



# A “K-means++” style algorithm for improved initialization of composite mixture model parameters prior to batch EM

Siddharth Dangi, Todd Wasson, Jason Lenderman, and Barry Chen  
Knowledge Systems and Informatics Group, Computational Engineering Division

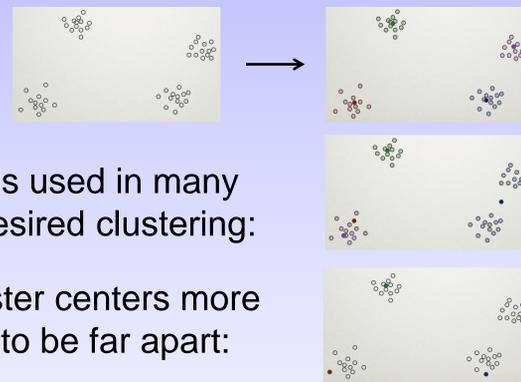
## Introduction and Problem Statement

- **Expectation-Maximization (EM)** is an algorithm for estimating parameters in certain probability models such as **mixture models**.
- In **composite mixture models**, each component is not a just single distribution (e.g., a Gaussian), but rather a product of multiple distributions that could be *continuous* or *categorical*, e.g.: Gaussian × Poisson × Multinomial
- However, EM only finds a local maximum, and it’s performance can be very sensitive to how model parameters are initialized before the algorithm is actually run.
- In composite mixture models, there can be *many local maxima*, and a poor parameter initialization can severely hamper performance.

*Can we design an algorithm to achieve a favorable initialization of composite mixture model parameters prior to running EM?*

## Methods

- We adapted the K-means++ method from the clustering literature. In **clustering**, the goal is to group data points into different classes:
- **K-means** is a simple clustering algorithm that is used in many applications, but doesn’t always achieve the desired clustering:
- **K-means++** is an extension that initializes cluster centers more intelligently, by probabilistically choosing them to be far apart:



K-means++ requires the ability to compute distances between data points – *but how do we compute distances between composite data points? E.g.:*

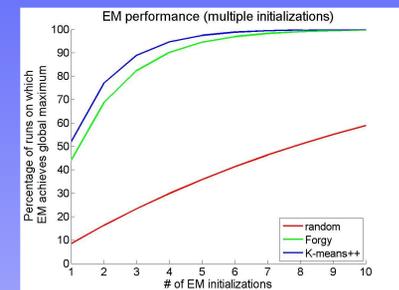
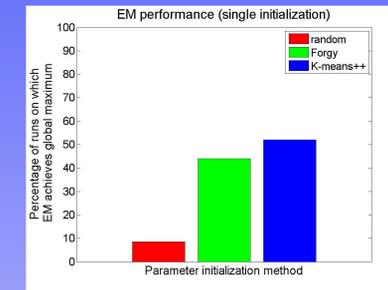
- Point 1: ('192.168.1.7', 80, 'TCP', 1244)
- Point 2: ('34.25.126.93', 80, 'UDP', 5103)
- Point 3: ('192.168.1.7', 4431, 'UDP', 120)

### Main Idea of Our Solution:

1. Represent each composite data point with a composite probability distribution whose mean is that data point.
2. Compute KL divergences between these distributions as a proxy measure of the “distance” between the data points they represent.

## Results

- Constructed a benchmark dataset with known underlying model parameters in order to compare the performance of different EM initialization methods.
- Each composite data point comes from one of 5 mixture components, each of which is a product of a Poisson, Von Mises, Multinomial, Exponential, and 2 Gaussian distributions.



*Our K-means++ style initialization outperforms other methods. Arbitrarily high performance can be achieved by running EM from multiple initializations.*

## Application: Process and User Classification

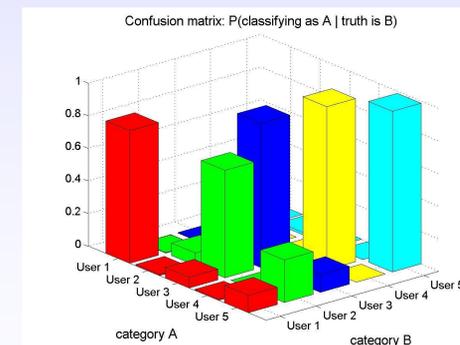
- Using our parameter initialization method as part of a tool called “EMFUDD++” for training composite mixture models of processes, users, and network hosts.
- Analyzed data collected from host- and network-based sensors (**volunteer lab users only; opt-in policy**).
- Simulated imposters by evaluating data from other processes on a model trained for one process (same for users/hosts).
- Combination of host- and network-based sensors generally achieves better performance than host-based sensors alone.

### Examples:

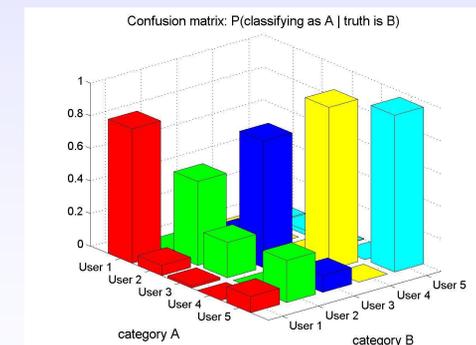
		Actual type	
		Browser	Non-Browser
Predicted type	Browser	98.0%	9.3%
	Non-Browser	2.0%	90.7%

*Browsers (Firefox, IE) are distinguishable from non-browser processes based on protocol, number and sizes of packets, and dest. port.*

### Host-based sensors only



### Host- and network-based sensors



*Users are mostly distinguishable when using only host-based sensors, but classification performance improves with the addition of network-based sensors measuring the number and byte sizes of packets being sent (e.g., User 2).*