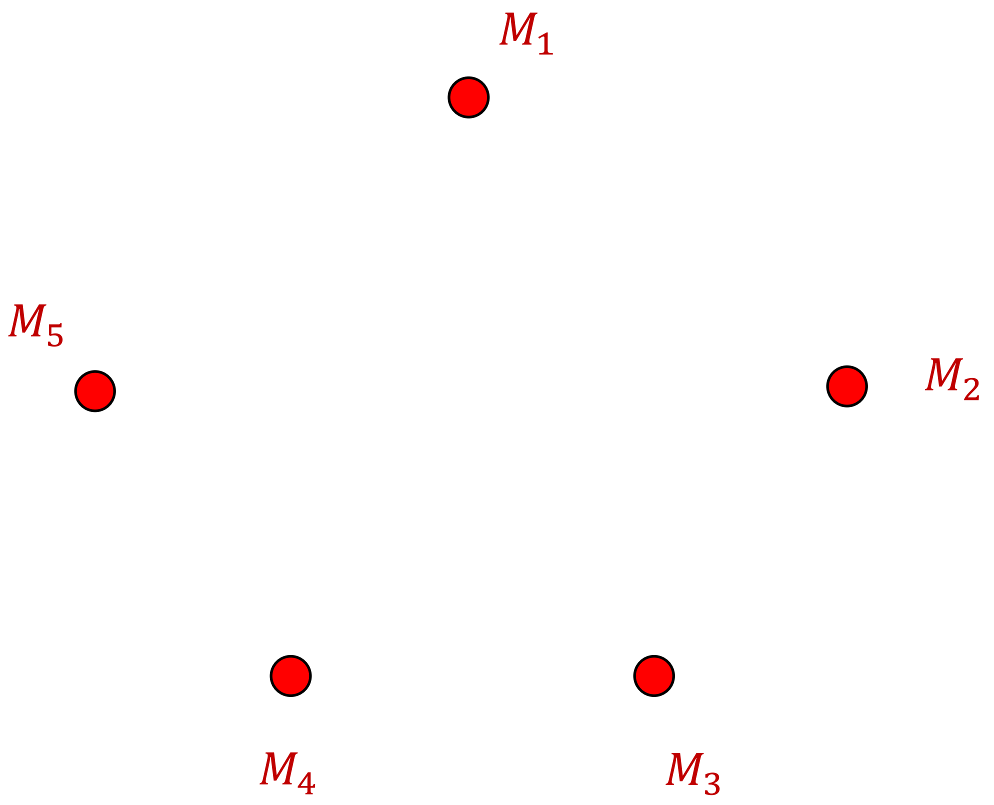
Learning a Latent Simplex in Input-Sparsity Time

Ainesh Bakshi (Carnegie Mellon University)
Chiranjib Bhattacharyya (Indian Institute of Science)
Ravi Kannan (Microsoft Research India)
David P. Woodruff (Carnegie Mellon University)
Samson Zhou (Carnegie Mellon University)

ICLR

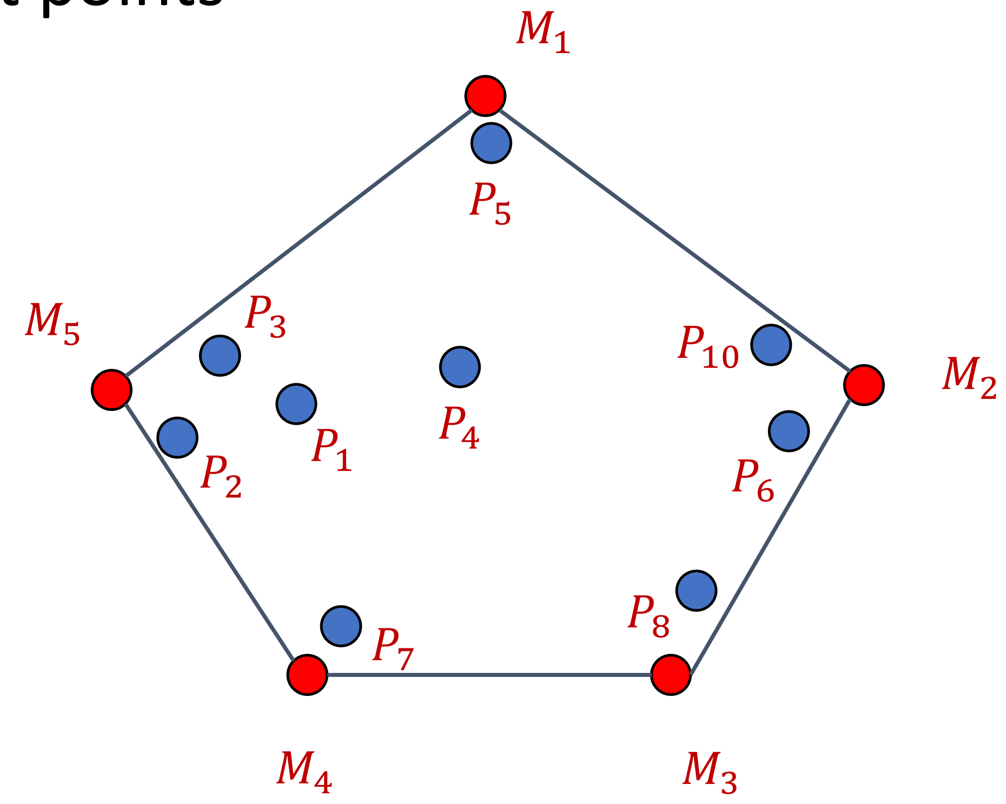
Latent Simplex Model

- ❖ $M_1, \ldots, M_k \in R^d$ vertices of a k-simplex S
- ❖ $d = 2, k = 5$:



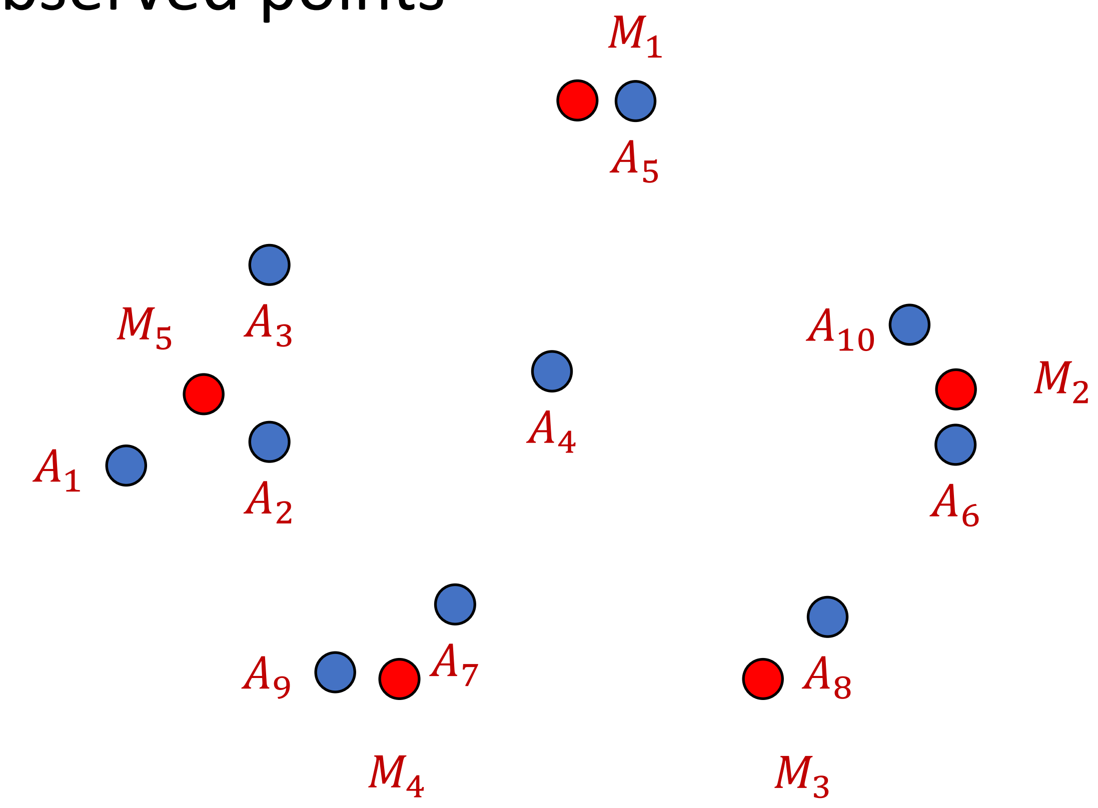
Latent Simplex Model

- $M_1, \ldots, M_k \in R^d$ vertices of a k-simplex S
- $P_1, \ldots, P_n$ latent points



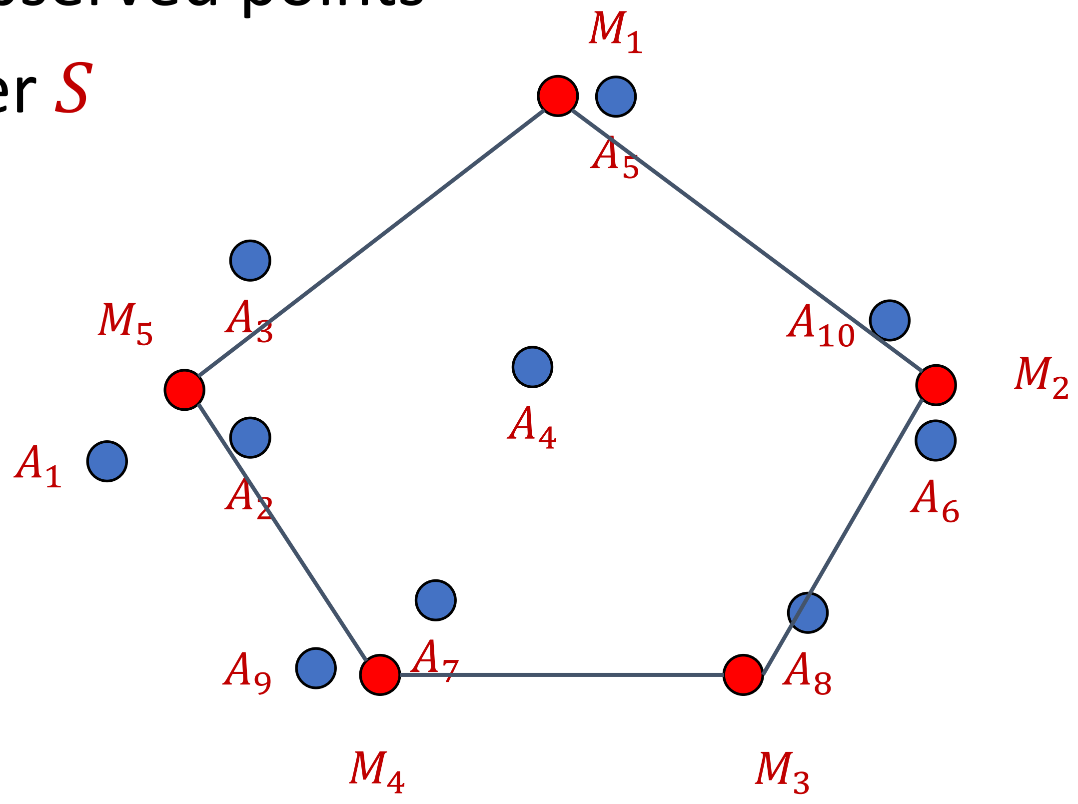
Latent Simplex Model

- $M_1, \ldots, M_k \in R^d$ vertices of a k-simplex S
- $A_1, \ldots, A_n$ observed points



Latent Simplex Model

- $M_1, \ldots, M_k \in R^d$ vertices of a k-simplex S
- $A_1, \ldots, A_n$ observed points
- Goal: recover S



Applications/Motivations

- Topic models: identify abstract topics in a collection of documents by discovering latent semantic structure
- Mixed membership block stochastic model: recover communities in a network by observing frequencies of communication between nodes
- Adversarial clustering: learn the centers of clusters whose mixture forms a set of latent points that may be perturbed adversarially but with bounded norm

Topic Model (Latent Dirichlet Allocation)

- $M_1, \dots, M_k \in R^d$ vertices of a k-simplex S, where d is the size of the dictionary and the i-th entry of M_j represents the frequency of word i in topic j
- $P_1, \dots, P_n$ are latent points (distributions) so that $P_i = W_i M$, where $W_i \sim \mathrm{Dir}(1/k)$
- $A_1, \dots, A_n$ are observed points (documents) so that $A_i = \frac{1}{m}\sum_{j=1}^{m} X_j$, where X_j is an elementary vector drawn from the multinomial distribution P_i

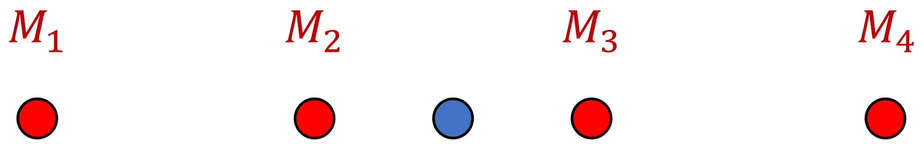
Results and Related Work

- Our result: Given certain geometric assumptions, there exists an algorithm with runtime $\tilde{O}(nnz(A))$ that recovers S, e.g. each vertex is recovered up to "small" Euclidean distance

- Previous: Bhattacharyya and Kannan [BK20] showed that given certain geometric assumptions, there exists an algorithm with runtime $\tilde{O}(k \cdot nnz(A))$ that recovers S, e.g. each vertex is recovered up to "small" Euclidean distance
  - k can be large in applications

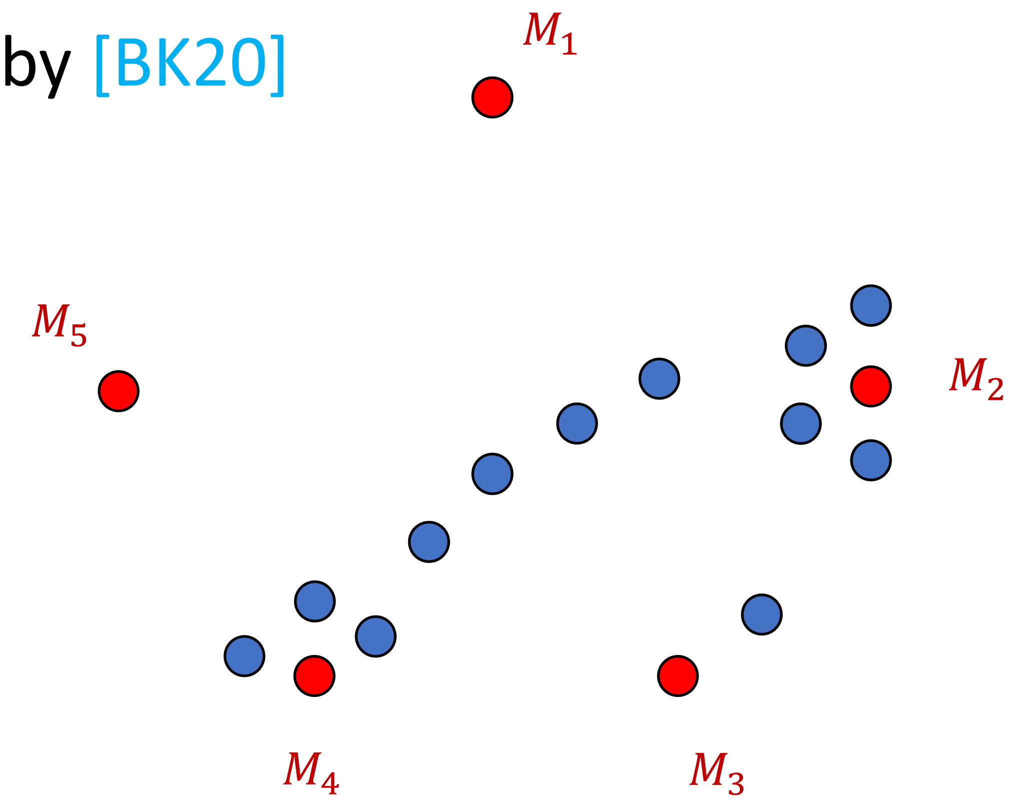
Assumptions (1)

- [Well-Separateness] Each simplex vertex has non-trivial mass in the orthogonal complement of the span of the remaining vectors
- Also assumed by [BK20]



Assumptions (2)

- [Proximate Latent Points] There exists a significant fraction of points near each simplex vertex
- Also assumed by [BK20]

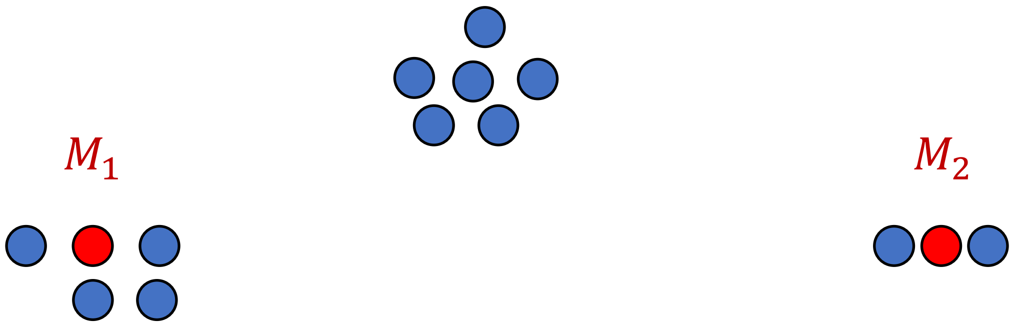


Assumptions (3)

- [Spectrally Bounded Perturbation] Total mass of perturbation should not be too large
- Also assumed by [BK20]

Assumptions (4)

- [Significant Singular Values] The top k singular values should make up most of the mass of the Frobenius norm
- We show this assumption is necessary in improving upon $\tilde{O}(k \cdot nnz(A))$ runtime (else there exists algorithm with same runtime for spectral low rank approximation, which would be algorithmic breakthrough)



Technical Contribution

❖ Show that the subspaces obtained via spectral low-rank approximation are close to the true left and right top k singular space in angular ($\sin \Theta$) distance

❖ To recover S, it suffices to consider the d-dimensional smoothed polytope in the k-dimensional space spanned by the top k singular vectors of the data matrix A

Input-Sparsity Spectral-Frobenius LRA

- Given $A \in R^{d \times n}$, $\epsilon > 0$, there exists an algorithm that outputs matrices Y, Z such that $\|A - YZ\|_2^2 \leq (1+\epsilon)\, \|A - A_k\|_2^2 + \frac{\epsilon}{k} \|A - YZ\|_F^2$ in time $\tilde{O}(nnz(A) + (n+d)poly(k/\epsilon))$ [CohenElderMuscoMuscoPersu15]

- Set ϵ to be the gap in the significant singular values assumption gives constant factor spectral low-rank approximation in input-sparsity time! Use YZ as a proxy for A.
- Avoids $\tilde{O}(k \cdot nnz(A))$ runtime from repeated power iteration

Our Algorithm

- Compute rank k matrices Y, Z so that $\|A - YZ\|_2^2 \leq (1+\epsilon)\|A - A_k\|_2^2$
- Initiate $\tilde{S} = \emptyset$ and repeat k times:
  - Let U_t be an orthonormal basis for the vectors in $\tilde{S}$
  - Compute the projection matrix P_t that projects onto the row span of $\tilde{S}$
  - Generate a random Gaussian $g_t \in R^k$ and set $u_t = g_t Y^\top (I_d - P_t) YZ$
  - Add into $\tilde{S}$ the average of the δn columns of A indexed by the largest δn coordinates of u_t
- Output $\tilde{S}$

Intuition

- The subspaces obtained via spectral low-rank approximation are close to the true left and right top k singular space in angular ($\sin \Theta$) distance
- To recover S, it suffices to consider the d-dimensional smoothed polytope in the k-dimensional space spanned by the top k singular vectors of the approximate data matrix YZ
- Subset smoothing (average of the δn coordinates) to reduce the affects of outliers
- Repeatedly sample random vectors from the subspace orthogonal to the set of vertex approximations picked thus far

рахмат
danke
謝謝
ngiyabonga
spas
teşekkür ederim
Баярлалаа
mersi
kia ora
barka
welalin
tack
mahalo
tapadh leat
спасибо
faafetai lava
vinaka
спасибі
blagodaram
dank je
misaotra
paldies
grazzi
matondo
thank you
gracias
kiitos
dankie
nanni
nandri
dhanyavad
хвала
mauruuru
hvala
bedankt
bayarlalaa
gràçie
köszönöm
akun dankon aĉiŭ
asante
manana
obrigada
tenki
murakoze
enkosi
djiere dieuf
tau
mochchakkeram
дякую
mamnun
chokrane
dziękuję
chnorakaloutioun
gratias ago
gràcies
sulpáy
go raibh maith agat
taiku
sobodi
děkuji
sukriya
kop khun krap
sagolun
obrigado
mèsi
didi madloba
ありがとう
tanemirt
grazie
arigatô
takk
dakujem
trugarez
najis tuke
terima kasih
rahmet
kam sah hamnida
rahmat
diolch
dhanyavadagalu
shukriya
mercé
мерси
তোমাকে ধন্যবাদ
감사합니다
xièxie
ευχαριστώ
merci