Gradient Optimization

Evgenii Egorov

Plan

- We would like to make predictions with <w, x>!
  - What is x?
  - How to measure the error?
- We would like to minimize this error over w!
  - How to find the best one?
- Machine learning is not just optimization!
  - How this affects optimization routine?

Linear model

$$a(x) = \langle w, x \rangle + w_0$$

- For each object we have its feature representation: x
- We can do simple predictions:
  - Multiple each coordinate with weight and sum them all
- Benefits of such model?
  - * Your suggestions here * (“Simple” means nothing)

Linear model

$$a(x) = \langle w, x \rangle + w_0$$

- For each object we have its feature representation: x
- We can do simple predictions:
  - Multiple each coordinate with weight and sum them all
- Benefits of such model?
  - Predictions = 1 vector product -> fast
  - Linear change of input/weight -> Linear change in output
    - Interpretation and diagnostic
  - Online learning, sparsity, closed form solution (in many cases), effective training ...

Linear model

$$a(x) = \langle w, x \rangle + w_0$$

- Two frequent statements:
  - Linear assumption is to restrictive
  - Linear models are robust and avoid overfitting
- Are they correct?
  - *Your suggestions here *

Linear model

$$a(x) = \langle w, x \rangle + w_0$$

- Linear assumption is to restrictive
  - Take a big enough “feature space” and be happy
    - Fourier basis, polynomial basis
    - Random Functions
- Linear models are robust and avoid overfitting
  - Take number of features greater than number of objects :)
  - But for *linear model* we can design a way to find robust w:
    - Because number of non-zero weights and complexity of model is “the same”
    - Introducing bias (or prior knowledge) is clear because of linear response

Linear model

$$a(x) = \langle w, x \rangle + w_0$$

- Main restriction for linear model:
  - You should know how to extract features from object
  - (Before deep learning) Computer vision, natural language answer this question for particular domain and particular task in particular domain
  - Most popular algorithms are all about how to avoid hand-crafting

Linear model

$$a(x) = \langle w, x \rangle + w_0$$

- How to make a suitable representation for categorical feature?
  - * Your suggestions here *
- What features we can extract from text?
  - * Your suggestions here *

Linear model

$$a(x) = \langle w, x \rangle + w_0$$

- I have a training dataset!
  - And I extract features
- How to measure the error?

$$\text{MSE}(a, X) = \frac{1}{\ell} \sum_{i=1}^{\ell} (a(x_i) - y_i)^2$$

$$\text{MAE}(a, X) = \frac{1}{\ell} \sum_{i=1}^{\ell} |a(x_i) - y_i|$$

- We can minimize an error measure over the weights!
  - Is it important what to minimize? What if we weight samples?
  - Is model is how to predict or what to minimize?

Linear model

$$a(x) = \langle w, x \rangle + w_0$$

- Let’s derive closed-form solution:
  - Vanila -> with regularization
  - And discuss it a bit
- * Go to whiteboard*

Linear model

$$a(x) = \langle w, x \rangle + w_0$$

- Closed-form solution is nice to have!
  - Study algorithm properties from it
  - Or derive updates for 1-new object arriving

Linear model

$$a(x) = \langle w, x \rangle + w_0$$

- Closed-form solution is nice to have!
- But it is not always available:
  - For some cases we just don’t have it
    - Some cases is nearly all cases
  - Due computational restrictions

Linear model

$$a(x) = \langle w, x \rangle + w_0$$

- Closed-form solution is nice to have!
- But it is not always available:
  - For some cases we just don’t have it
    - Some cases is nearly all cases
- Let’s go to the iterative methods!

Optimization

- General idea:
  - Start from somewhere
  - Try to modify in some direction
- Question:
  - How to select direction?
  - * Your suggestions here *

Optimization

- Let’s go to white-board and derive some algorithms
  - Algorithm =
    - Direction + step size in this direction
    - What information about function it uses
    - And assumptions for convergence