# Gap Statistics and Advanced Sampling Methods

**TLDR:** we first talk about model selection problem for K-Means (how to choose optimal number of K?) and analyze how Gap Statistics benefits in this task. Next, the focus is on sampling methods (Direct, Rejection, Adaptive Rejection, Gibbs, MCMC) and how does it relate to GMMs.

## Gap Statistics

Problem statement: how to find optimal number of clusters $k$ for a given dataset?

Optimization objective is the within-cluster distance, defined as:

$$W(C) = \sum_{k=1}^{K} N_k \sum_{C(i)=k} ||x_i - \mu_k||^2$$

where $K$ is the total number of clusters, $C(i)$ is the cluster ID for sample $x_i$, $N_k$ the number of points in cluster $k$, and $\mu_k$ the mean of all points in cluster k.Of course, we can directly minimize $W(C)$, but that's not what we are after. If the number of clusters is equal to the number of points in the dataset, naturally $W(C)$ is 0, forming a "perfect fit". In this case, increasing $K$ will result in lower values for $W(C)$.

So, what do we do? How can we specify some reasonable **lower bound** for $W(C)$?

Remember among probability distributions $p(x)$ which are nonzero over a *finite* range of values $x \in [a, b]$, the maximum entropy distribution is the *uniform* distribution. We leverage this fact. We draw $B$ sets of uniformly distributed samples and on those distributions, we calculate the $\log$ of $W(C)$ . The clustering where largest **gap** between these reference distributions and our original distribution happens is the $k$ we're looking for.

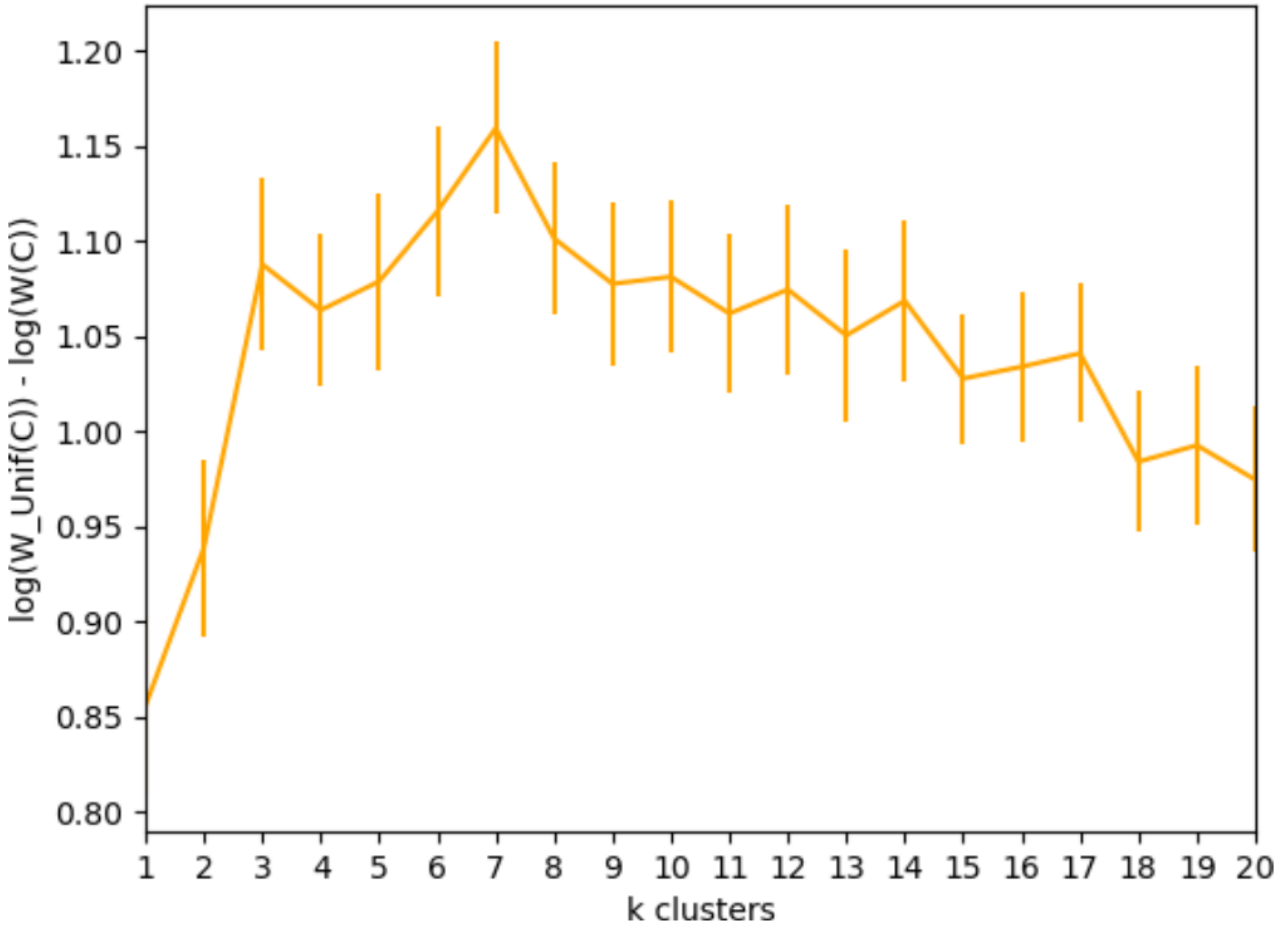
More formally,

1. Draw B sets of uniformly distributed samples
2. On those distributions, calculate for different k the mean of the log of $W(C)$, denote the result $\log(W_{unif}(C))$
3. For $k$ clusters, calculate the gap $G(k)$ as the difference between the reference $log(W_{unif}(C))$ and our $\log(W(C))$.
4. Select the optimum $k$ as

   $$k^* = argmin G(k) > G(k+1) - s_{k+1}$$

   where $s_{k+1} = s_k * \sqrt{1 + 1/B})$ which is an unbiased estimate of the standard deviation $s_k$ of $log(W_{unif}(C)$ .

Since we expect our clustering to be performing better than a random cluster assignment or total point-wise cluster assignment (uniform distribution), the $log(W(C))$ curve is always lower than any of the $B$ uniformly distributed dataset $\log(W(C))$ on within-cluster distance(variance) graph.
In practice, one can identify such optimal $K$ by looking at this graph:




The very first time when the gap is bigger than the next gap minus the standard deviation in the next gap, we have found the optimal $K$. Why do we choose the first such $K$? It's based on Occam's razor.

## Advanced Sampling Methods

Why do want to do sampling?
In most practical cases, we often want to evaluate expectations over the posterior.

**A sampling algorithm** takes a distribution $P$ as input and outputs random values $X_1, \ldots, X_n$ whose marginal distribution is $P$. Ideally, these draws are independent.

## The Key Idea

Consider a probability density $p$ on the interval $[a, b]$.

Suppose we can define a uniform distribution $U_A$ on the blue area $A$ under the curve.

> If we generate $(X_1, Y_1), (X_2, Y_2), \ldots \sim U_A$ then $X_1, X_2, \ldots \sim p$

# Regular Sampling (Transformation/Inverse Sampling)

We look at how we can sample from simple, known distributions and verify the correctness of the Monte Carlo approximation.
Assume we draw $N$ samples from $p(z)$ : $z_i \sim p(z), i = 1, \ldots, N$
We can calculate (approximate ) $E[f]$ by sampling $N$ points and taking their mean. This gives us an *unbiased estimator* :

$$E[f] = E[\frac{1}{N}\sum_{i=1}^{N} f(z_i)] = \frac{1}{N}\sum_{i=1}^{N} E[f(z_i)] = E[f(z)]$$

The variance goes to 0 with an infinite amount of samples.
Given a uniform sampler, we can sample from any *known* distribution:

1. **Discrete RV**: first calculate the CDF $p(z < C)$. Then, given $u_i \sim U(0,1)$ , the sample $k$ of $p(z)$ is where $p(z < k-1) < \mu_i < p(z < k)$. Check out Lecture 1(Densities & Sampling) for more details.
2. **Continuous RV**: calculate CDF by integral and take its inverse $F^{-1}$. Then, given $u_i \sim U(0,1)$, our sample is $z_i = F^{-1}(u_i)$ which is a change of variables.

Inverse sampling fails for two reasons:

1. If the inversion function is impossible to derive analytically, the method cannot be used. For example, the normal integral cannot be directly solved, and hence, the inversion method cannot be used to simulate from the distribution.
2. It does not work with multivariate distributions, because the inverse is generally not unique beyond one dimension.

# Sampling from Parametric Standard Distributions

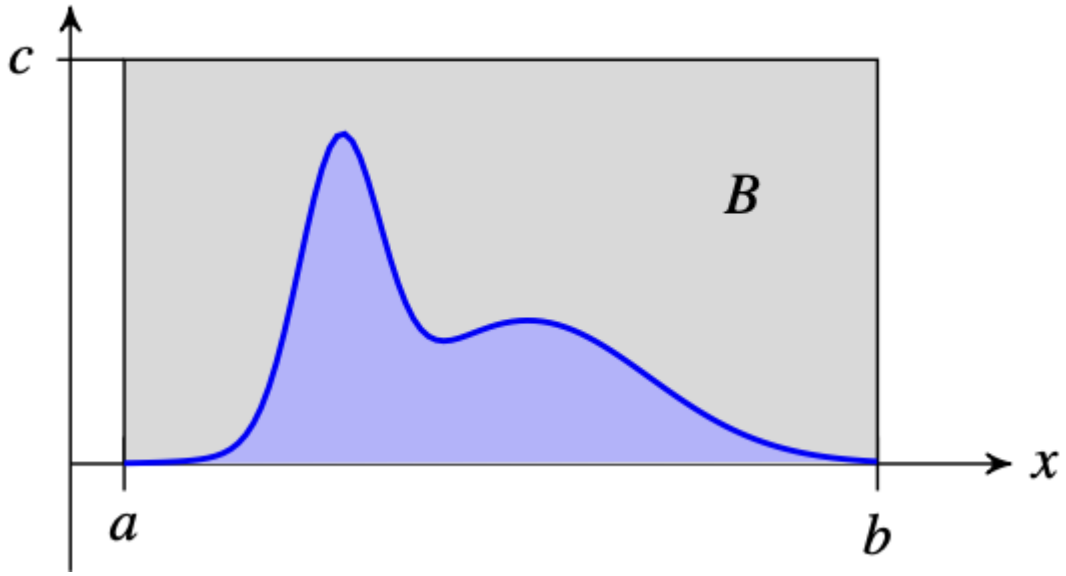
- Analytic mappings from uniform distributions to other distributions exist for
  - Gaussian distributions
  - Exponential distributions $p(y) = \lambda \exp(-\lambda y)$
  - Cauchy distributions $p(y) = \frac{1}{\pi}\frac{1}{1+y^2}$
- This enables a straightforward sampling algorithm:
  - Draw a sample $p(z)$ from a uniform distribution
  - Transform that sample to the target distribution $p(y)$ with the analytic mapping $y = f(z)$
- Note that you need to include the derivative (1-D) or the Jacobian ($>$ 1-D):

$$p(y) = p(z)\left|\frac{dz}{dy}\right| \quad p(y_1, \ldots, y_M) = p(z_1, \ldots, z_M)\left|\frac{\partial(z_1, \ldots, z_M)}{\partial(y_1, \ldots, y_M)}\right| \quad (4)$$

- This sampler is highly efficient, since the mapping is analytic
- Conceptually, this is almost identical to our lecture-1-sampler: replace the CDF calculation by the mapping $y = f(z)$

## Rejection Sampling

**Problem**: we do not know how to define a uniform distribution on an arbitrarily shaped area.
**Solution**: we enclose $p$ in a box $B$, sample uniformly from the box, and discard all draws not in the blue area.




**Algorithm: Rejection Sampling**

- Generate $(X_i, Y_i)$ uniformly on $B$, by independently sampling $X_i \sim Uniform[a, b]$ and $Y_i \sim Uniform[0, c]$
- If $Y_i < p(X_i)$ , keep the sample
- Otherwise: discard ('reject') it
  As a result, the remaining (non-rejected) samples are uniformly distributed on $A$.
  This strategy still works if we scale vertically by some constant $k > 0$. In this case, we

simple draw $Y_i \sim Uniform[0, kc]$ instead of $Y_i \sim Uniform[0, c]$
For this sampling, we just need to know the **shape** (analytical description) of the $p$.

## Rejection Sampling Ver 2.0

**Problem**: if we are not on the interval, sampling uniformly from an enclosing box is not possible (since there is no uniform distribution on all of $R$ or $R^d$ )
**Solution**: instead of a box, we use **a reference distribution** $q(z)$ and a constant $k$ as an envelope such that $k(q(z)) > p(z)$ for all z.

The algorithm is identical to the above case. We always choose $q(z)$ as a simple distribution that we know how to sample and evaluate.

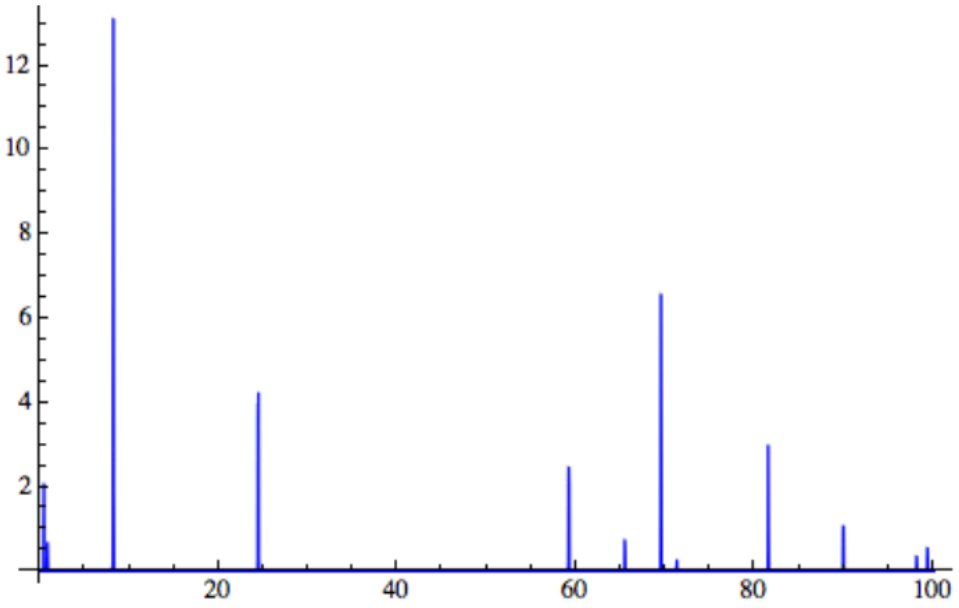
## Rejection Sampling: Version 3.0

**Adaptive Rejection Sampling**:
**Problem:** if target and reference distribution differ too much, rejection sampling becomes too inefficient. ARS constructs $q$ ad-hoc from $p(z)$. This strategy works well for log non-concave functions, since the derivatives of $log(p(z))$ are non-increasing functions of $z$.
I suppose $q(z)$ would be a piece-wise function defined over separate intervals:

$$q(z) = k_i \lambda_i exp(-\lambda_i(z - z_i))$$

**Independence**
If we draw proposal samples **i.i.d** from $q$, the accepted samples are **i.i.d**. If $X_1, X_2, \ldots$ are generated by rejection sampling, $\frac{1}{m} \sum_{i<m} f(X_i)$ is an **unbiased** estimate of $E_p[f(X)]$ .
**Efficiency**:
If q is not a reasonably close approximation of $p$, most proposals are rejected.
Sampling is typically used for distributions of multiple, dependent random variables. Reason: high D distributions with dependence often concentrate on many small areas strewn out over

the sample space, with regions of effectively zero probability in between:



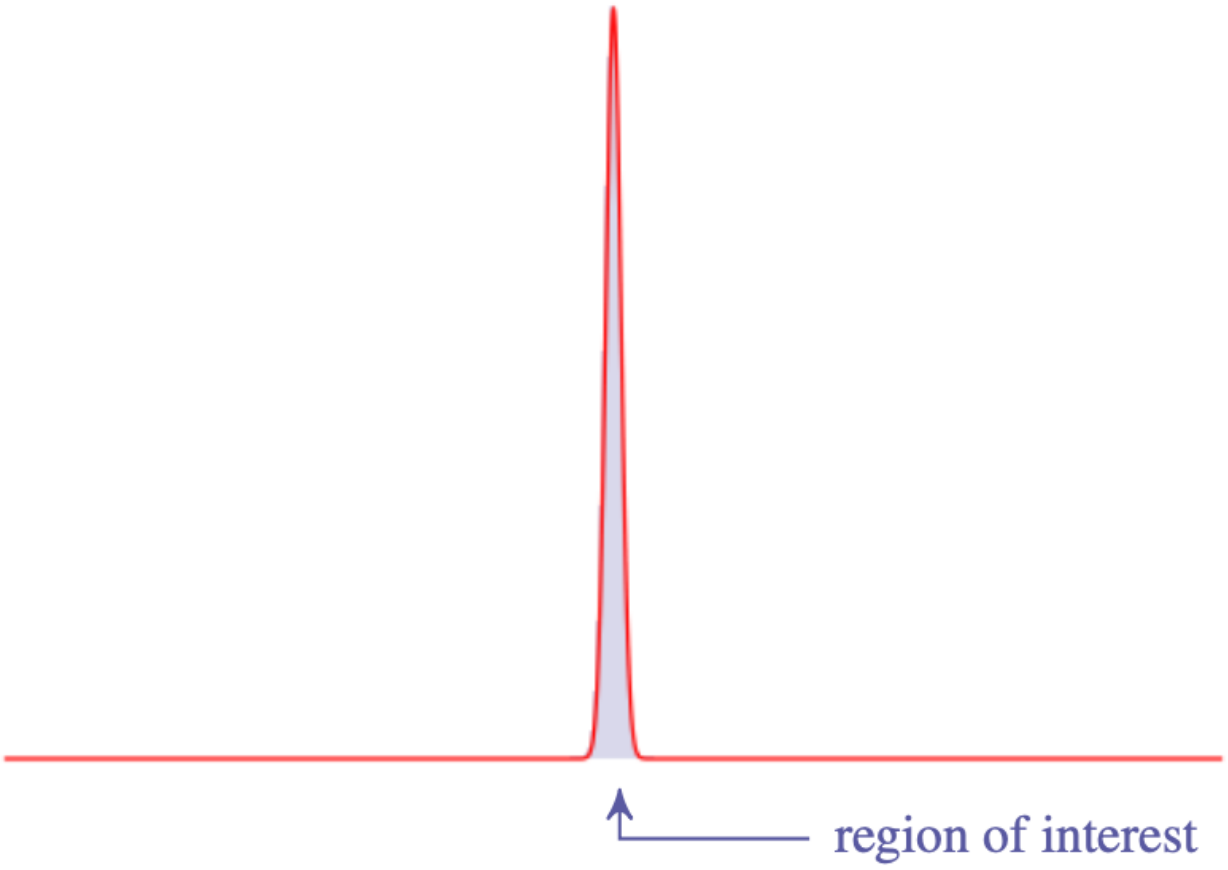

This is a (purely qualitative) attempt to visualize in one dimension what a high-dimensional distribution might look like.

Therefore, sampling in models with many variables or attributes (high dimensional spaces) is difficult with *rejection sampling* or *adaptive rejection sampling*. The gap between $q$ and $p$ increases **significantly** over the dimensions.

## Markov Chain Monte Carlo Sampling

MCMC mitigates these issues in *high-dimensional spaces*. MCMC allows to sample from multivariate densities that are not easy to sample from, often by breaking these densities down into more manageable univariate or multivariate densities.

Suppose we rejection-sample a distribution like this:




Once we have drawn a sample in the narrow region of interest, we would like to continue drawing samples from the same region. That is only possible if each sample *depends on the location of the previous sample*.

Proposals in rejection sampling are *i.i.d*. Hence, once we have found the region where $p$ concentrates, we forget about it for the next sample.
The idea is to introduce Markov Chain property in sampling. In other words, sample $z^t$ using the previous iterations $z^1, \ldots, z^{t-1}$.
MCMC constructs a MC with invariant distribution $p$.

## Gibbs Sampling

By far the most widely used MCMC algorithms is the Gibbs sampler.
**Problem**: sample from the full conditional distribution $p(z) = p(z_1, \ldots, z_M)$ of $M$ random variables.
Each step of Gibbs sampling updates one variable by drawing from the distribution of that variable conditioned on the others.

Suppose $\theta = (\theta_1, \ldots, \theta_K)$, then an iteration of a $K$-component Gibbs sampler is

$$\begin{aligned}
\theta_1^{(t)} &\sim p\left(\theta_1 | \theta_2^{(t-1)}, \ldots, \theta_K^{(t-1)}, y\right) \\
\theta_2^{(t)} &\sim p\left(\theta_2 | \theta_1^{(t)}, \theta_3^{(t-1)}, \ldots, \theta_K^{(t-1)}, y\right) \\
&\vdots \\
\theta_k^{(t)} &\sim p\left(\theta_k | \theta_1^{(t)} \ldots, \theta_{k-1}^{(t)}, \theta_{k+1}^{(t-1)}, \ldots, \theta_K^{(t-1)}, y\right) \\
&\vdots \\
\theta_K^{(t)} &\sim p\left(\theta_K | \theta_1^{(t)}, \ldots, \theta_{K-1}^{(t)}, y\right)
\end{aligned}$$

The distributions above are called the full conditional distributions. If some of the $\theta_k$ are vectors, then this is called a block Gibbs sampler.