Machine learning and big data

Another area that can be exploited using machine learning is big data. After the first release of Apache Hadoop, which implemented an efficient MapReduce algorithm, the amount of information managed in different business contexts grew exponentially. At the same time, the opportunity to use it for machine learning purposes arose and several applications such as mass collaborative filtering became reality.

Imagine an online store with a million users and only one thousand products. Consider a matrix where each user is associated with every product by an implicit or explicit ranking. This matrix will contain 1,000,000 x 1,000 cells, and even if the number of products is very limited, any operation performed on it will be slow and memory-consuming. Instead, using a cluster, together with parallel algorithms, such a problem disappears and operations with higher dimensionality can be carried out in a very short time.

Think about training an image classifier with a million samples. A single instance needs to iterate several times, processing small batches of pictures. Even if this problem can be performed using a streaming approach (with a limited amount of memory), it's not surprising to wait even for a few days before the model begins to perform well. Adopting a big data approach instead, it's possible to asynchronously train several local models, periodically share the updates, and re-synchronize them all with a master model. This technique has also been exploited to solve some reinforcement learning problems, where many agents (often managed by different threads) played the same game, providing their periodical contribute to a **global** intelligence.

Not every machine learning problem is suitable for big data, and not all big datasets are really useful when training models. However, their conjunction in particular situations can drive to extraordinary results by removing many limitations that often affect smaller scenarios.

In the chapter dedicated to recommendation systems, we're going to discuss how to implement collaborative filtering using Apache Spark. The same framework will be also adopted for an example of Naive Bayes classification.

Note

If you want to know more about the whole Hadoop ecosystem, visit http://hadoop.apache.org (http://hadoop.apache.org). Apache Mahout (http://mahout.apache.org (http://mahout.apache.org)) is a dedicated machine learning framework and Spark (http://spark.apache.org (http://spark.apache.org)), one the fastest computational engines, has a module called **MLib** that implements many common algorithms that benefit from parallel processing.

Previous Section (/book/big_data_and_business_intelligence/9781785889622/1/ch01lvl1s

Next Section (/book/big_data_and_business_intelligence/9781785889622/1/ch01lvl1sec1