Object Oriented Machine Learning with a Multicore
Real-Time Java Processor

JTRES ‘10

Rasmus Ulslev Pedersen
Dept. of Informatics, CBS

Martin Schoeberl
DTU Informatics
Department of Informatics and Mathematical Modeling
Technical University of Denmark
Overview of talk

- Motivation
- Real time machine learning
- Multicore machine learning
- Implementation platform
- Experiments
- Conclusion
Motivation

- Machine learning is important in a number of domains
- Java is widely used – also in machine learning systems
- Previous experience with JOP (multicore)
Support Vector Machines

\[ f(x, \alpha, b) = \{ \pm 1 \} = \text{sgn} \left( \sum_{i=1}^{l} \alpha_i y_i k(x_i, x) + b \right) \]

\[ \text{maximize } W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j k(x_i, x_j) \]

```c
/**
   * Method getKernelOutput, which returns the kernel of two points.
   *
   * @param i1 - index of alpha_fp 1
   * @param i2 - index of alpha_fp 2
   * @return kernel output
   */
float getKernelOutputFloat(int i1, int i2) {
    kernelCalls ++;
    return KFloat.kernel(i1, i2);
}
```
Implementation

```java
private static class Test implements Execute {
    final static int N = 100;
    static int a[] = new int[N];
    
    // the work method for one iteration
    public void execute(int nr) {
        a[nr] = nr;
    }
    
    public static void result() {
        for (int i = 0; i < N; ++i) {
            System.out.println(a[i]);
        }
    }
}
```
**Discussion**

- **Hard-real time SVM**

![Graph showing Kernel counts (KC) vs. number of cores](image)

```java
public float getFunctionOutput(float[] p, boolean parallel) {
    float functionalOutput_fp = 0;
    if (parallel) {
        functionalOutput_fp = 0.0f;
    } else {
        functionalOutput_fp = 0.0f;
    }
    for (int i = 0; i < p.length; i++) {
        functionalOutput_fp *= (1.0f - p[i]);
    }
    return functionalOutput_fp;
}
```
Conclusion

- We achieved linear scalability for two cores
- Presented a popular machine learning algorithm
- Conclusion that objected oriented intelligent algorithms are prime for further investigation