



Aalto University
School of Science
and Technology

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*

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

- ▶ y_i is a *microlabel* and $y_i \in \{1, \dots, r_i\}$, $r_i \in \mathbb{Z}$.
- ▶ 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 $\phi(\mathbf{x}, \mathbf{y})$.
- ▶ \mathbf{w} is the weight vector in the joint feature space.
- ▶ Define a linear score function $F(\mathbf{w}, \mathbf{x}, \mathbf{y}) = \langle \mathbf{w}, \phi(\mathbf{x}, \mathbf{y}) \rangle$.
- ▶ The prediction $\mathbf{y}_{\mathbf{w}}(\mathbf{x})$ of an input \mathbf{x} is the multilabel \mathbf{y} that maximizes the score function

$$\mathbf{y}_{\mathbf{w}}(\mathbf{x}) = \underset{\mathbf{y} \in \mathcal{Y}}{\operatorname{argmax}} \langle \mathbf{w}, \phi(\mathbf{x}, \mathbf{y}) \rangle. \quad (1)$$

- ▶ (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 spanning trees

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

$$\mathbf{y}_{\mathbf{w}}(\mathbf{x}) = \operatorname{argmax}_{\mathbf{y} \in \mathcal{Y}} F(\mathbf{w}, \mathbf{x}, \mathbf{y}) = \operatorname{argmax}_{\mathbf{y} \in \mathcal{Y}} \sum_{e \in E} \langle \mathbf{w}_e, \phi_e(\mathbf{x}, \mathbf{y}_e) \rangle. \quad (2)$$

- ▶ 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

$$F(\mathbf{w}, \mathbf{x}, \mathbf{y}) = \sum_{T \in U(G)} \mathbf{E} a_T \langle \mathbf{w}_T, \phi_T(\mathbf{x}, \mathbf{y}) \rangle$$
$$\sum_{T \in U(G)} \mathbf{E} a_T^2 = 1, \quad \sum_{T \in U(G)} \mathbf{E} a_T < 1.$$

Multilabel classification

- ▶ Multilabel classification is an important research field in machine learning.
 - ▶ For example, a document can be classified as “science”, “genomics”, and “drug discovery”.
 - ▶ Each input variable $\mathbf{x} \in \mathcal{X}$ is associated with multiple output variables $\mathbf{y} \in \mathcal{Y}$, $\mathcal{Y} = \mathcal{Y}_1 \times \cdots \times \mathcal{Y}_l$, $\mathcal{Y}_i = \{+1, -1\}$.
 - ▶ The goal is to find a mapping function that predicts the best values of an output given an input $f \in \mathcal{H} : \mathcal{X} \rightarrow \mathcal{Y}$.
- ▶ The central problems of multilabel classification:
 - ▶ The size of the output space \mathcal{Y} is exponential in the number of microlabels.
 - ▶ The dependency of microlabels needs to be exploited to improve the prediction performance.

Structured output learning

- ▶ There is an *output graph* connecting multiple labels.
 - ▶ A set of nodes represents multiple labels.
 - ▶ A set of edges represents the correlation between labels.
- ▶ Hierarchical classification:
 - ▶ The output graph is a rooted tree or a directed graph defining different levels of granularities.
 - ▶ For example, SSVM, ...
- ▶ Graph labeling:
 - ▶ The output graph often takes a general form (e.g., a tree, a chain).
 - ▶ For example, M^3N , CRF, MMCRF, ...
- ▶ The output graph is assumed to be known *a priori*.

Research question

- ▶ The output graph is hidden in many applications.
 - ▶ For example, a surveillance photo can be tagged with “building”, “road”, “pedestrian”, and “vehicle”.
- ▶ We study the problem in structured output learning when the output graph is not observed.
- ▶ In particular:
 - ▶ Assume the dependency can be expressed by a complete set of pairwise correlations.
 - ▶ Build a structured output learning model with a complete graph as the output graph.
 - ▶ Solve the optimization problem and the inference problem (\mathcal{NP} -hard).

Today

- ▶ A structured prediction model which performs max-margin learning on a random sample of spanning tree.
- ▶ Two ways to combine the set of random spanning trees
 - ▶ conical combination in NIPS paper.
 - ▶ convex combination as future work.
- ▶ Derivations and the corresponding optimization problems.

Model

- ▶ Training examples comes in pair $S = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^m \in \mathcal{X} \times \mathcal{Y}$.
- ▶ A complete graph $G = (E, V)$ is used as the output graph.
- ▶ $\varphi(\mathbf{x})$ is the input feature map, e.g., a feature vector of d dimension.
- ▶ $\Gamma_G(\mathbf{y})$ is the output feature map of \mathbf{y} on G of $4 \times |E|$ dimension

$$\begin{aligned}\Gamma_G(\mathbf{y}) &= \{\Gamma_e(\mathbf{y}_e)\}_{e \in G}, \\ \Gamma_e(\mathbf{y}_e) &= [\mathbf{1}_{\mathbf{y}_e=00}, \mathbf{1}_{\mathbf{y}_e=01}, \mathbf{1}_{\mathbf{y}_e=10}, \mathbf{1}_{\mathbf{y}_e=11}]\end{aligned}$$

- ▶ A joint feature map of $(\mathbf{x}_i, \mathbf{y}_i)$

$$\phi_G(\mathbf{x}_i, \mathbf{y}_i) = \varphi(\mathbf{x}_i) \otimes \Gamma_G(\mathbf{y}_i) = \{\phi_e(x_i, \mathbf{y}_{i,e})\}_{e \in G}.$$

- ▶ A compatibility score is defined as

$$F(\mathbf{x}, \mathbf{y}; \mathbf{w}_G) = \langle \mathbf{w}_G, \phi_G(\mathbf{x}, \mathbf{y}) \rangle = \sum_{e \in G} \langle \mathbf{w}_{G,e}, \phi_e(\mathbf{x}, \mathbf{y}_e) \rangle$$

Model (cont.)

- ▶ \mathbf{w} ensures an input \mathbf{x}_i with a correct multilabel \mathbf{y}_i achieves a higher score than with any incorrect multilabel $\mathbf{y} \in \mathcal{Y}$.
- ▶ The predicted output $\mathbf{y}(\mathbf{x})$ for a given input \mathbf{x} is computed by

$$\mathbf{y}(\mathbf{x}) = \operatorname{argmax}_{\mathbf{y} \in \mathcal{Y}} F(\mathbf{x}, \mathbf{y}; \mathbf{w}_G) = \operatorname{argmax}_{\mathbf{y} \in \mathcal{Y}} \sum_{e \in G} \langle \mathbf{w}_{G,e}, \phi_{G,e}(\mathbf{x}, \mathbf{y}_e) \rangle,$$

which is called *inference problem*.

- ▶ The inference problem is \mathcal{NP} -hard for most joint feature maps on the complete graph.

How to learn w on a complete graph?

- ▶ The *margin* of an example \mathbf{x}_i is

$$\gamma_G(\mathbf{x}_i; \mathbf{w}_G) = F(\mathbf{x}_i, \mathbf{y}_i; \mathbf{w}_G) - \max_{\mathbf{y} \in \mathcal{Y}/\mathbf{y}_i} F(\mathbf{x}_i, \mathbf{y}; \mathbf{w}_G).$$

- ▶ \mathbf{w} is solved by *max-margin principle* which aims to maximize $\gamma(\mathbf{x}_i; \mathbf{w}_G)$ over all training example $\mathbf{x}_i, i \in \{1, \dots, m\}$.
- ▶ The inference problem on a complete graph is \mathcal{NP} -hardness.
- ▶ The parameter space is quadratic in the number of microlabels k .
- ▶ We aim to use a joint feature map that allows the inference problem be solved in polynomial time.

Superposition of random trees

- ▶ $S(G)$ is a complete set of spanning tree generate from G , $|S(G)| = \ell^{\ell-2}$.
- ▶ Recall
 $\phi_G(\mathbf{x}, \mathbf{y}) = \{\phi_{G,e}(\mathbf{x}, \mathbf{y}_e)\}_{e \in G}$, $\mathbf{w}_G = \{\mathbf{w}_{G,e}\}_{e \in G}$, $\|\phi_G(\mathbf{x}, \mathbf{y})\| = \|\mathbf{w}_G\| = 1$.
- ▶ $\phi_T(\mathbf{x}, \mathbf{y}) = \{\phi_e(\mathbf{x}, \mathbf{y})\}_{e \in T}$ is the projection of $\phi_G(\mathbf{x}, \mathbf{y})$ on $T \in S(G)$.
- ▶ $\mathbf{w}_T = \{\mathbf{w}_{G,e}\}_{e \in T}$ is the projection of \mathbf{w}_G on $T \in S(G)$.
- ▶ Rewrite

$$\begin{aligned} F(\mathbf{x}, \mathbf{y}, \mathbf{w}_G) &= \sum_{e \in G} \langle \mathbf{w}_{G,e}, \phi_{G,e}(\mathbf{x}, \mathbf{y}_e) \rangle \\ &= \frac{1}{\ell^{\ell-2}} \sum_{T \in S(G)} \sqrt{\frac{\ell}{2}} \langle \mathbf{w}_T, \phi_T(\mathbf{x}, \mathbf{y}_e) \rangle \\ &= \frac{1}{n} \sum_{i=1}^n a_{T_i} \langle \hat{\mathbf{w}}_{T_i}, \hat{\phi}_{T_i}(\mathbf{x}, \mathbf{y}_e) \rangle, \end{aligned}$$

$$\|\hat{\phi}_T(\mathbf{x}, \mathbf{y})\| = \|\hat{\mathbf{w}}_T\| = 1, \frac{1}{n} \sum_{i=1}^n a_{T_i}^2 = 1, \frac{1}{n} \sum_{i=1}^n a_{T_i} \leq 1, a_{T_i} \geq 0, n = \ell^{\ell-2}.$$

How many trees?

- ▶ If there is a predictor \mathbf{w}_G on complete graph achieves a margin on some training data, with high probability we need n spanning tree predictors $\{\mathbf{w}_{T_i}\}_{i=1}^n$ to achieve a close margin. n is quadratic in terms of ℓ .
- ▶ Recall

$$F(\mathbf{x}, \mathbf{y}, \mathbf{w}_T) = \frac{1}{n} \sum_{i=1}^n a_{T_i} \underbrace{\langle \hat{\mathbf{w}}_{T_i}, \hat{\phi}_{T_i}(\mathbf{x}, \mathbf{y}_e) \rangle}_{F(\mathbf{x}, \mathbf{y}, \mathbf{w}_{T_i})},$$

$$\|\hat{\phi}_T(\mathbf{x}, \mathbf{y})\| = \|\hat{\mathbf{w}}_T\| = 1, \frac{1}{n} \sum_{i=1}^n a_{T_i}^2 = 1, \frac{1}{n} \sum_{i=1}^n a_{T_i} \leq 1, a_{T_i} \geq 0, \cancel{n = \ell^2}.$$

Conical combination

- ▶ A sample $\mathcal{T} = \{T_1, \dots, T_n\}$ of n spanning trees drawn from G .
- ▶ Normalized feature vectors $\hat{\phi}_{T_i}(\mathbf{x}, \mathbf{y}) = \frac{\phi_{T_i}(\mathbf{x}, \mathbf{y})}{\|\phi_{T_i}(\mathbf{x}, \mathbf{y})\|}$, $T_i \in \mathcal{T}$.
- ▶ Normalized feature weights $\hat{\mathbf{w}}_{T_i} = \frac{\mathbf{w}_{T_i}}{\|\mathbf{w}_{T_i}\|}$, $T_i \in \mathcal{T}$.
- ▶ Conical combination of spanning trees

$$F(\mathbf{x}, \mathbf{y}, \mathbf{w}_{\mathcal{T}}) = \frac{1}{\sqrt{n}} \sum_{i=1}^n q_i \underbrace{\langle \hat{\mathbf{w}}_{T_i}, \hat{\phi}_{T_i}(\mathbf{x}, \mathbf{y}) \rangle}_{F(\mathbf{x}, \mathbf{y}, \mathbf{w}_{T_i})}$$

$$\sum_{i=1}^n q_i^2 = 1, q_i \geq 0, \forall i \in \{1, \dots, n\}.$$

Conical combination (cont.)

- To solve $\{\mathbf{w}_{T_i}\}_{T_i \in \mathcal{T}}$, we need to work on the optimization problem

$$\begin{aligned} \min_{\xi, \gamma, \mathbf{q}, \mathcal{W}} \quad & \frac{1}{2\gamma^2} + \frac{C}{\gamma} \sum_{k=1}^m \xi_k \\ \text{s.t.} \quad & \frac{1}{\sqrt{n}} \sum_{i=1}^n q_i \langle \hat{\mathbf{w}}_{T_i}, \hat{\phi}_{T_i}(\mathbf{x}_k, \mathbf{y}_k) \rangle - \max_{\mathbf{y} \in \mathcal{Y}} \frac{1}{\sqrt{n}} \sum_{i=1}^n q_i \langle \hat{\mathbf{w}}_{T_i}, \hat{\phi}_{T_i}(\mathbf{x}_k, \mathbf{y}) \rangle \\ & \geq \gamma - \xi_k, \xi_k \geq 0, \forall k \in \{1, \dots, m\}, \sum_{i=1}^n q_i^2 = 1, q_i \geq 0, \forall i \in \{1, \dots, n\}. \end{aligned}$$

- This is equivalent to

$$\begin{aligned} \min_{\mathbf{w}_{T_i}, \xi_i} \quad & \frac{1}{2} \sum_{i=1}^n \|\mathbf{w}_{T_i}\|^2 + C \sum_{k=1}^m \xi_k \\ \text{s.t.} \quad & \frac{1}{\sqrt{n}} \sum_{i=1}^n \langle \mathbf{w}_{T_i}, \phi_{T_i}(\mathbf{x}_k, \mathbf{y}_k) \rangle - \max_{\mathbf{y} \neq \mathbf{y}_k} \frac{1}{\sqrt{n}} \sum_{i=1}^n \langle \mathbf{w}_{T_i}, \phi_{T_i}(\mathbf{x}_k, \mathbf{y}) \rangle \geq 1 - \xi_k, \\ & \xi_k \geq 0, \forall k \in \{1, \dots, m\}. \end{aligned}$$

Inference Problem

- ▶ The inference problem of RTA is defined as finding the multilabel $\mathbf{y}_{\mathcal{T}}(\mathbf{x})$ that maximizes the sum of scores over a collection of trees

$$\mathbf{y}_{\mathcal{T}}(\mathbf{x}) = \operatorname{argmax}_{\mathbf{y} \in \mathcal{Y}} F_{\mathcal{T}}(\mathbf{x}, \mathbf{y}; \mathbf{w}_{\mathcal{T}}) = \operatorname{argmax}_{\mathbf{y} \in \mathcal{Y}} \sum_{t=1}^n \langle \mathbf{w}_{T_t}, \phi_{T_t}(\mathbf{x}, \mathbf{y}) \rangle.$$

- ▶ The inference problem on each individual spanning tree can be solve efficiently in $\Theta(l)$ by *dynamic programming*

$$\mathbf{y}_{T_t}(\mathbf{x}) = \operatorname{argmax}_{\mathbf{y} \in \mathcal{Y}} F_{T_t}(\mathbf{x}, \mathbf{y}; \mathbf{w}_{T_t}) = \operatorname{argmax}_{\mathbf{y} \in \mathcal{Y}} \langle \mathbf{w}_{T_t}, \phi_{T_t}(\mathbf{x}, \mathbf{y}) \rangle.$$

- ▶ There is no guarantee that there exists a tree $T_t \in \mathcal{T}$ in which the maximizer of F_{T_t} is the maximizer of $F_{\mathcal{T}}$.

Fast Inference Over a Collection of Trees

- ▶ For each tree T_t , instead of computing the best multilabel \mathbf{y}_{T_t} , we compute K -best multilabels in $\Theta(KI)$ time

$$\mathcal{Y}_{T_t, K} = \{\mathbf{y}_{T_t, 1}, \dots, \mathbf{y}_{T_t, K}\}.$$

- ▶ Performing the same computation on all trees gives a candidate list of $n \times K$ multilabels in $\Theta(nKI)$ time

$$\mathcal{Y}_{\mathcal{T}, K} = \mathcal{Y}_{T_1, K} \cup \dots \mathcal{Y}_{T_n, K}.$$

- ▶ For now, we assume the best scoring multilabel of a collection of trees exists in the list $\mathcal{Y}_{\mathcal{T}, K}$.
- ▶ We proved that with a high probability $\mathbf{y}_{\mathcal{T}}$ will appear in $\mathcal{Y}_{\mathcal{T}, K}$.
- ▶ We can identify $\mathbf{y}_{\mathcal{T}}$ from $\mathcal{Y}_{\mathcal{T}, K}$.

Convex combination

- ▶ A sample \mathcal{T} of n spanning trees drawn from G .
- ▶ Normalized feature weights $\hat{\mathbf{w}}_{T_i} = \frac{\mathbf{w}_{T_i}}{\|\mathbf{w}_{T_i}\|}$, $T_i \in \mathcal{T}$.
- ▶ Normalized feature vectors $\hat{\phi}_{T_i}(\mathbf{x}, \mathbf{y}) = \frac{\phi_{T_i}(\mathbf{x}, \mathbf{y})}{\|\phi_{T_i}(\mathbf{x}, \mathbf{y})\|}$, $T_i \in \mathcal{T}$.
- ▶ Convex combination of spanning trees

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

$$\sum_{i=1}^n q_i = 1, q_i \geq 0, \forall i \in \{1, \dots, n\}.$$

Convex combination (cont.)

- To solve $\{\mathbf{w}_{T_i}\}_{T_i \in \mathcal{T}}$, we need to work on the optimization problem

$$\begin{aligned} \min_{\xi, \gamma, \mathbf{q}, \mathcal{W}} \quad & \frac{1}{2\gamma^2} + \frac{C}{\gamma} \sum_{k=1}^m \xi_k \\ \text{s.t.} \quad & \frac{1}{n} \sum_{i=1}^n q_i \langle \hat{\mathbf{w}}_{T_i}, \hat{\phi}_{T_i}(\mathbf{x}_k, \mathbf{y}_k) \rangle - \max_{\mathbf{y} \in \mathcal{Y}} \frac{1}{n} \sum_{i=1}^n q_i \langle \hat{\mathbf{w}}_{T_i}, \hat{\phi}_{T_i}(\mathbf{x}_k, \mathbf{y}) \rangle \\ & \geq \gamma - \xi_k, \xi_k \geq 0, \forall k \in \{1, \dots, m\}, \sum_{i=1}^n q_i = 1, q_i \geq 0, \forall i \in \{1, \dots, n\}. \end{aligned}$$

- This is equivalent to

$$\begin{aligned} \min_{\mathbf{w}_{T_i}, \xi_i} \quad & \frac{1}{2} \left(\sum_{i=1}^n \|\mathbf{w}_{T_i}\| \right)^2 + C \sum_{k=1}^m \xi_k \\ \text{s.t.} \quad & \frac{1}{n} \sum_{i=1}^n \langle \mathbf{w}_{T_i}, \phi_{T_i}(\mathbf{x}_k, \mathbf{y}_k) \rangle - \max_{\mathbf{y} \neq \mathbf{y}_k} \frac{1}{n} \sum_{i=1}^n \langle \mathbf{w}_{T_i}, \phi_{T_i}(\mathbf{x}_k, \mathbf{y}) \rangle \geq 1 - \xi_k, \\ & \xi_k \geq 0, \forall k \in \{1, \dots, m\}. \end{aligned}$$

Convex combination (cont.)

- This can be expressed equivalently as

$$\begin{aligned} \min_{\mathbf{w}_{T_i}, \xi_i, \lambda_i} \quad & \frac{1}{2} \sum_{i=1}^n \frac{1}{\lambda_i} \|\mathbf{w}_{T_i}\|^2 + C \sum_{k=1}^m \xi_k \\ \text{s.t.} \quad & \frac{1}{n} \sum_{i=1}^n \langle \mathbf{w}_{T_i}, \phi_{T_i}(\mathbf{x}_k, \mathbf{y}_k) \rangle - \max_{\mathbf{y} \neq \mathbf{y}_k} \frac{1}{n} \sum_{i=1}^n \langle \mathbf{w}_{T_i}, \phi_{T_i}(\mathbf{x}_k, \mathbf{y}) \rangle \geq 1 - \xi_k, \\ & \xi_k \geq 0, \forall k \in \{1, \dots, m\}, \sum_{i=1}^n \lambda_i = 1, \lambda_i \geq 0, \forall i \in \{1, \dots, n\}. \end{aligned}$$

Conclusions

- ▶ We show that if there is a learner \mathbf{w}_G defined on a complete graph achieves a margin on some training data, then with a random collection of spanning tree learners $\{\mathbf{w}_{T_i}\}_{i=1}^n$ we can achieve a similar margin with high probability. Besides, n is polynomial in k .
- ▶ We propose two methods to combine the random collection of trees, namely, convex combination and conical combination.