THE UNIVERSITY OF TEXAS AT AUSTIN
Department of Statistics and Data Sciences
College of Natural Sciences

SDS 385: Stat Models for Big Data

Lecture 6: Support Vector Machines

Purnamrita Sarkar
Department of Statistics and Data Science
The University of Texas at Austin

https://psarkar.github.io/teaching

Support Vector Machines

- Given training data $(x_i, y_i)_{i=1}^n \in \mathbb{R}^p \times \{-1, 1\}$, we want to minimize:

$$\min_w \frac{w^T w}{2} + C \sum_i \max(0, 1 - y_i w^T x_i)$$

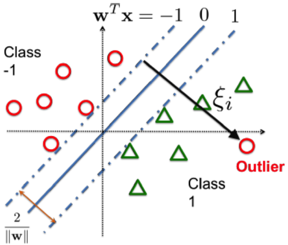

Figure 1: Courtesy Cho-Jui Hsieh's class

SGD for SVM

- Define:

$$f(w) = \frac{1}{n}\sum_i \left(\underbrace{\frac{w^T w}{2} + nC \max(0, 1 - y_i w^T x_i)}_{f_i(w)} \right)$$

- For $t = 1 \ldots$
  - Pick j uniformly at random.
  - Compute $\nabla f_j(w)$
  - Update $w = w - \eta_t \nabla f_j(w)$

SGD for SVM

- In this case, the hinge loss is not differentiable.
- A subgradient of the hinge loss $\max(0, 1 - y_i w^T x_i)$

$$\begin{cases} -y_i x_i & \text{if } 1 - y_i w^T x_i > 0 \\ 0 & \text{if } 1 - y_i w^T x_i < 0 \\ 0 & \text{if } 1 - y_i w^T x_i = 0 \end{cases}$$

SGD for SVM

- For $t = 1 \dots$
  - Pick j uniformly at random.
  - If $y_j w^T x_j > 0$
    - $w_{t+1} = w_t(1 - \eta_t) + \eta_t C n y_i x_i$
  - Else update $w_{t+1} = w_t(1 - \eta_t)$
    - If you store w as a scalar vector pair $w = \gamma v$, then just updating γ leads to $O(1)$ computation.
- This is in "Pegasos: primal estimated subgradient solver for SVM", ICML 2007, Shalev-Schwartz et al.

SVM: the dual problem

- The dual of SVM is given by:

$$\min_{\alpha} \frac{1}{2} \alpha^T Q \alpha - \sum_i \alpha_i$$

$$\text{s.t.} \alpha_i \in [0, C]$$

  Where $Q_{ij} = y_i y_j x_i^T x_j$.

- The primal solution can be written in terms of the dual solution as:

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

Stochastic Dual coordinate ascent

- For $t = 1 \ldots$
  - Compute
    $$\delta^* = \arg \min_{0 < \alpha_i + \delta < C} f(\alpha + \delta e_i)$$
  - Update $\alpha_i = \alpha_i + \delta^*$
  - Update $w = w + \delta^* y_i x_i$ (time complexity $O(nnz(x_i))$)
  - After convergence this gives $w^* = \sum_i \alpha_i^* y_i x_i$

Stochastic Dual coordinate ascent

- Consider the one variable problem:

$$f(\alpha + \delta e_i) = \frac{1}{2}(\alpha + \delta e_i)^T Q(\alpha + \delta e_i) - \sum_i \alpha_i - \delta$$
$$= \frac{1}{2}\alpha^T Q\alpha + \delta\alpha^T Q e_i + \delta^2 \frac{Q_{ii}}{2} - \sum_i \alpha_i - \delta$$

- Set the gradient to zero:

$$(Q\alpha)_i + Q_{ii}\delta^* - 1 = 0 \rightarrow \delta^* = \frac{1 - (Q\alpha)_i}{Q_{ii}}$$

- But we have the constraint $0 \leq \alpha_i + \delta \leq C$, so we have:

$$\alpha_i + \delta^* = \begin{cases} \alpha_i + \dfrac{1 - (Q\alpha)_i}{Q_{ii}} & \text{If } \alpha_i + \delta \in [0, C] \\ 0 & \text{If } \alpha_i + \delta < 0 \\ C & \text{If } \alpha_i + \delta > C \end{cases}$$

Fast computation

- Main computational bottleneck $Q\alpha$
- Write $Q = \underbrace{\mathrm{diag}(y)X}_{R}\underbrace{X^T\mathrm{diag}(y)}_{R^T}$
- Note that:

$$(Q\alpha)_i = R_i \underbrace{R^T\alpha}_{w} = y_i x_i^T w$$

- If you maintain w through the steps, computational complexity becomes $O(nnz(x_i))$

Acknowledgment

Cho-Jui Hsieh's class notes at UC Davis.