**Spark-crowd A Spark Package for Learning from Crowd sourced Big Data**

**ABSTRACT**

As the data sets increase in size, the process of manually labeling data becomes unfeasible  
by small groups of experts. Thus, it is common to rely on crowd sourcing platforms which  
provide inexpensive, but noisy, labels. Although implementations of algorithms to tackle  
this problem exist, none of them focus on scalability, limiting the area of application to  
relatively small data sets. In this paper, we present spark-crowd, an Apache Spark package  
for learning from crowdsourced data with scalability in mind.

**Existing System**

The most important of them is the considerable uncertainty associated with members of the group providing the annotations in the process of data acquisition. Some of them may be better qualified for the task than others; some may give useless information or even try to ruin the results. After this process, the practitioner receives a data set of tuples (example; annotator; annotation). With these, the problem usually consists in estimating the ground truth for each instance using the annotations obtained. As the annotator’s quality is unknown, it is also common to want an estimation of the quality of each annotator. Thus, to obtain good results with crowd sourced data, the machine learning process needs to take into account all these difficulties

**Disadvantages**

1. Some may give useless information or even try to ruin the result
2. provide inexpensive, but noisy, labels.

**Proposed System**

Spark-crowd solved these cases in few minutes. Truth-Inf obtained better results in terms of execution time in comparison to Ceka. As the algorithms chosen in this comparison are quite simple in terms of time complexity, the benefits of parallelization are less apparent, especially in the first two instances for the Majority Voting algorithm, where the implementation in Truth-Inf obtained better results in terms of execution time. However, as the data and the complexity of the algorithm increases, the cost of parallelization is less noticeable. In the Majority Voting algorithm one can see good results in the last two data sets. For the DawidSkene algorithm, the benefits of parallelization can be seen from the second data set, with a significant speedup especially in the third. Truth-Inf was not able to complete the execution for the last data set, due to a memory error.

**Advantages**

1. The cost of parallelization is less noticeable.
2. Increase the efficiency and scope of their research

# Hardware Requirements:

# Processor - Pentium –IV

* Speed - 1.1 GHz
* Ram - 256 MB
* Hard Disk - 20 GB
* Key Board - Standard Windows Keyboard
* Mouse - Two or Three Button Mouse
* Monitor - SVGA

**Software Requirements:**

* Operating System - Windows XP
* Coding Language - java