|  |  |  |  |
| --- | --- | --- | --- |
|  | Slides | Time | Say |
| 1 |  | 20 | (0:00-0:20) Hello, everyone. I’m Jared. Today, my representation is about “Hybrid Random-Deterministic Parallel Algorithms for Optimization”. |
|  |  | 30 | (0:20-0:50) Firstly, let’s see what is optimization in this paper. It’s a task to solve the minimization of the sum of a smooth function F and a non-smooth convex one G, just as the equation in this slide shows, that is to say, to find a x to make sure that the target function V(x) is minimal |
|  |  | 20 | (0:50-1:10) Actually, there are several solutions to this optimization problem. Four typical of them are listed. Now let’s analyze them one by one. |
|  |  | 30 | (1:10-1:40) The most original solution is BCD, Block Coordinate Descent. In this method, at each iteration, one block of variables is updated using first order information. It has low-cost per iteration and scalability, but it’s inefficient when the dimension increases. |
|  |  | 20 | (1:40-2:00) Next, parallel BCD, it’s the parallel version of BCD, which mean several blocks of variables are updated simultaneously in a parallel environment. So this method can solve big data problem by using parallel environment, but it’s only for convex problems. |
|  |  | 20 | (2:00-2:20) The third kind, random BCD, it performs random selection of blocks of variables to update. So, it’s simple and useful for distributed environments, meanwhile, it remain disconnected from the status of the optimization process. |
|  |  | 20 | (2:20-2:40) Final method, it uses more-than-first-order information to update variables, therefore, it converges faster. The key is the trade-off between cost per iteration and overall cost of the optimization process, which is also the difficulty in this method. |
|  |  | 2:30 | (2:40-5:10) We have seen existing solutions just now. Now I’ll introduce our solution to you. We call it “Hybrid Random-Deterministic Parallel Algorithms”. In this slide, the right graph shows the process of this algorithm. Let me explain it. Step one is a termination criterion. Then in step two, we a random selection is performed, the set is generated , which corresponds to “random” in the title. Next, step three is to find “promising” blocks and step four can update these “promising” blocks, which is called “deterministic” in the title. Finally, all the promising blocks are updated in parallel, that’s why we call it “parallel”. All these steps are performed iteratively until the termination criterion is satisfied. Then, let’s dive into the details of this algorithm. |
|  |  | 40 | (5:10-5:50) Firstly, how to perform random selection? There are mainly two methods. First is uniform sampling, which means all blocks selected with the same probability. Second is double uniform sampling, which means all sets of equal cardinality are generated with equal probability. |
|  |  | 20 | (5:50-6:10) Then, the choice of the step-size gamma upper sub k. It can vary according to the rule in the slide. |
|  |  | 30 | (6:10-6:40) We have seen the details of the proposed algorithm, which enjoys all features in this slide. That is, it’s hybrid random-deterministic. It can solve separable convex function but also non-separable function. It can deal with a non-convex function. It’s parallel. It can use inexact updates. It converges almost surely. |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | 1:00 | (6:40-7:40) This graph is the part of the results. It shows the relative error of different solutions changes with time. The black lines are of our solution, blue lines of random BCD, green lines of random and parallel BCD, pink lines of parallel BCD, red lines of parallel deterministic BCD. Almost the error of every solution decreases sharply, but our solution converges first, which means this hybrid solution performs better than previous solutions in this class of problems. |
|  |  | 20 | (7:40-8:00) In conclusion, we propose a hybrid random-deterministic and parallel algorithm for convex and non-convex functions in big data problems. |
|  |  | 20 | (8:00-8:20) The experiments on Lasso problems showed our hybrid algorithm performs better than state-of-art random and deterministic algorithms. |
|  |  | 20 | (8:20-8:40) In the future, we are going to perform experiments on more varied classes of problems. Thank you, do you have any questions? |