Randomized Dimensionality Reduction for k-Means Clustering

—

Andrew Ma
Chris Tang

Randomized Dimensionality Reduction for k-means Clustering

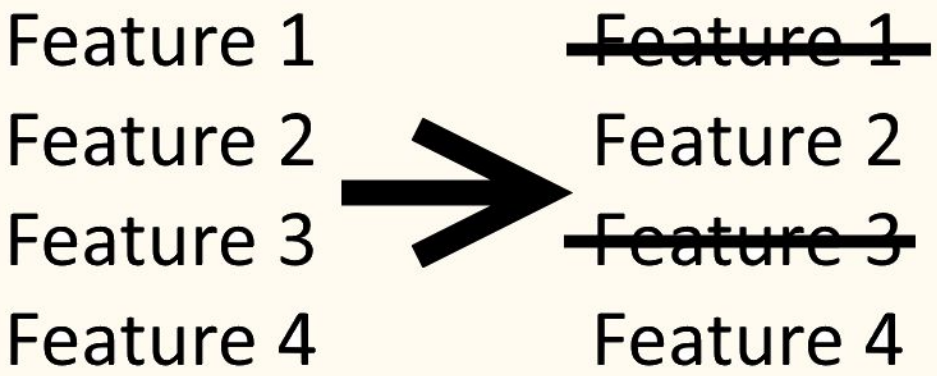
- Authors —
  - Christos Boutsidis
  - Anastasios Zouzias
  - Michael W. Mahoney
  - Petros Drineas
- Date —
  - 13 October 2011 (4 November 2014)
- Venue of publication —
  - IEEE Transactions on Information Theory

Problem & Motivation

- Dimensionality Reduction
  - Reduce the number of features (random variables) considered
  - Union of two approaches —
    - Feature selection
    - Feature extraction
- Motivation
  - Reduce space and time complexity

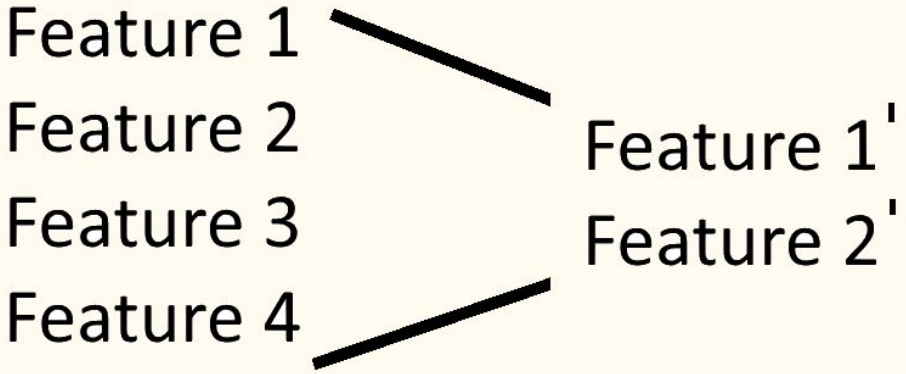
Feature Selection

- Reduces R^p to R^d
- $d << p$



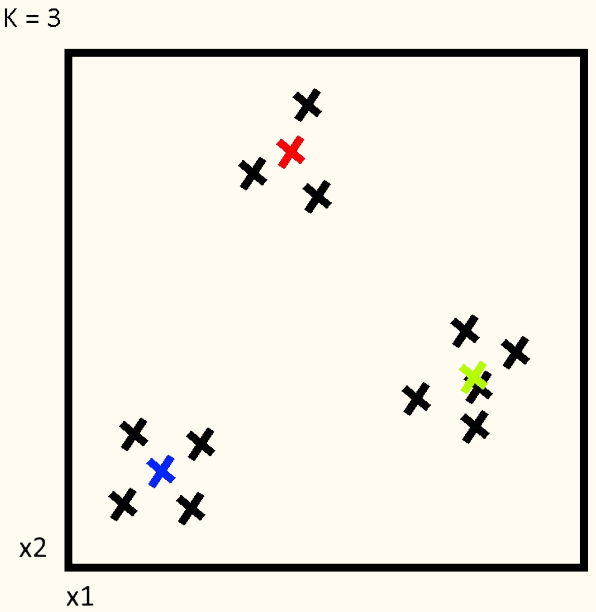
Feature Extraction

- Again, reduces R^p to R^d where d << p
- Construct smaller set of new features from current ones
- Restricts the dataset of features to linear transformations of the input dataset to output our dataset



k-means Clustering (Intuitive)

- Goal —
  - Minimize the averaged distances between the center points and points within each cluster
- Input —
  - m (data points/sets of features)
  - k (number of clusters)
- Output —
  - k clusters centered on k center points
- NP-complete
- Commonly solved with Lloyd's Algorithm



k-means Clustering (Nitty-gritty)

- Indicator matrix —
  - iff point i is in cluster j —
  - $X_{ij} = 1 / \sqrt{\text{number of points in cluster j}}$
- Given —
  - Dataset $A \in R^{mxn}$ (m data points wrt n features)
  - k clusters
- Output —
  - Indicator matrix $X_{OPT} \in R^{mxk}$ which satisfies — $$\mathbf{X}_{\text{opt}} = \underset{\mathbf{X} \in \mathcal{X}}{\operatorname{argmin}} \|\mathbf{A} - \mathbf{X}\mathbf{X}^{\mathsf{T}}\mathbf{A}\|_{\text{F}}^2.$$
- Strange formatting! —
  - Viewed from a linear algebraic standpoint for later ease of manipulation

γ-approximate k-means

$$\min_{\mathbf{X}\in\mathcal{X}} \|\mathbf{A} - \mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{A}\|_{\mathrm{F}}^2 = \mathrm{F}_{\mathrm{opt}}$$

- Given —
  - $A \in R^{mxn}$ (m data points with n features)
  - k clusters
- Goal —
  - Indicator matrix $X_\gamma \in R^{mxk}$ with probability at least $1 - \delta_\gamma$,

$$\|\mathbf{A} - \mathbf{X}_\gamma\mathbf{X}_\gamma^{\mathrm{T}}\mathbf{A}\|_{\mathrm{F}}^2 \leq \gamma \cdot \mathrm{F}_{\mathrm{opt}}$$

Focus

- Dimensionality reduction via —
  - Feature selection
  - Feature extraction
- Focus — Feature Extraction
- Extends JLS by bypassing preserving pairwise (Euclidean) distances, and instead proving that after dimensionality reduction, the optimal clustering of the data is still preserved

Algorithm

- Input —
  - Dataset $\mathbf{A} \in \mathrm{R}^{mxn}$
  - k number of clusters
  - $0 < \varepsilon < ⅓$
- Output —
  - $\mathbf{C} \in \mathrm{R}^{mxr}$, $r = O(k/\varepsilon^2)$
- Algorithm —
  - Set $r = c_2 * k/\varepsilon^2$, for a sufficiently large constant c_2 (theory vs practice)
  - Compute a random nxr matrix $\mathbf{R}$ like for all $i = 1,...,n$, $j = 1,...,r$ (iid)
    - $\mathbf{R}_{ij} = \{+1/\sqrt{r}$ w.p. ½□, $-1/\sqrt{r}$ w.p. ½$\}$
  - Compute $\mathbf{C} = \mathbf{AR}$ (using the Mailman Algorithm)
  - Return $\mathbf{C} \in \mathrm{R}^{mxr}$

Cost Comparisons

- Paper —
  - Space — $O(k/\varepsilon^2)$ dimensions (features)
  - Time — $O(mn\lceil k/\varepsilon^2 \log(n) \rceil)$
  - Approximation ratio — $2 + \varepsilon$
- Exact SVD (1)
  - Space — k dimensions (features)
  - Time — $O(mn \min\{m, n\})$
  - Approximation ratio — 2
- Exact SVD (2)
  - Space — $O(k/\varepsilon^2)$ dimensions (features)
  - Time — $O(mn \min\{m, n\})$
  - Approximation ratio — $1 + \varepsilon$

(1) P. Drineas, A.Frieze, R. Kannan, S.Vempala, V.Vinay. Clustering in large graphs and matrices, *Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms*, 1999

(2) P. Drineas, R. Kannan, and M. Mahoney. Fast Monte Carlo algorithms for matrices I: Approximating matrix multiplication. *SIAM Journal of Computing*, 2006

Theorem

- Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ and k be the inputs for the k-means clustering problem. Let $\varepsilon \in (0, \frac{1}{3})$, and construct features $\mathbf{C} \in \mathbb{R}^{m \times r}$ with $r = O(k/\varepsilon^2)$. Run any k-means γ-approximation algorithm with failure probability Δ_γ on $\mathbf{C}$, k to construct $\mathbf{X_\gamma}$, then with probability $0.96 - \Delta_\gamma$ —

$$||A - X_\gamma X^T_\gamma A||_F^2 \leq (1 + (1 + \varepsilon)\gamma) ||A - X_{opt} X^T_{opt} A||_F^2$$

Theorem (Intuitive)

- Given any set of points in n-dimensional space and k number of clusters, it suffices to create roughly $O(k)$ new features via random projections and then run some k-means algorithm on the new input.
- The clustering obtained in the low-dimensional space will be close to the clustering it would have been obtained after running the k-means method on the original high-dimensional data.
- $(2+\varepsilon)$-error

Supporting Lemma 1 (Lemma 9)

- Argues that the Frobenius norm squared of matrix $\mathbf{Y}$ is comparable to the Frobenius norm squared of matrix $\mathbf{YR}$
- Given —
  - Matrix $\mathbf{Y} \in \mathrm{R}^{mxn}$
  - $k > 1$
  - $\varepsilon > 0$
- $P(|||\mathbf{YR}||_{\mathrm{F}}^{2} \geq (1 + \varepsilon)||\mathbf{Y}||_{\mathrm{F}}^{2}) \leq 0.01$

Supporting Lemma 2 (Lemma 10)

- Given —
  - Matrix $\mathbf{A} \in \mathrm{R}^{mxn}$ with rank p ($k < p$)
  - SVD of $\mathbf{A_k} = \mathbf{U\Sigma V}$
  - $0 < \varepsilon < ⅓$
  - $\mathbf{R} \in \mathbb{R}^{nxr}$ be the rescaled sign matrix
- With $P \geq 0.97$ (failure rate of 0.03)
  - For all $i = 1,...,k$:
    - $1 - \varepsilon \leq \sigma_k^{\,2} (\mathbf{V_K}^{\mathbf{T}} \mathbf{R}) \leq 1 + \varepsilon$
  - There exists an mxn matrix $\mathbf{E}$ such that
    - $\mathbf{A = AR(V\ R)V + E}$
    - $||\mathbf{E}||_{\mathbf{F}} \leq 3\varepsilon||\mathbf{A - A_k}\,||_{\mathrm{F}}$

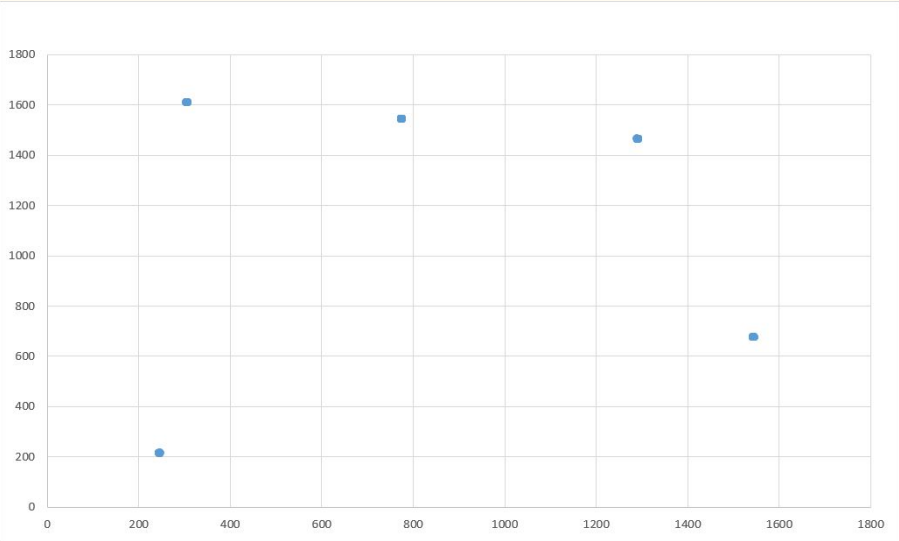
Failure Probability (Theoretically)

- Union bound on Lemmata 1 and 2, along with the failure probability Δ_γ of the γ-approximation k-means algorithm
- Failure probability — $0.04 + \Delta_\gamma$

Dataset

1. Generate 5 centers uniformly random in a n-dimensional hypercube of range of [0, 2000]. (centers will not be part of the data set)
2. From those 5 centers have each generate 200 data points using a Gaussian distribution with a variance of one (1) centered at that center.

N = 2:



Kmeans Experiment

Recall the goal of LLoyd's algorithm is to split points into k clusters such that the total sum of the squared Euclidean distances of each point to its nearest cluster center is minimized.

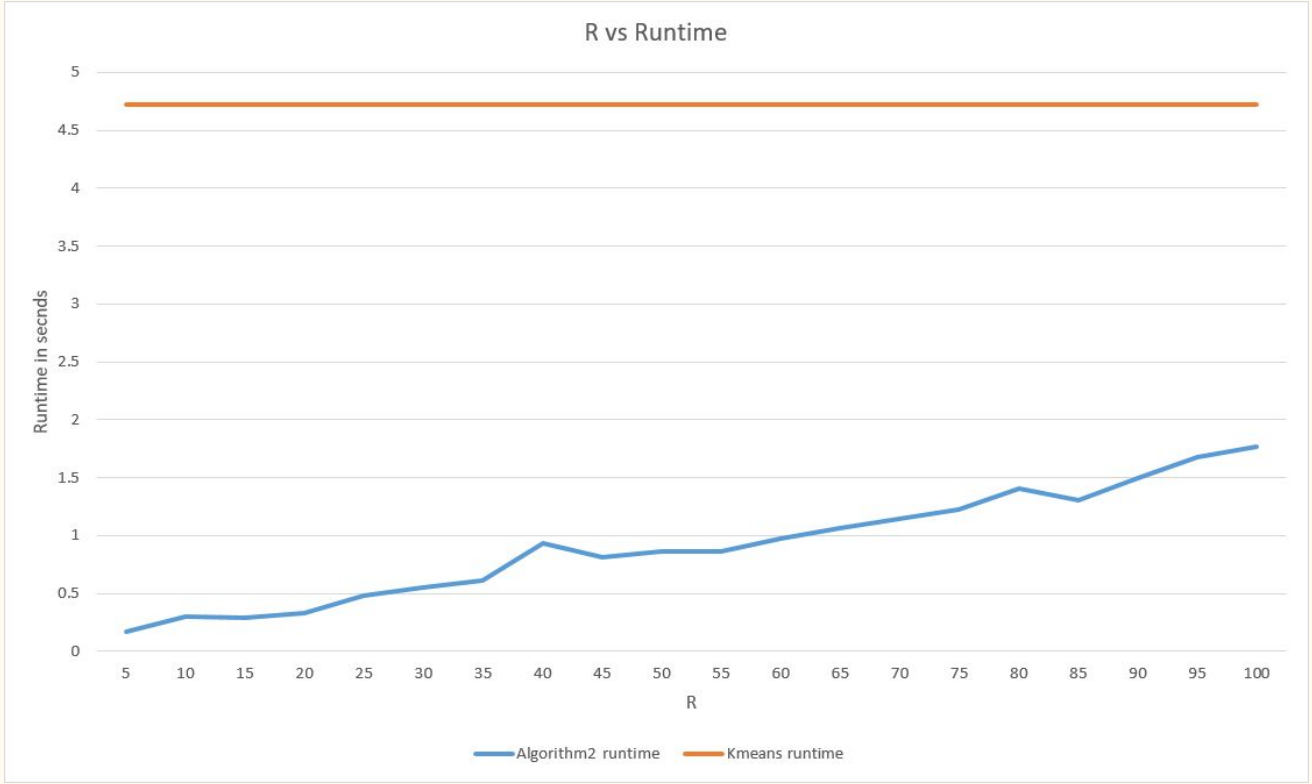
1. Run Lloyd's algorithm 5 times on the dataset. We stop either when max iterations are reached or improvements can no longer be made.
2. Take the best result from Lloyd's algorithm to get our cluster centers.

Algorithm 2 Experiment

Recall the goal of LLoyd's algorithm is to split points into k clusters such that the total sum of the squared Euclidean distances of each point to its nearest cluster center is minimized.

1. Use algorithm time to generate a Matrix $C \in R^{m \times r}$
2. Run Lloyd's algorithm 5 times on C. We stop either when max iterations are reached or improvements can no longer be made.
3. Take the best result from Lloyd's algorithm to get our cluster centers.

Empirical Evaluation



Questions?