{tr(W_k)}{tr(W_{k+1})} - 1\right)(n-k-1)$$

- Krzanowski-Lai Index

$$diff_k = (k-1)^{2/m} tr(W_{k-1}) - k^{2/m} tr(W_k)$$

$$KL_k = |diff_k| / |diff_{k+1}|$$

Experimental Setup

- Text data sets from CLUTO Toolkit
- 12 different data sets
- 878 – 4069 documents with 6 to 25 classes
- Stop words removed, stemmed, tf-idf
- Cluster performance using f-score

$$F(L_r, S_i) = \frac{2 * R(L_r, S_i) * P(L_r, S_i)}{(R(L_r, S_i) + P(L_r, S_i))}$$

$$FScore = \sum_{r=1}^{C} \frac{n_r}{n} \text{argmax}_{S_i \in T} F(L_r, S_i)$$

Experimental Setup

Name	Source	$\lvert\mathcal{X}\rvert$	# Classes
hit	S. J. M. (TREC)	2301	6
rev	S. J. M. (TREC)	4069	5
la1	LA Times (TREC)	3204	6
la2	LA Times (TREC)	3075	6
tr31	TREC	927	7
k1b	WebACE	2340	6
tr41	TREC	878	10
re0	Reuters-21578	1504	13
fbis	FBIS (TREC)	2463	17
k1a	WebACE	2340	20
wap	WebACe	1560	20
re1	Reuters-21578	1657	25

Experimental Setup

- Batch and Online Split & Merge K-Means
- 12 data sets and 5 validity indices
- Different initial partitioning
  - → 5, 15 or 35 clusters
  - → 2, 8 or 15 clusters
- Seeding mechanism using K-Means++
- Square root annealing factor with initial at 0.2
- 10 runs for each setting
- Mean, std. dev., maximum of f-score, cluster number

Results Batch Updates

data	ind.	f_μ (k_μ)	f_σ (k_σ)	f_m (k_m)
hit	KL	0.52 (5.4)	0.05 (0.97)	0.6 (5)
rev	BIC_h	0.72 (5.6)	0.05 (1.26)	0.77 (5)
la1	BIC_h	0.68 (5.9)	0.08 (1.79)	0.79 (5)
la2	BIC_h	0.71 (10)	0.06 (1.25)	0.76 (10)
tr31	CH	0.78 (7.9)	0.06 (0.74)	0.87 (7)
k1b	KL	0.78 (6)	0.09 (0.82)	0.91 (5)
tr41	H	0.66 (9.9)	0.04 (2.42)	0.74 (10)
re0	CH	0.51 (12.2)	0.02 (0.42)	0.53 (12)
fbis	CH	0.63 (14.1)	0.04 (1.2)	0.68 (14)
k1a	$Hart$	0.56 (16.1)	0.04 (1.91)	0.62 (13)
wap	KL	0.55 (18.5)	0.03 (0.71)	0.61 (18)
re1	KL	0.52 (26.9)	0.04 (2.13)	0.57 (26)

Results Online Updates

data	ind.	f_μ (k_μ)	f_σ (k_σ)	f_m (k_m)
hit	BIC_h	0.55 (5.4)	0.02 (0.84)	0.59 (6)
rev	KL	0.73 (5.6)	0.06 (0.84)	0.78 (5)
la1	BIC_h	0.72 (5.6)	0.06 (0.7)	0.8 (6)
la2	CH	0.75 (5.4)	0.04 (0.52)	0.79 (5)
tr31	CH	0.82 (7.5)	0.06 (1.18)	0.92 (7)
k1b	CH	0.85 (5.4)	0.04 (0.52)	0.93 (6)
tr41	BIC	0.7 (13.1)	0.06 (0.99)	0.84 (12)
re0	BIC_h	0.52 (10.5)	0.05 (4.35)	0.57 (8)
fbis	CH	0.63 (15.9)	0.03 (2.33)	0.68 (17)
k1a	BIC	0.57 (16.7)	0.04 (2)	0.64 (19)
wap	KL	0.56 (18.8)	0.04 (0.92)	0.62 (20)
re1	H	0.55 (30.9)	0.02 (1.29)	0.58 (29)

Discussion F-Score

- Dynamic vs. Static cluster number selection*
  - Similar f-scores in most cases
  - Better results for few class problems (tr31, k1b)
  - Worse results for many class problems (re0, re1)
- Online vs. Batch Updates
  - Mean higher with online updates
  - Standard deviation mostly lower with online updates
  - Maximum value can be reached with both
  - Online updates are better and more robust

*Y. Zhao and G. Karypis, “Evaluation of hierarchical clustering algorithms for document datasets”

Discussion Cluster Number

- Cluster number close to the given class number
  - Few class problem deviation is around 1
  - e.g. for data set k1b mean of cluster number is 5.4
  - Many class problems deviation increases
- Better results for good separable data sets (k1b, tr31)
- Standard deviation of cluster number is low → robust
- F-Scores are always comparable with best static results
- Selected cluster number → optimizes cost function

Discussion Validity Indices

- No clear winner – quite equally good results
- Bayesian Information Criterion seems to be best
  - ➔ X-Means and heuristic version provide similar results
  - ➔ Slight favor for heuristic version
- Calinski-Harabasz index performs good as well
- Hartigan, Krzanowski-Lai index less suitable
- All indices approximate the cluster number quite well!

Discussion Dynamic Updates

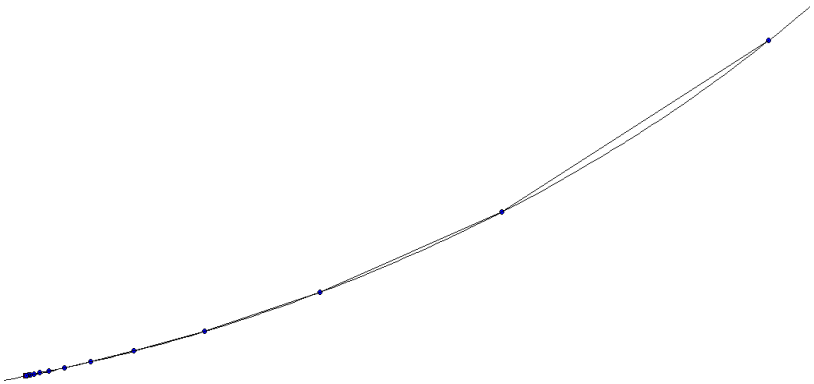
- Initial partitioning does not alter results considerably
  - Fewer initial clusters (2, 5) → slight underestimation
  - Many initial clusters (15, 35) → slight overestimation
  - Almost accurate initial clusters (5, 15) – correct estimation
- Suitable for incremental work flow
- Good separable problems → always correct number
- Bad separable problems → Under- / Overestimations
- Many classes problem → Underestimations

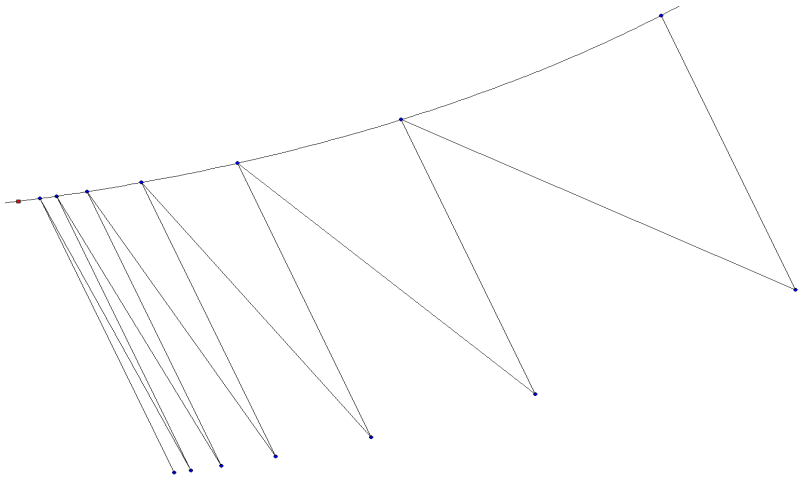
Discussion Runtime

- Split & Merge K-Means
  - Faster than X-Means → less splits and cheap merges
  - Validity indices can be calculated fast
  - Runtimes comparable with bisecting K-Means
- Online vs. batch updates
  - Online updates converge faster, but take longer
  - Batch → sum of very sparse documents
  - Online → updates all non-zero dimensions of the centroid (rather dense) in each update step
  - Online updates 10 times slower than batch updates

Discussion Runtime Outlook

- Heuristic approach
  - ➔ Weighted sum of documents in update step $c_i = c_i + \eta x_n$
  - ➔ Instead of geometrical correct update $c_i = c_i + \eta(x_n - c_i)$
- l2-norm exceeds threshold → normalize centroids
- Increases convergence time, decreases update time





Conclusion

- Split & Merge K-Means guesses clusters quite good
- Appropriate for incremental work flow
- Online updates better than batch updates
- In real-world applications use heuristic online update
- Type of validity index of lesser concern
- Bayesian Information Criterion provides best results

Thank you for your attention!
Questions?