CS – 549

Paper Review - QuickFOIL: Scalable Inductive Logic Programming

Akhil Sai Chintala (934409906)

- **What is the problem discussed in the paper?**
  Before discussing the problem in the paper, the main topic the paper explains is Inductive Logic Programming (ILP). ILP is a part of machine learning which is used to represent hypotheses and data which helps in learning first order rules from relational structured data. Here as explained above, the greatest number of ILP systems are only applied to small datasets. The challenge here is to scale ILP methods to large datasets. This was done by using a method called QUICKFOIL which enables some strategies to find high quality rules and can be used in Relational-DBMS.

- **Why is it important?**
  From the paper, I feel it is important because ILP systems cannot handle the large datasets which have beyond thousands of training examples. However, many enterprises and scientific datasets have a greater number of larger and complex datasets. So, it is important to find ILP methods that can handle large datasets. The ILP method mentioned in the paper is QUICKFOIL.

- **What are the main ideas of the proposed solution for the problem?**
  The main ideas of the proposed solution are,
  a) By introducing a good learning algorithm that can produce high quality results in a small number of search iterations without exhausting a large number of clauses which results in a scalable and efficient implementation.
  b) A new ILP method named QUICKFOIL which uses a top-down, greedy search with a novel scoring function and a new pruning strategy to meet all the challenges. They have also developed query processing techniques for efficiency in database implementation of QUICKFOIL.

- **What are the shortcomings of the proposed solution?**
  One of the challenges for the proposed solution is that building a scalable R-DBMS version of QUICKFOIL is to optimize the performance of large number of join queries on potentially large relations when searching for the best literals. The only solution which can solve this problem is to maximize the sharing amongst the queries while minimizing the materialization cost that is incurred by sharing.