Lecture Notes for

Machine Learning in Python

Logistic Regression

Optimization

Agenda

- Lab 1 general feedback
- Numerical Optimization Techniques
  - Types of Optimization
  - Programming the Optimization
- **Whirlwind Lecture Alert**
  - Entire classes cover these concepts in expanded form
  - But we can cover them in one lecture to get a good intuition!
  - And then you can look over this even more for better understanding.
  - If you feel confused after this lecture, that's okay. These are not easy the first time you see them. Keep going, you got this.

Last time

$$p(y^{(i)} = 1 \mid x^{(i)}, w) = \frac{1}{1 + \exp\left(w^T x^{(i)}\right)}$$

$$l(w) = \sum_i \left(y^{(i)} \ln[g(w^T x^{(i)})] + (1 - y^{(i)})(\ln[1 - g(w^T x^{(i)})]) \right)$$

$$\underbrace{w_j}_{\text{new value}} \leftarrow \underbrace{w_j}_{\text{old value}} + \eta \underbrace{\sum_{i=1}^{M} (y^{(i)} - g(x^{(i)}))x_j^{(i)}}_{\text{gradient}}$$

$$w \leftarrow w + \eta \sum_{i=1}^{M} (y^{(i)} - g(x^{(i)}))x^{(i)}$$

```python
def _get_gradient(self,X,y):
    # programming \sum_i (yi-g(xi))xi
    gradient = np.zeros(self.w_.shape) # set
    for (xi,yi) in zip(X,y):
        # the actual update inside of sum
        gradi = (yi - self.predict_proba(xi,
        # reshape to be column vector and ad
        gradient += gradi.reshape(self.w_.sh

    return gradient/float(len(y))
```

$$w \leftarrow w + \eta \left[\underbrace{\nabla l(w)_{old}}_{\text{old gradient}} - C \cdot 2w \right]$$

Demo Lecture

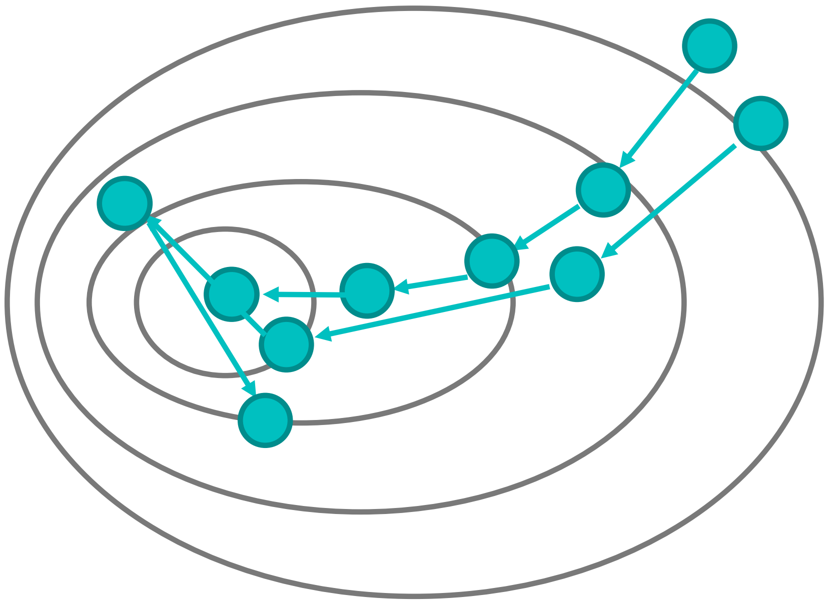
06. Optimization

Optimization: gradient descent

- What we know thus far:

$$\underbrace{w_j}_{\text{new value}} \leftarrow \underbrace{w_j}_{\text{old value}} + \eta \underbrace{\left[\left(\sum_{i=1}^{M} (y^{(i)} - g(x^{(i)}))x_j^{(i)} \right) - C \cdot 2w_j \right]}_{\nabla l(w)}$$

$$w \leftarrow w + \eta \nabla l(w)$$

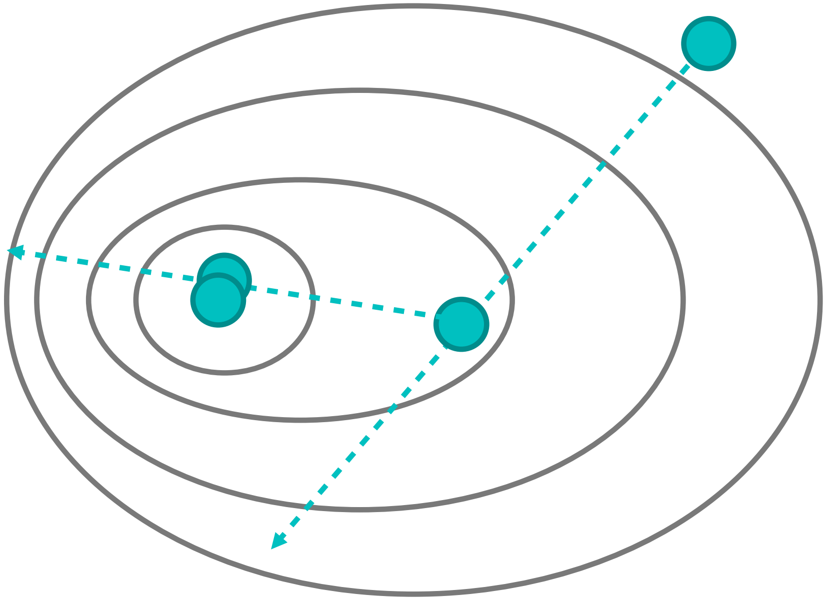


Line Search: a better method

- Line search in direction of gradient:

$$w \leftarrow w + \eta \nabla l(w)$$

$$w \leftarrow w + \underbrace{\eta}_{\text{best step?}} \nabla l(w)$$



Stochastic Methods

- How much computation is required (for gradient)?

$$\sum_{i=1}^{M}(y^{(i)} - \hat{y}^{(i)})x^{(i)} - 2C \cdot w$$

$$w \leftarrow w + \eta \underbrace{\left((y^{(i)} - \hat{y}^{(i)})x^{(i)} - 2C \cdot w\right)}_{\text{approx. gradient}},$$

i chosen at random