

Running Multicore Knn on Amazon EC2

Pseudo code

Input: Train Data D, Test Data X, Number of nearest neighbors k

Output: Predicted class labels of X

Step 1: Split D into multiple cores

Step 2: Calculate the distance between each $d_i \in D$ and each $x_j \in X$

Step 3: Send distances to central core

Step 4: Split the distances into multiple cores

Step 5: For each test instance sort the distances

Step 6: Send indices of k nearest neighbors to central core

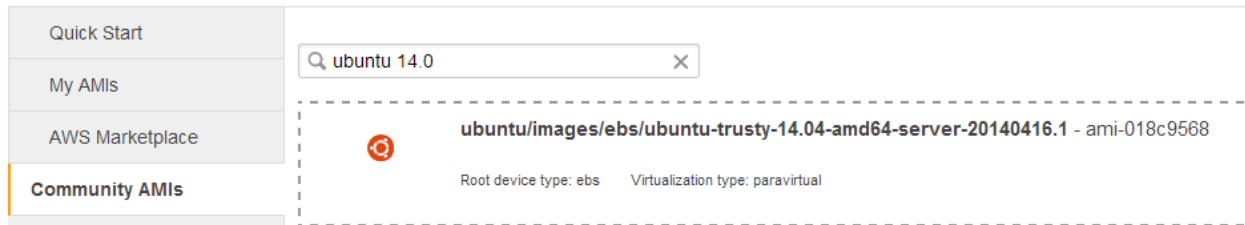
Step 7: Assign most frequent class label from nearest neighbors as predicted class label

How to run the code

Step 1: Logon to Amazon AWS and under Services select 'Ec2'. Click on Launch Instance button.

Step 2: Under the Quick Start column on the left, click on Community AMIs

Step 3: Search for "ubuntu 14.0", select the first AMI "**ubuntu/images/ebs/ubuntu-trusty-14.04-amd64-server-20140416.1 - ami-018c9568**". Click on 'Select' on the right side of the instance.



Step 4: Select the 'Compute Optimized', 'c3.2xlarge' instance and click on 'Next: Configure Instance Details'; at the bottom right.

<input type="checkbox"/>	Compute optimized	c3.xlarge	14	4	7.5	2 x 40 (SSD)	Yes
<input checked="" type="checkbox"/>	Compute optimized	c3.2xlarge	28	8	15	2 x 80 (SSD)	Yes
<input type="checkbox"/>	Compute optimized	c3.4xlarge	55	16	30	2 x 160 (SSD)	Yes

Step 5: Keep everything same and click on 'Next: Add Storage'.

Step 6: Increase the size to 20 Gib and change the 'type' from 'Magnetic' to 'General Purpose (SSD)' and click on 'Next: Tag Instance'.

Type <small>(i)</small>	Device <small>(i)</small>	Snapshot <small>(i)</small>	Size (GiB) <small>(i)</small>	Volume Type <small>(i)</small>	IOPS <small>(i)</small>	Delete on Termination <small>(i)</small>
Root	/dev/sda1	snap-7ac25ca6	20	General Purpose (SSD) ▼	60 / 3000	<input checked="" type="checkbox"/>
Instance Store 0 ▼	/dev/sdb ▼	N/A	N/A	N/A	24 / 3000	N/A

Step 7: Name the instance as 'MulticoreInstance' and click on 'Next: Configure Security Group'.

Key <small>(127 characters maximum)</small>	Value <small>(255 characters maximum)</small>
Name	MulticoreInstance



[Create Tag](#) (Up to 10 tags maximum)

Step 8: Next comes the 'Configure Security Group', keep the default settings and click "Next: Review and Launch".

Step 9: Just check that what we configured is showing up. Click on "launch".

Step 10: In the 'Select existing Key pair or create a new key pair' dialog box, select create new key pair and name it. We gave the name 'dmkd_cpu'. This key pair file will be used to login to the instance. Click on 'Download Key Pair' and click on Launch instance.

Step 11: If everything went well Launch Status page will show up with "Your instance is now launching" statement. Click on "View Instance" button on bottom right. You could see you instance in the running instances by the name '**MulticoreInstance**'.

 MulticoreInstance	i-c1f096eb	c3.2xlarge	us-east-1c	 running
---	------------	------------	------------	---

Step 12: When you select the instance, Instance description shows up at the bottom. Copy the 'Public DNS' value, it will be something like 'ec2-.....-amazonaws.com'.

Windows Login

Step 1: Open '[Puttygen](#)' and in the dialogbox click on 'Load' and select the keypair file which you downloaded in Step 10 "dmkd_cpu.pem". In the search dialog box, select 'All files' at the bottom right which will show the '.pem' file. Click on open and click 'OK' for successful import notice. Click on 'Save private key' at the bottom right and click 'yes' to the warning. Save the file with the same name and without the '.pem' extension, Putty will automatically add the .ppk extension to the newly created file.

Step 2: Open [Putty](#) and in the 'Host name' type 'ubuntu@<DNS value which you copied in Step 12>'. Like 'ubuntu@ec2-.....-amazonaws.com'.

Step 3: Under the connection category in the left panel, select the '+' near 'SSH' and click on 'Auth'. Browse for the '.ppk' which we created in Step 12 and click open. If everything goes well, it will connect to the amazon instance.

Copying Code

- Step 1:** Open [WinSCP](#) and in the dialog box for 'Host name', paste the DNS name which you copied in Step 12 of instance creation. Under User name, type "ubuntu" and click on 'Advanced' under Password text box.
- Step 2:** In the 'Advanced Site Settings' pop up, click on 'Authentication' under 'SSH' in the left column. Browse for the Private Key file '.ppk' which you created in Step 1 of 'Windows Login'.
- Step 3:** Click on 'Login' in the main window and it will connect you the amazon AWS instance.
- Step 4:** Browse to the **MultiCore_Knn** code in the left window and copy the folder on the right side (on instance, /home/ubuntu). Close WinSCP, once done.

Running the Code

Step 1: Login into AWS through PuTTY.

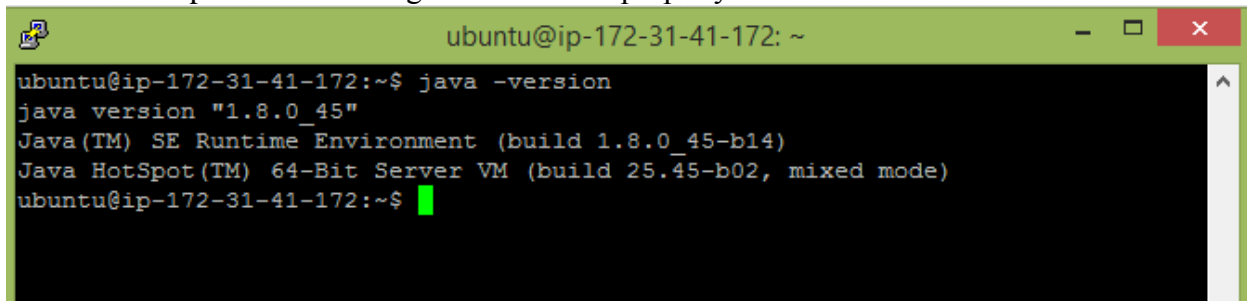
Step 2: Install Java 1.8 with the following commands.

- `sudo add-apt-repository ppa:webupd8team/java`
- `sudo apt-get update`
- `sudo apt-get install oracle-java8-installer`

You can check whether java has installed properly by the following command

- `java -version`

You should expect the following if it's installed properly



```
ubuntu@ip-172-31-41-172: ~  
ubuntu@ip-172-31-41-172:~$ java -version  
java version "1.8.0_45"  
Java(TM) SE Runtime Environment (build 1.8.0_45-b14)  
Java HotSpot(TM) 64-Bit Server VM (build 25.45-b02, mixed mode)  
ubuntu@ip-172-31-41-172:~$
```

Step 3: go to the directory **MultiCore_Knn** and find the executable jar **MultiCore_KNN.jar** and data file **data_30k.txt**. Then run multi threaded Kmeans using the following command

- `java -Xmx16000m -jar MultiCore_KNN.jar train.txt test.txt 15997 4000 10000 20`
- `java <set max heap> -jar <jar file name> <train data name> <test data name> <number of rows in train data> < number of rows in test data> <number of columns> <number of nearest neighbors>`

Note that default JVM heap space of any instance is very low; therefore running the code requires manually setting java heap space so that it can accommodate the big input data.

Cleanup (Important)

Step 1: Logon to Amazon AWS and under Services select 'Ec2'.

Step 2: Under the 'Instances' tab in the left column; click on 'Instances'.

Step 3: Locate your instance (here MulticoreInstance) and select it. On the top locate 'Actions' drop down button and click 'Stop' to stop instance. You can start it and connect to the same settings whenever you want. If you terminate it, you have to create a new instance all together.