Newton update in L_2-norm random tree approximation

Hongyu Su

Helsinki Institute for Information Technology HIIT
Department of Computer Science
Aalto University

May 20, 2015

Preliminaries

- $\mathcal{X}$ is an arbitrary input space, $\mathbf{x} \in \mathcal{X}$.
- $\mathcal{Y}$ is an output space of a set of ℓ-dimensional *multilabels*

$$\mathbf{y} = (y_1, \cdots, y_\ell) \in \mathcal{Y}.$$

- y_i is a *microlabel* and $y_i \in \{1, \cdots, r_i\}, r_i \in \mathbb{Z}$.
- For example, multilabel binary classification $y_i \in \{-1, +1\}$.
- Training examples are sampled from $(\mathbf{x}, \mathbf{y}) \in \mathcal{X} \times \mathcal{Y}$.
- Each example $(\mathbf{x}, \mathbf{y})$ is mapped into a joint feature space $\boldsymbol{\phi}(\mathbf{x}, \mathbf{y})$.
- $\mathbf{w}$ is the weight vector in the joint feature space.
- Define a linear score function $F(\mathbf{w}, \mathbf{x}, \mathbf{y}) = \langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x}, \mathbf{y}) \rangle$.
- The prediction $\mathbf{y_w}(\mathbf{x})$ of an input $\mathbf{x}$ is the multilabel $\mathbf{y}$ that maximizes the score function

$$\mathbf{y_w}(\mathbf{x}) = \operatorname*{argmax}_{\mathbf{y} \in \mathcal{Y}} \langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x}, \mathbf{y}) \rangle. \tag{1}$$

- (1) is called *inference* problem which is $\mathcal{NP}$-hard for most output feature maps.

Markov network

- We assume that the output feature map ϕ is a potential function on a Markov network $G = (E, V)$.

- G is a complete graph with $|V| = \ell$ nodes and $|E| = \frac{\ell(\ell-1)}{2}$ undirected edges.

- $\varphi(\mathbf{x})$ is the input feature map, e.g., bag-of-words feature of an example $\mathbf{x}$.

- $\psi(\mathbf{y})$ is the output feature map which is a collection of edges and labels

$$\varphi(\mathbf{y}) = (u_e)_{e \in E}, u_e \in \{-1, +1\}^2.$$

- The joint feature is the Kronecker product of $\varphi(\mathbf{x})$ and $\psi(\mathbf{y})$

$$\phi(\mathbf{x}, \mathbf{y}) = (\phi_e(\mathbf{x}, \mathbf{y}))_{e \in E} = (\varphi(\mathbf{x}) \otimes \psi_e(\mathbf{y}_e))_{e \in E}.$$

- The score function is

$$F(\mathbf{w}, \mathbf{x}, \mathbf{y}) = \langle \mathbf{w}, \phi(\mathbf{x}, \mathbf{y}) \rangle = \sum_{e \in E} \langle \mathbf{w}_e, \phi_e(\mathbf{x}, \mathbf{y}_e) \rangle.$$

Inference in terms of all spanning trees

▶ Solving the following inference problem on a complete graph is $\mathcal{NP}$-hard

$$\mathbf{y_w}(\mathbf{x}) = \underset{\mathbf{y} \in \mathcal{Y}}{\operatorname{argmax}} F(\mathbf{w}, \mathbf{x}, \mathbf{y}) = \underset{\mathbf{y} \in \mathcal{Y}}{\operatorname{argmax}} \sum_{e \in E} \langle \mathbf{w}_e, \boldsymbol{\phi}_e(\mathbf{x}, \mathbf{y}_e) \rangle.$$

▶ For a complete graph, there are $\ell^{\ell-2}$ unique spanning trees.

▶ We can write $F(\mathbf{w}, \mathbf{x}, \mathbf{y})$ as a conic combination of all spanning trees

$$F(\mathbf{w}, \mathbf{x}, \mathbf{y}) = \underset{T \in U(G)}{\mathbf{E}} a_T \langle \mathbf{w}_T, \boldsymbol{\phi}_T(\mathbf{x}, \mathbf{y}) \rangle$$

$$\underset{T \in U(G)}{\mathbf{E}} a_T^2 = 1, \underset{T \in U(G)}{\mathbf{E}} a_T < 1.$$

▶ $U(G)$ is the uniform distribution over $\ell^{\ell-2}$ spanning trees.

▶ There is a exponential dependency on the number of spanning trees.

A sample of n spanning trees

- Instead of using all spanning trees, we can just use n spanning trees

$$F_{\mathcal{T}}(\mathbf{w}, \mathbf{x}, \mathbf{y}) = \frac{1}{n} \sum_{i=1}^{n} a_{T_i} \langle \mathbf{w}_{T_i}, \boldsymbol{\phi}_{T_i}(\mathbf{x}, \mathbf{y}) \rangle$$

$$\frac{1}{n} \sum_{i=1}^{n} a_{T_i}^2 = 1, \frac{1}{n} \sum_{i=1}^{n} a_{T_i} < 1.$$

- When

$$n \geq \frac{\ell^2}{\epsilon^2} \left(\frac{1}{16} + \frac{1}{2} \ln \frac{8\sqrt{n}}{\delta} \right),$$

with high probability, we have $|F_{\mathcal{T}}(\mathbf{w}, \mathbf{x}, \mathbf{y}) - F(\mathbf{w}, \mathbf{x}, \mathbf{y})| \leq \epsilon$.

- A sample of $n \in \Theta(\ell^2/\delta^2)$ random spanning tree is sufficient to estimate the score function.

- Margin achieved by $F(\mathbf{w}, \mathbf{x}, \mathbf{y})$ is also preserved by the sample of n random spanning trees $F_{\mathcal{T}}(\mathbf{w}, \mathbf{x}, \mathbf{y})$.

Optimization problem

▶ The primal optimization problem is defined as

$$\min_{\mathbf{w}_{T_i}, \xi_i} \quad \frac{1}{2} \sum_{i=1}^{n} ||\mathbf{w}_{T_i}||^2 + C \sum_{k=1}^{m} \xi_k$$

$$\text{s.t.} \quad \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \langle \mathbf{w}_{T_i}, \boldsymbol{\phi}_{T_t}(\mathbf{x}_k, \mathbf{y}_k) \rangle - \max_{\mathbf{y} \neq \mathbf{y}_k} \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \langle \mathbf{w}_{T_t}, \boldsymbol{\phi}_{T_i}(\mathbf{x}_k, \mathbf{y}) \rangle \geq 1 - \xi_k,$$

$$\xi_k \geq 0, \forall\, k \in \{1, \ldots, m\}.$$

▶ The marginalized dual problem is defined as

$$\max_{\mu \in \mathcal{M}} \quad \sum_{i=1}^{n} \left(\mu_{T_i} \ell_{T_i} - \frac{1}{2} \mu_{T_i} K_{T_i} \mu_{T_i} \right)$$

$$\text{s.t.} \quad \sum_{u_e} \mu_{T_i, e}(u_e) \leq C.$$

Optimization algorithm for a single spanning tree

- We can solve the optimization problem efficiently for each individual spanning tree.
- The algorithm iterates over all training example until convergence.
- For the kth iteration:
  1. Obtain the solution of the jth example in kth iteration $\mu_{T_i}^k(j)$.
  2. Compute the gradient $g_{T_i}^k(j) = \ell_{T_i}(j) - K_{T_i}\mu_{T_i}^k(j)$.
  3. Compute the update direction

  $$\hat{\mu}_{T_i}^{k+1}(j) = \underset{\mu \in \mathcal{M}}{\textbf{argmax}}\, \mu^{\mathsf{T}} g_{T_i}^k(j).$$

  4. Compute the difference $\Delta\mu_{T_i}^{k+1}(j) = \hat{\mu}_{T_i}^{k+1}(j) - \hat{\mu}_{T_i}^k(j)$.
  5. Perform the update $\mu_{T_i}^{k+1}(j) = \mu_{T_i}^k(j) + \tau\Delta\mu_{T_i}^{k+1}(j)$
- The step size along the update direction τ is given by the exact line search.

$$\frac{\partial\left(f(\mu_{T_i}^{k+1}(j)) - f(\mu_{T_i}^k(j))\right)}{\partial \tau} = 0, 0 \leq \tau \leq 1.$$

κ-best inference for a collection of n spanning trees

- ▶ The algorithm iterates over all training example until convergence.
- ▶ For the kth iteration:
    1. Obtain the solutions of the jth example over all trees $(\mu_{T_i}^k(j))_{i=1}^n$.
    2. Compute the gradients over all trees $(g_{T_i}^k(j))_{i=1}^n$.
    3. Compute the update directions

    $$\hat{\mu}_{T_i}^{k+1}(j) = \operatorname*{argmax}_{\mu \in \mathcal{M}} \mu^\mathsf{T} g_{T_i}^k(j), \; \forall i.$$

    4. Compute the best direction

    $$\tilde{\mu}_{T_i}^{k+1}(j) = \operatorname*{argmax}_{\mu \in (\hat{\mu}_{T_i}^{k+1}(j))_{i=1}^n} \sum_{i=1}^n \mu^\mathsf{T} g_{T_i}^k(j)$$

    5. Compute the difference $\Delta \mu_{T_i}^{k+1}(j) = \hat{\mu}_{T_i}^{k+1}(j) - \hat{\mu}_{T_i}^k(j), \; \forall i.$
    6. Perform the update $\mu_{T_i}^{k+1}(j) = \mu_{T_i}^k(j) + \tau \Delta \mu_{T_i}^{k+1}(j), \; \forall i.$
- ▶ The step size along the update direction τ is given by the exact line search.

$$\frac{\partial \left(\sum_{i=1}^n f(\mu_{T_i}^{k+1}(j)) - \sum_{i=1}^n f(\mu_{T_i}^k(j)) \right)}{\partial \tau} = 0, 0 \leq \tau \leq 1.$$

Conclusions

►