SMO and Stochastic SVM

Machine Learning
CSx824/ECEx242
Bert Huang
Virginia Tech

Outline

- Is SVM too slow?
- Fix #1: Sequential minimal optimization
- Fix #2: Stochastic gradient descent

QP Running Time

- Depends on algorithm, but most have $O(N^3)$ worst-case time
    - N = number of variables + number of constraints
- No good for “big data”
- Can we exploit known form of SVM QP?

Dual SVM

$$\min_{\alpha} \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i^\top x_j - \sum_i \alpha_i$$

$$\text{s.t.} \quad \sum_i \alpha_i y_i = 0, \quad \alpha_i \in [0, C]$$

$$w = \sum_i \alpha_i y_i x_i$$

$$b = y_i - \sum_j \alpha_j y_j x_j^\top x_i$$

for examples *i* where

$$0 < \alpha_i < C$$

Sequential Minimal Optimization

- Optimize two variables at a time
- Closed form updates

$$\min_{\substack{\cancel{\alpha} \\ \alpha_a,\, \alpha_b}} \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j \boxed{K_{ij}} - \sum_i \alpha_i$$

$$\text{s.t.} \quad \sum_i \alpha_i y_i = 0, \quad \alpha_i \in [0, C]$$

$$\min_{\alpha_a,\alpha_b} \quad \frac{1}{2}K_{aa}\alpha_a^2 + \frac{1}{2}K_{bb}\alpha_b^2 + \frac{1}{2}\alpha_a y_a \sum_{j \neq a} y_j \alpha_j K_{aj} + \frac{1}{2}\alpha_b y_b \sum_{j \neq b} y_j \alpha_j K_{bj} - \alpha_a - \alpha_b$$

$$y_a \alpha_a + y_b \alpha_b = -\sum_{i \neq a,b} \alpha_i y_i \qquad 0 \leq \alpha_a, \alpha_b \leq C$$

(Platt, 1998)

Hinge-Loss Primal SVM Form

Primal SVM

$$\min_{\substack{w \in \mathbb{R}^d \\ \xi \in [0,\infty]^n}} \frac{1}{2} w^\top w + C \sum_{i=1}^{n} \xi_i \quad \text{s.t.} \quad y_i(w^\top x_i + b) - 1 + \xi_i \geq 0 \ \ \forall i \in \{1, \dots, n\}$$

$$\min_{w \in \mathbb{R}^d} \frac{\lambda}{2} w^\top w + \frac{1}{n} \sum_{i=1}^{n} h(1 - y_i(w^\top x_i + b))$$

$$h(z) = \max\{0, z\}$$

$$\nabla_w = \lambda w - \frac{1}{n} \sum_{i=1}^{n} y_i x_i \boxed{I(y_i(w^\top x_i + b) < 1)} \longleftarrow \text{indicator function}$$

Stochastic SVM

(E.g., Shalev-Shwartz et al., '07)

$$\min_{w \in \mathbb{R}^d} \frac{\lambda}{2} w^\top w + \frac{1}{n} \sum_{i=1}^{n} h(1 - y_i(w^\top x_i + b))$$

$$\nabla_w = \lambda w - \frac{1}{n} \sum_{i=1}^{n} y_i x_i I(y_i(w^\top x_i + b) < 1)$$

$$= \lambda w - \mathbb{E}_{i \in \mathbb{U}} \left[y_i x_i I(y_i(w^\top x_i + b) < 1) \right]$$

$$w^t \leftarrow w^{t-1} + \underbrace{\frac{1}{t}}_{\text{step size}} \underbrace{\left(y_i x_i I(y_i(x_i^\top w^{t-1} + b) < 1) - \lambda w^{t-1} \right)}_{\text{negative \textbf{sub}gradient}} \quad \text{for random } i$$

…kinda like perceptron with a margin and regularization

Summary

- Both SMO and stochastic SVM training consider one or two examples at a time
- Dramatic speedups in practice
- Another fast SVM training method cutting-plane or active-set optimization
  - Hope to find only the active constraints (support vectors)
  - Greedily add constraints to the problem