

I'm not a bot



























Coordinate descent does not require knowledge of or computation of the derivative of the objective function; rather, it only considers the coordinates of the function itself. Whereas gradient descent moves in the direction of steepest descent, coordinate descent searches for the minimum of the function by separately moving along the axes of its coordinates. Gradient descent can be conceptualized by picturing a ball rolling down a hill until it reaches the bottom, while coordinate descent is more like someone starting at one corner of a city and following streets arranged in a grid pattern in order to reach the other side. Coordinate descent is preferred when computation and evaluation of a function's derivative is time consuming. Coordinate descent does not require knowledge of or computation of the derivative of the objective function; rather, it only considers the coordinates of the function itself. Whereas gradient descent moves in the direction of steepest descent, coordinate descent searches for the minimum of the function by separately moving along the axes of its coordinates. Gradient descent can be conceptualized by picturing a ball rolling down a hill until it reaches the bottom, while coordinate descent is more like someone starting at one corner of a city and following streets arranged in a grid pattern in order to reach the other side. Coordinate descent is preferred when computation and evaluation of a function's derivative is time consuming. Share — copy and redistribute the material in any medium or format for any purpose, even commercially. Adapt — remix, transform, and build upon the material for any purpose, even commercially. The licensor cannot revoke these freedoms as long as you follow the license terms. Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. ShareAlike — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original. No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits. You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation. No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material. Skip to main content arXiv is a free distribution service and an open-access archive for nearly 2.4 million scholarly articles in the fields of physics, mathematics, computer science, quantitative biology, quantitative finance, statistics, electrical engineering and systems science, and economics. Materials on this site are not peer-reviewed by arXiv. Astrophysics (astro-ph new, recent, search) Astrophysics of Galaxies; Cosmology and Nongalactic Astrophysics; Earth and Planetary Astrophysics; High Energy Astrophysical Phenomena; Instrumentation and Methods for Astrophysics; Solar and Stellar Astrophysics Condensed Matter (cond-mat new, recent, search) Disordered Systems and Neural Networks; Materials Science; Mesoscale and Nanoscale Physics; Other Condensed Matter; Quantum Gases; Soft Condensed Matter; Statistical Mechanics; Strongly Correlated Electrons; Superconductivity General Relativity and Quantum Cosmology (gr-qc new, recent, search) High Energy Physics - Experiment (hep-ex new, recent, search) High Energy Physics - Lattice (hep-lat new, recent, search) High Energy Physics - Phenomenology (hep-ph new, recent, search) High Energy Physics - Theory (hep-th new, recent, search) Mathematical Physics (math-ph new, recent, search) Nonlinear Sciences (nlin new, recent, search) includes: Adaptation and Self-Organizing Systems; Cellular Automata and Lattice Gases; Chaotic Dynamics; Exactly Solvable and Integrable Systems; Pattern Formation and Solitons Nuclear Experiment (nucl-ex new, recent, search) Nuclear Theory (nucl-th new, recent, search) Physics (physics new, recent, search) includes: Accelerator Physics; Applied Physics; Atmospheric and Oceanic Physics; Atomic and Molecular Clusters; Atomic Physics; Biological Physics; Chemical Physics; Classical Physics; Computational Physics; Data Analysis, Statistics and Probability; Fluid Dynamics; General Physics; Geophysics; History and Philosophy of Physics; Instrumentation and Detectors; Medical Physics; Optics; Physics and Society; Physics Education; Plasma Physics; Popular Physics; Space Physics Quantum Physics (quant-ph new, recent, search) General information How to Submit to arXiv Membership & Giving Who We Are When it comes to function minimization, it's time to open a book of optimization and linear algebra. I am currently working on variable selection and lasso-based solutions in genetics. What lasso does is basically minimizing the loss function and a penalty in order to set to zero some regression coefficients and select only those covariates that are really associated with the response. Phew, the shortest summary of lasso ever! We all know that, provided the function to be minimized is convex, a good direction to follow, in order to find a local minimum, is towards the negative gradient of the function. Now, my question is how good or bad is following the negative gradient with respect to a coordinate descent approach that loops across all dimensions and minimizes along each? There is no better way to try this with real code and start measuring. Hence, I wrote some code that implements both gradient descent and coordinate descent. The comparison might not be completely fair because the learning rate in the gradient descent procedure is fixed at 0.1 (which in some cases might be slower indeed). But even with some tuning (maybe with some linear search) or adaptive learning rates, it's quite common to see that coordinate descent overcomes its brother gradient descent many times. This occurs much more often when the number of covariates becomes very high, as in many computational biology problems. In the figure below, I plot the analytical solution in red, the gradient descent minimisation in blue and the coordinate descent in green, across a number of iterations. A small explanation is probably necessary to read the function that performs coordinate descent. For a more mathematical explanation refer to the original post. It's quite common to see that coordinate descent overcomes its brother gradient descent. Coordinate descent will update each variable in a Round Robin fashion. Despite the learning rate of the gradient descent procedure (which could indeed speed up convergence), the comparison between the two is fair at least in terms of complexity. Coordinate descent needs to perform operations for each coordinate update. Gradient descent performs the same number of operations. The R code that performs this comparison and generates the plot above is given below: 

```
library(lasso2) # make data rm(list = ls(all = TRUE)) # make sure previous work is clear ls() p = 70; n = 300
```

x