An Exploration of the paper ”Power k-Means Clustering” by Jason Xu, Kenneth Lange

Kathryn Carbone, Erik Swanke, Cindy Lyu

April 2023

1 Motivation

1.1 An introduction to clustering

K-Means clustering broadly is a method to separate data into categories of similarity. Formally it is defined as partitioning n data points into k clusters, where each data point belongs to the closest cluster, such that the total variance (squared distance between x_i and the mean of all data inside a cluster) in every cluster is minimized. The general case, where the number of dimensions d and clusters k are uncapped, is NP-complete. Because of this, there exist approximation algorithms that will achieve a local minimum in a relatively fast time complexity, such as Lloyd’s algorithm. Notably, however, the K-Means objective is a nonconvex optimization problem and thus makes finding a global minimum difficult.

1.2 Lloyd’s algorithm and center-based clustering

Vanilla K-Means clustering famously makes use of Lloyd’s algorithm, a simple method for grouping datapoints based on their distance from a pre-selected cluster center. This algorithm is one of several center-based clustering algorithms, which, true to their name, perform clustering around central points within the dataset. For a concrete motivation of Lloyd’s algorithm, suppose we have a simply three-class dataset. We then want to group the dataset into three clusters, each corresponding to a class.

Let $\mathbf{X}$ denote the dataset. C_j denotes a cluster, defined as the set of all datapoints x which belong to this cluster. Finally, define Θ as the center matrix, or the matrix containing all central points for which clustering is performed. The objective of K-Means clustering is defined as such:

$$f_{-\infty} = \sum_{i=1}^{n} min_{1 \leq j \leq k} ||x_i - \theta_j||^2$$

In other words, for each cluster C_j in the range of 1 and k (in our example 3), minimize the Euclidean distance of each datapoint x_i from the center of C_j. Lloyd’s algorithm makes use of a two-step update process to converge at a local minimum, making it popular for its simplicity. This gives us a unique partitioning of datapoints into clusters (without repetition), but is highly susceptible to performance issues in higher dimensional data, and given poor selection of initial centroids. This is because Lloyd’s algorithm employs a greedy update scheme, making it difficult to overcome local minima within the problem space to reach a global minimum. Thus, for a poor selection of initial

Algorithm 1 Lloyd's Algorithm

```
Initialize $\Theta$ with $k$ random datapoints
$X \leftarrow x$
$\Theta \leftarrow \theta$ Randomly initialize centroids
while Not Converged do
    for $x_i \in X$ do
        $j = argmin_j\{||x_i - \theta_j||\}$
        $C_j \leftarrow C_j \cup x_i$                ▷ Reassign each datapoint to its nearest cluster
    end for
    for $j = 1 \cdots k$ do
        $\theta_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i$        ▷ Recalculate each centroid as the average of points in its cluster
    end for
end while
```

centroids, one may be able to achieve a local optimum after running Lloyd's algorithm, but this could be significantly less optimal than other solutions with different initial clusterings.

1.3 Generalized and power means

Before discussing the Power K-Means algorithm, the concept of generalized and power means must first be discussed. The generalized means formula is simply a method of describing many different types of means in a single formula, rather than separate, more specific formulas for each. This is restricted to the domain of functions which are continuous and strictly monotonic. Given a positive integer k and a function $g(y)$ which satisfies these properties, the generalized means formula is defined as such:

$$M_g(\mathbf{y}) = g^{-1}[\frac{g(y_1) + \cdots + g(y_k)}{k}]$$

In words, the generalized mean of a vector $\mathbf{y}$ is the inverse of the function g applied to the mean of $g(\mathbf{y})$. (Recall that an inverse function is defined as such: given $f(x) = y$, $f^{-1}(y) = x$).

Suppose we let $g(y) = y^s$. Then we restrict the generalized mean to the subset of power means, also known as Hölder means. This is expressed as

$$M_s(\mathbf{y}) = [\frac{y_1^s + \cdots + y_k^s}{k}]^{\frac{1}{s}}$$

Consider the gradient for the family of power means, $\frac{\partial}{\partial y_j} M_s(\mathbf{y})$:

$$\frac{\partial}{\partial y_j} M_s(\mathbf{y}) = \frac{1}{k}(\sum_{i=1}^{k} y_i^s)^{\frac{1}{s}-1} \frac{1}{k} y_j^{s-1} > 0$$

Here we can easily see that that the derivative is always greater than 0 and thus the Power Means family consists of strictly increasing functions. Furthermore, since this restriction is taking with respect to each entry of $\mathbf{y}$, each $y_j \in \mathbf{y}$ is also strictly increasing.

Two interesting facts to note are that $s > 1$ yields the family of l_s-norms acting on vector $\mathbf{y}$, $s = 1$ results in the arithmetic mean, and $s = -1$ results in the harmonic mean. Indeed, the harmonic mean has previously been investigated as an improvement to Lloyd's algorithm.

1.4 K-Harmonic Means Algorithm

K-Harmonic Means (KHM) is a center-based, iterative algorithm that refines the clusters defined by K centers. It takes the sum over all data points of the harmonic average of the squared distance from a data point to all the centers as its objective function:

$$f_{-1}(\Theta) = \sum_{i=1}^{N} \frac{k}{\sum_{j=1}^{k}(||x_i - \theta_j||^2)^{-1}}$$

Instead of assigning each datapoint to its closest center (Lloyd's Algorithm), we calculate the membership of x_i according to $C_j \leftarrow \max(\frac{||x_i - \theta_j||^{-4}}{\sum_{j=1}^{k}||x_i - \theta_j||^{-4}})$.

And reassign the new center as $\theta_j = \dfrac{\displaystyle\sum_{i=1}^{N} \frac{1}{d_{i,j}{}^3(\displaystyle\sum_{l=1}^{K} \frac{1}{d_{i,l}{}^2})^2} x_i}{\displaystyle\sum_{i=1}^{N} \frac{1}{d_{i,j}{}^3(\displaystyle\sum_{l=1}^{K} \frac{1}{d_{i,l}{}^2})^2}}$

K-Harmonic Means utilizes the harmonic mean as a proxy function for the minimization function in the k-means optimization problem, with the additional benefit of smoothing the geometry of the problem space. This is significant in that a smoother problem space reduces local minima, making clustering less susceptible to poor selection of initial centroids.

2 Concepts

2.1 Power K-Means objective function

The Power K-means algorithm has the objective function:

$$f_s(\Theta) = \sum_{i=1}^{n} M_s(||x_i - \theta_1||^2, \cdots, ||x_i - \theta_k||^2)$$

which can be expanded to

$$f_s(\Theta) = [\frac{(||x_i - \theta_1||^2)^s + \cdots + (||x_i - \theta_k||^2)^s}{k}]^{-s}$$

as previously mentioned, $f_{-1}(\Theta)$ is KHM objective. $f_{-\infty}(\Theta)$ is the standard k-means because of the end behavior $\lim_{s \to -\infty} M_s(y) = \min\{y_1, \cdots, y_k\}$. Finding the hessian of $f_s(\Theta)$, Minimizing this objective function, and deriving the weight update function in Algorithm 2 will be part of the problem set.

2.2 Majorization minimization

Majorization-minimization (MM) is a general optimization framework for finding the minimum of a non-convex or non-smooth objective function. The principle of MM is to iteratively minimize a sequence of surrogate functions that majorize the original objective function at each step until convergence to a stationary point.

Given a non-convex objective function $f(\theta)$, the MM principle seeks to find a sequence of surrogate functions $g(\theta|\theta_m)$ at the m-th step that satisfy the following two properties:

$$\begin{array}{cc} f(\theta_m) = g(\theta_m \mid \theta_m) & \text{tangency at } \theta_m \\ f(\theta) \leq g(\theta \mid \theta_m) & \text{domination for all } \theta \end{array}$$

The constructed function $g(\theta|\theta_m)$ is the majorized version of $f(\theta)$ with respect to θ at θ_m. Then, update θ_{m+1} by minimizing $g(\theta|\theta_m)$ instead of $f(\theta)$:

$$\theta_{m+1} = \arg\min_{\theta} g(\theta \mid \theta_m),$$

which implies the decrease of $f(\theta)$:

$$f(\theta_{m+1}) \leq g(\theta_{m+1}|\theta_m) \leq g(\theta_m|\theta_m) = f(\theta).$$

Note that all Expectation-Maximization (EM) algorithms for maximum likelihood estimation are instances of MM. Since Lloyd's algorithm can be considered EM for Gaussian mixtures with limiting $\sigma^2 \to 0$, it can be improved by the broader MM principle which leads to the Power k-Means Algorithm.

2.3 The Power K-Means algorithm

The Power k-Means Clustering is derived from Lloyd's Algorithm and MM principle. It can be easily proved that $M_s(\mathbf{y})$ is concave whenever $s \leq 1$ and convex otherwise. By exploiting the concavity and convexity of $M_s(y)$, we find the linear majorization:

$$M_s(\mathbf{y}) \leq M_s(\mathbf{y}) + \sum_{j=1}^{K} \nabla_y M_s(y_m)^T (y_j - y_{m,j}).$$

Note that y_m denotes the estimate f y at iteration m. Substituting $||x_i - \theta_j||^2$ for y_i and $||x_i - \theta_{m,j}||^2$ for $y_{m,j}$ and summing over all data points, the right-hand side of the above inequality then serves as a surrogate function majorizing $f_s(\Theta)$, which can by simplified and denoted as weights:

$$\omega_{m,ij} = \frac{\frac{1}{k}||x_i - \theta_{m,l}||^{2(s-1)}}{\left(\frac{1}{k}\sum_{l=1}^{k}||x_i - \theta_{m,l}||^{2s}\right)^{(1-\frac{1}{s})}}$$

The Power k-Means Clustering successively minimizes the majorization given by weights instead of the Euclidean distance between data points and centers and updates the centroids accordingly as shown in Algorithm 2.

```
Algorithm 2 Power $k$-Means Algorithm
Input: Data matrix $X \in R^{n \times d}$, number of clusters $k$
1: Initialize $s_0 < 0$, $\Theta_0$, $\eta > 1$        ▷ initial exponent $s_0$, exponent increment factor $\eta$
2: repeat
3:     $\omega_{m,ij} \leftarrow \left(\sum_{l=1}^{k} ||x_i - \theta_{m,l}||^{2s_m}\right)^{\frac{1}{s_m}-1} ||x_i - \theta_{m,l}||^{2(s_m-1)}$
4:     $\theta_{m+1,j} \leftarrow \left(\sum_{i=1}^{n} \omega_{m,ij}\right)^{-1} \sum_{i=1}^{n} \omega_{m,ij} x_i$
5:     $s_{m+1} \leftarrow \eta \cdot s_m$        ▷ Optional Annealing
6: until convergence
```

By analogy to the Lloyd's algorithm, this new algorithm updates the weights $\omega_{m,ij}$ (step 3) and the centers $\theta_{m+1,j}$ (step 4) at each iteration with time complexity $O(nkd)$, which retains the efficiency of the original Lloyd's algorithm. Moreover, the additional power parameter s takes the advantage of annealing and makes the Power k-Means Clusering more robust to initialization than the previous Lloyd's Algorithm and KHM. (See proof in section 2.4.1)

2.4 Performance guarantees

Motivation for Power K-means lies in some of the mathematical properties derived from its objective function.

2.4.1 Annealing

Refer to the optional annealing step with respect to s in Algorithm 2.

Geometrically, as s tends to $-\infty$, the power means surfaces $f_s(\Theta)$ provide a family of progressively rougher landscapes, which means more and more local valleys appear on the objective surfaces (bad initialization in Lloyd' Algorithm). In the other extreme when $s = 1$, all optimal centers θ_j collapse to the sample mean $\overline{X}$.

Proposition 2.1 For any decreasing sequence $s_m \leq 1$, the iterates Θ_m produced by the MM updates generates a decreasing sequence of objective values $f_{s_m}(\Theta_m)$ bounded below by 0. Because by the power mean inequality, $M_{s_{m+1}}(f(\theta_{m+1})) \leq M_{s_m}(f(\theta_m))$ for $s_{m+1} \leq s_m$ where $\lim\limits_{s \to \infty} M_s(f(\Theta)) = \min(f(\Theta)) = 0$. As a consequence, the sequence of objective values converges.

In words, altering the objective function at each MM step in our power k-means algorithm still guarantees the descent property and the global minimizer. This also means we can freely choose a schedule for decreasing s in the initialization of the power k-means algorithm.

With s equals to -1, the algorithm serves as k-harmonic means clustering which attempts to optimize one member $f_{-1}(\Theta)$ of an entire family of objectives. Thus, KHM is more robust to initialization than Lloyd's algorithm by applying the heuristic annealing methods. While moving along a sequence of power mean objectives $f_s(\Theta)$, the power k-means algorithm automatically provides a form of annealing that guides the solution path toward a global minimizer as it threads its way across the landscapes. Therefore, the Power K-Means retains the benefits of annealing as dimension d increases, remaining successful in settings where both KHM and Lloyd's algorithm deteriorate.

2.4.2 Proposition 3.1: For any $s \leq 1$, the global minimum $f_s(\Theta)$ lies in the Cartesian product C^k

$P_C(\theta)$ denotes the euclidean projection of θ onto C. Whenever θ_j is not in C, the distance $||x_i - P_C(\theta_j)||^2$ is less than the distance $||x_i - \theta_j||^2$. This is because the angle between x_i, $P_C(\theta_j)$, and θ_j makes an obtuse angle, and a property of obtuse angles is that the side corresponding to the obtuse angle (in this case side x_i to θ_j) must be the longest side. Since

$$M_s(\mathbf{y}) = [\frac{(||x_i - \theta_1||^2)^s + \cdots + (||x_i - \theta_k||^2)^s}{k}]^{-s}$$

is strictly increasing for each of its entries (as shown in section 1.3), we can decrease each term by replacing θ_j with $P_C(\theta_j)$. This means the global minimum must have $P_C(\theta_j) = \theta_j$ for all j, which means Θ is in C^k.

2.4.3 Proposition 3.2: For any decreasing sequence $s_m \leq 1$ tending to $-\infty$, the sequence $f_{s_m}(\Theta)$ converges uniformly to $f_{-\infty}(\Theta)$ on C^k

The proposition states that, supposing there exists a decreasing sequence of powers $s_m \leq 1$ which tends toward $-\infty$, then the corresponding objective function sequence for power means denoted as $f_{s_m}(\Theta)$ will converge uniformly to $f_{-\infty}(\Theta)$, the objective function for K-Means clustering. This is restricted to the compact set C^k, a closed and bounded subset which includes $-\infty$ and ∞. This is a fact which much be proven:

In order to discuss this, first one must consider two properties of the Power K-Means function: first,

$$\lim_{s \to -\infty} M_s(y) = min\{y_1 \cdots y_k\}$$

This means that, as the power s approaches negative infinity, the power mean of vector $\mathbf{y}$ evaluated at s can be approximated as the minimizer of the vector $\mathbf{y}$ and thus converges to a minimum.

Second, for any $s \leq t$, $M_s(\mathbf{y}) \leq M_t(\mathbf{y})$. This is known as the power mean inequality and states that, for any power s less than t, the power means function over vector $\mathbf{y}$ at s must also be less than or equal to the power mean of $\mathbf{y}$ evaluated at t.

Combining these, it is easy to see that, given a continuous decreasing sequence of power means objective functions $f_{s_m}(\Theta)$, an m tending toward infinity (meaning an infinite sequence) implies that $f_{s_m}(\Theta)$ will converge toward $f_{-\infty}(\Theta)$. Furthermore, this convergence is monotonic since every $m+1$-th term in the decreasing sequence s_m is less than its predecessor.

We have thus seen two important properties of the Power Means function: pointwise convergence and monotonicity. Using these properties, we now consider Dini's Theorem: if a monotone sequence of continuous functions converges pointwise on a compact space and the limit function is continuous, then convergence of the sequence of functions is also uniform: let $\epsilon > 0$. Then, for each $f_m \in f_{s_m}$, we know that $f_{-\infty} \leq f_m$.

Consider $g_m = f_m - f_{-\infty}$. Since $f_{-\infty} \leq f_m$, $g_m \geq 0$ for every $f_m \in f_{s_m}$. Furthermore, every g_m is continuous. Next, we know that s_m is a decreasing sequence, so $f_{m+1} \leq f_m$. Therefore

$$f_{m+1} - f_{-\infty} \leq f_m - f_{-\infty} \to g_{m+1} \leq g_m$$

We now define a set C_m such that $C_m = \{\theta \in C^k : g_m(\theta) < \epsilon\}$. Thus, since $-\infty$ is the lower bound on C^k, we see that $-\infty < g_m(\theta) < \epsilon$. Since the set of all functions $\{g_m\}$ decreases, as shown above, each subset C_m increases. We know this because of the definition of C_m, which contains all θ such that $g_m(\theta) < \epsilon$. If g_m decreases, the amount of θ which satisfy this condition increases and thus $C_m \subset C_{m+1}$. We thus know that, since C^k is compact (closed and bounded), there must exist some C_M such that $C_M = C^k$ (note: this makes use of set properties which are above the assumed knowledge of undergraduate students and will this be excluded to simplify proof presentation).

Recall that C_m is the subset of θ such that $g_m(\theta) < \epsilon$. Thus, if $m \geq M$, $|f_{-\infty}(\theta) - f_m(\theta)| < \epsilon$. Also recall that $g_{m+1} \leq g_m$. Thus, $g_{m+1} < \epsilon$ is true as well. We now see that the sequence of g_m (and additionally the original sequence f_{s_m}) converges uniformly (taking step sizes of finite and bounded length) toward the minimum value 0 (when $f_m = f_{-\infty}$).

To conclude, this property is significant in that it guarantees convergence in a finite time and is thus a desireable property for the sequence of Power Means objective functions to have. As the original paper only references Dini's Theorem, resources used in understanding its proof are listed in Appendix B.

3 Experimental Results

Our experimentation will focus on comparing the performance of K-Means, K-Harmonic Means, and Power K-Means on real and synthetic datasets. Areas of analysis will include performance comparisons based on dimensionality and dataset size, power hyperparameter initialization, and distance metric variation. Performance will be measured empirically using the VI (variation of information), NMI (normalized mutual information), and ARI (adjusted Rand index) metrics.

VI measures the distance between two partitions (clusterings) of the same dataset or, in other words, the amount of "information" gained or lost between the two. If this amount of "information," or partitioned datapoints, is large, then the distance is greater and the similarity between the two is also large. As clustering converges, a smaller VI denotes less entropy between the two and thus a stabilization to the clustering [1]. NMI measures the mutual information, or shared datapoints, between each clustering such that a desirable NMI score reveals high similarity within data points in a single cluster and low similarity across different clusters. Finally, the ARI measures the similarity of two clusterings, or the number of objects assigned to the same cluster or different clusters in two clusterings (eg, given two proposed clusterings, is datapoint x_i assigned to C_j in both clusterings).

Current progress has centered around synthetic data performance for K-Means, K-Harmonic Means, and Power K-Means with variable dimensions and power initializations (for Power K-Means). Synthetic datasets are generated using a binomial distribution with true centers at coordinates $(10 * d), (20 * d), (40 * d)$ to account for variance in feature count. Each cluster contains the same amount of data points to better compare performance across all clustering methods. Synthetic dataset generation and Power K-Means algorithm implementation makes use of code from [2], an extension of the paper [3] analyzed in this report.

Code for this project is available at `https://github.com/carbonkat/Power_K_Means_ML_Opt`

Above are results for initial experimentation with varied feature counts for vanilla K-Means using Lloyd's algorithm and Power K-Means. All experiments with Power K-Means use an initial s_0 (power) value of -2 subject to annealing, although future experimentation will include testing with other powers. Interestingly we observe a higher initial NMI score for Power K-Means across all dimensions, although NMI decreases as dimension increases. Future experimentation will seek to determine whether NMI converges or continues to decrease as dimension increases.

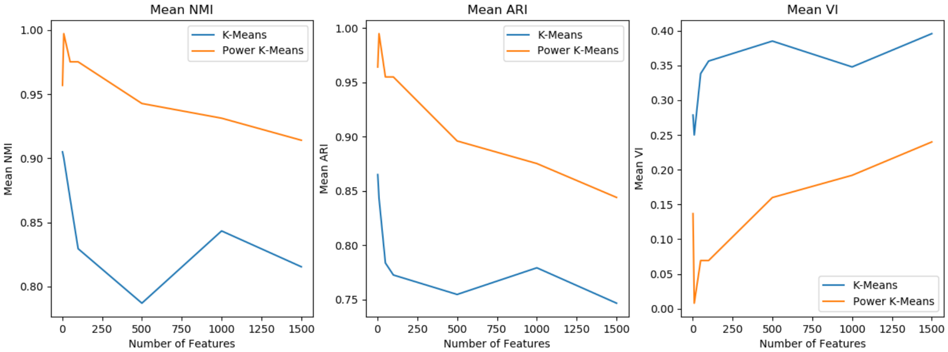

Figure 1: Performances over synthetic datasets with variable features and $s_0 = -2$

	Features	VI Lloyd	ARI Lloyd	NMI Lloyd	VI Power	ARI Power	NMI Power
0	2.0	0.279	0.865	0.905	0.137	0.964	0.957
1	10.0	0.25	0.843	0.9	0.008	0.995	0.997
2	50.0	0.338	0.784	0.868	0.069	0.955	0.975
3	100.0	0.356	0.772	0.829	0.069	0.955	0.975
4	500.0	0.385	0.755	0.787	0.16	0.896	0.943
5	1000.0	0.348	0.779	0.843	0.192	0.875	0.931
6	1500.0	0.396	0.746	0.815	0.24	0.844	0.914

Table 1: Mean performances for K-Means and Power K-Means with $s_0 = -2$

Appendix A Problem Set

The paper introduces a novel algorithm for the classical task of k-means clustering, the power k-means algorithm. The problem set is designed to facilitate the understanding of power k-means clustering and test its performance on additional data set. You will show the derivation process of the formulas discussed in the paper and implement the algorithm by hand.

1. Define the power mean of a vector y:

$$M_s(y) = \left(\frac{1}{k} \sum_{i=1}^{k} y_i^s \right)^{\frac{1}{s}}.$$

   Explore the concavity and convexity of $M_s(y)$.
   Hint: Examine the convexity by computing the gradient and Hessian matrix of $M_s(y)$. Determine the condition under which $M_s(y)$ is convex or concave otherwise.

2. Define the power k-means objective function for given powers s:

$$f_s(\Theta) = \sum_{i=1}^{n} M_s(||x_i - \theta_1||^2, ..., ||x_i - \theta_k||^2).$$

Given the linear majorization derived from the concavity of $M_s(y)$:

$$M_s(y) \leq M_s(y) + \sum_{j=1}^{K} \nabla_y M_s(y_m)^T (y_j - y_{m,j}).$$

Construct MM step in the power k-means clustering by solving the weights of the majorization $\omega_{m,ij}$ and the update rule $\theta_{m+1,j}$ that minimizes the weights. Hint: Focus only on the terms that is relevent to Θ when minimizing the majorization.

3. Implement power k-means algorithm (Algorithm 2) and test your implementation on the Fashion-MNIST data set. Report NMI score on the test and train data sets: see `https://scikit-learn.org/stable/modules/generated/sklearn.metrics.normalized_mutual_info_score.html`. Additionally, set $s_0 = -2$ and $\eta = 1.05$ by default, and try different selection for the initial power s_0 and incremental factor *eta*. Observe the performance. Choose the parameters based on performance or explain why the initial value s_0 do not require careful tuning.

Appendix B Additional Resources

B.1 Dini's Theorem

`https://personal.math.ubc.ca/~feldman/m321/dini.pdf`: A useful walkthrough of Dini's Theorem by Joel Feldman at University of British Columbia

`https://www.youtube.com/watch?v=3E2rxmHhBdg`: a verbal walkthrough of Dini's Theorem by Mohamed A Khamsi.

References

[1] Marina Meilă. “Comparing clusterings—an information based distance”. In: *Journal of Multivariate Analysis* 98.5 (2007), pp. 873–895. ISSN: 0047-259X. DOI: `https://doi.org/10.1016/j.jmva.2006.11.013`. URL: `https://www.sciencedirect.com/science/article/pii/S0047259X06002016`.

[2] Adithya Vellal, Saptarshi Chakraborty, and Jason Q Xu. “Bregman Power k-Means for Clustering Exponential Family Data”. In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri et al. Vol. 162. Proceedings of Machine Learning Research. PMLR, 17–23 Jul 2022, pp. 22103–22119. URL: `https://proceedings.mlr.press/v162/vellal22a.html`.

[3] Jason Xu and Kenneth Lange. “Power k-Means Clustering”. In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, Sept. 2019, pp. 6921–6931. URL: `https://proceedings.mlr.press/v97/xu19a.html`.