Structured Prediction

Ningshan Zhang

Advanced Machine Learning, Spring 2016

Outline

- Ensemble Methods for Structured Prediction[1]
  - On-line learning
  - Boosting
- A Generalized Kernel Approach to Structured Output Learning[2]

Structured Prediction Problem

- Structured Output: $\mathcal{Y} = \mathcal{Y}_1 \times \ldots \times \mathcal{Y}_l$.
- Loss Function: $L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ decomposable.
- Training data: sample drawn i.i.d. from $\mathcal{X} \times \mathcal{Y}$ according to some distribution $\mathcal{D}$,

$$S = \left(\left(\boldsymbol{x}_1, \boldsymbol{y}_1\right), \ldots, \left(\boldsymbol{x}_m, \boldsymbol{y}_m\right)\right) \in \mathcal{X} \times \mathcal{Y}$$

- Problem: find hypothesis $h : \mathcal{X} \to \mathcal{Y}$ with small generalization error

$$\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[L\left(h\left(x\right), y\right)\right]$$

Ensemble Methods: Further assumptions

- The number of substructures $l \geq 1$ is fixed.
- Loss function L can be decomposed as a sum of loss functions $l_k : \mathcal{Y}_k \to \mathbb{R}+$, i.e.
  - for all $\boldsymbol{y} = \left(y^1, ..., y^l\right)$ and $\boldsymbol{y}' = \left(y'^1, ..., y'^l\right)$,

$$L\left(\boldsymbol{y}, \boldsymbol{y}'\right) = \sum_{k=1}^{l} l_k \left(y^k, y'^k\right)$$

- A set of structured prediction experts $h_1, ..., h_p$:

$$h_j\left(x\right) = \left(h_j^1\left(x\right), ..., h_j^l\left(x\right)\right), \forall j = 1, ..., p$$

Path Experts

- Intuition: one particular expert maybe better at predicting k-th substructure, but not elsewhere.

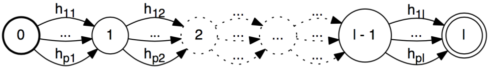


Figure: Path Expert

- Construct a larger set of experts, 'path experts':

$$\mathcal{H} = \left\{ \left(h^1_{j_1}, ..., h^l_{j_l} \right), j_k \in \{1, ..., p\}, k = 1, ..., l \right\}, |\mathcal{H}| = p^l$$

- When selecting the best h^* from $\mathcal{H}$, it may not coincide with any of original experts.

Review: Randomized Weighted Majority (WM)

- Task: find a distribution over experts.
- Initialize weights for all experts:

$$w_{1,j} = 1/p, j = 1, ..., p$$

- After receiving sample (x_t, y_t),
  - Update weight for expert $j, j = 1, .., p$:

$$w_{t+1,j} = \frac{w_{t,j} e^{-\eta L\left(\hat{y}_{t,j}, y_t\right)}}{\sum_{j=1}^{p} w_{t,j} e^{-\eta L\left(\hat{y}_{t,j}, y_t\right)}}, \eta > 0 \text{ constant}$$

- Return $\boldsymbol{w}_{T+1} = (w_{T+1,1}, ..., w_{T+1,p})$

Randomized Weighted Majority (WM)

- Apply the Weighted Majority algorithm to path experts set $\mathcal{H}$:

$$\forall h \in \mathcal{H}, w_{t+1}(h) = \frac{w_t(h)\, e^{-\eta L(h(x_t), y_t)}}{\sum_{h \in \mathcal{H}} w_t(h)\, e^{-\eta L(h(x_t), y_t)}}$$

- p^l updates per round!
- Using the structure of $\mathcal{Y}$ and additive loss assumption, we can reduce to pl updates per round.

Weighted Majority Weight Pushing (WMWP)

- For every $h = \left(h_{j_1}^1, \ldots, h_{j_l}^l\right)$, $w(h) = \prod_{k=1}^{l} w_{j_k}^k$
- Enforce the weights at each substructure sum to one:

$$\sum_{j=1}^{p} w_j^k = 1, \forall k \in \{1, \ldots, l\}$$

- The overall weights $\sum_{h \in \mathcal{H}} w(h)$ will automatically sum to one! Example: $p = 2, l = 2$,

$$w_1^1 w_1^2 + w_1^1 w_2^2 + w_2^1 w_1^2 + w_2^1 w_2^2 = \left(w_1^1 + w_2^1\right)\left(w_1^2 + w_2^2\right) = 1$$

Weighted Majority Weight Pushing (WMWP)

- Initialize weights: $w_{1,j}^{k} = 1/p$, $\forall\, (k, j) \in \{1, ..., l\} \times \{1, ..., p\}$,
- At round t, $\forall\, (k, j) \in \{1, ..., l\} \times \{1, ..., p\}$, update weight

$$w_{t+1,j}^{k} = \frac{w_{t,j}^{k} e^{-\eta l_k \left(h_j^k(x_t), y_t^k \right)}}{\sum_{j=1}^{p} w_{t,j}^{k} e^{-\eta l_k \left(h_j^k(x_t), y_t^k \right)}},$$

- Return $W_1, ..., W_{T+1}$, where $W_t = \left(w_{t,j}^{k} \right)$.

On-line to batch conversion

- How to define a hypothesis based on $\{W_1, ..., W_{T+1}\}$?
- Two steps:
  1. Choose a good collection of distributions. $P = \{W_1, ..., W_{T+1}\}$ is one choice, but not necessarily the best.
  2. Use P to define hypotheses for prediction.

On-line to batch conversion

- Step 1: Choose a good collection of distributions.
- For any collection P, define score

$$\Gamma(P) = \frac{1}{|P|} \sum_{W_t \in P} \sum_{h \in \mathcal{H}} W_t(h) L(h(x_t), y_t) + M\sqrt{\frac{\log \frac{1}{\delta}}{|P|}}$$

- Chose $P_\delta = \arg\min_{P \in \mathcal{P}} \Gamma(P)$

On-line to batch conversion

- Step 2: define hypotheses.
- Randomized: randomly select a path expert $h \in \mathcal{H}$ according to

$$p(h) = \frac{1}{|P_\delta|} \sum_{W_t \in P_\delta} W_t(h),$$

denote the set of generated hypotheses as $\mathcal{H}_{Rand}$.

- Deterministic: defined a scoring function

$$\tilde{h}_{MVote}(x, y) = \prod_{k=1}^{l} \left(\frac{1}{|P_\delta|} \sum_{W_t \in P_\delta} \sum_{j=1}^{p} w_{t,j}^k 1_{h_j^k(x)=y^k} \right)$$

$$\mathcal{H}_{MVote}(x) = \arg\max_{y \in \mathcal{Y}} \tilde{h}_{MVote}(x, y)$$

Learning Guarantees

- Regret R_T

$$R_T = \sum_{t=1}^{T} \mathbb{E}_{h \sim W_t} \left[L\left(h\left(x_t\right), y_t\right) \right] - \inf_{h \in \mathcal{H}} \sum_{t=1}^{T} L\left(h\left(x_t\right), y_t\right)$$

- For any $\delta > 0$, with probability at least $1 - \delta$ over sample $(x_i, y_i)_{i=1}^{T}$ drawn i.i.d from $\mathcal{D}$, the following hold:

$$\mathbb{E}\left[L\left(\mathcal{H}_{Rand}\left(x\right), y\right)\right] \leq \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_{h \sim W_t} \left[L\left(h\left(x_t\right), y_t\right) \right] + M \sqrt{\frac{\log \frac{T}{\delta}}{T}},$$

$$\mathbb{E}\left[L\left(\mathcal{H}_{Rand}\left(x\right), y\right)\right] \leq \inf_{h \in \mathcal{H}} \mathbb{E}\left[L\left(h\left(x\right), y\right)\right] + \frac{R_T}{T} + 2M \sqrt{\frac{\log \frac{2T}{\delta}}{T}}.$$

Learning Guarantees

- The following inequality always hold: with expectations taken over $(x, y) \sim \mathcal{D}$ and $h \sim p$ for $\mathcal{H}_{Rand}$,

$$\mathbb{E}\left[L_{Ham}\left(\mathcal{H}_{MVote}\left(x\right), y\right)\right] \leq 2\,\mathbb{E}\left[L_{Ham}\left(\mathcal{H}_{Rand}\left(x\right), y\right)\right]$$

- Proof: by definition of $\mathcal{H}_{MVote}$, the following always hold:

$$\frac{1}{2} 1_{\mathcal{H}^k_{MVote}(x) \neq y^k} \leq \frac{1}{|P_\delta|} \sum_{W_t \in P_\delta} \sum_{j=1}^{p} w^k_{t,j} 1_{h^k_j(x) \neq y^k}$$

summing over k and take expectations over $\mathcal{D}$ yields the desired results. $\square$

AdaBoost: Review

- Hypothesis space: $\mathcal{H} = \left\{ \sum_{j=1}^{N} \alpha_j h_j, \alpha_j \geq 0 \right\}$, where $h_j \subset \mathcal{H}_0$ are base classifiers, $h_j : \mathcal{X} \to \mathbb{R}$.
- Objective Function: $F(\boldsymbol{\alpha}) = \sum_{i=1}^{m} e^{-y_i \sum_{j=1}^{N} \alpha_j h_j(x_i)}$
- Apply Coordinate Descent to $F(\boldsymbol{\alpha})$
- Return hypothesis: $h(x) = \mathrm{sgn}\left(\sum_{j=1}^{N} \alpha_j h_j(x) \right)$

ESPBoost: hypothesis space

- How to make a convex combination of path experts? Prediction $\rightarrow$ Score $\rightarrow$ Convex Combination of Score $\rightarrow$ 'Combined' Prediction
- For each path experts $h_t \in \mathcal{H}$, define the score $\tilde{h}_t : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}_+$:

$$\forall (\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{X} \times \mathcal{Y}, \tilde{h}_t(\boldsymbol{x}, \boldsymbol{y}) = \sum_{k=1}^{l} 1_{h_t^k(x)=y^k}$$

- Convex combination of score:

$$\left\{ \tilde{h} = \sum_{t=1}^{T} \alpha_t \tilde{h}_t : \tilde{h}_t \text{ derived from path experts}, \alpha_t \geq 0 \right\}$$

- 'Combined' Prediction:

$$h(\boldsymbol{x}) = \arg\max_{\boldsymbol{y} \in \mathcal{Y}} \tilde{h}(\boldsymbol{x}, \boldsymbol{y}) = \arg\max_{\boldsymbol{y} \in \mathcal{Y}} \sum_{t=1}^{T} \sum_{k=1}^{l} \alpha_t 1_{h_t^k(x)=y^k}$$

Loss Function and Upper Bound

- Normalized Hamming Loss:

$$\frac{1}{m}\sum_{i=1}^{m}\left[L_{Ham}\left(\mathcal{H}_{ESPBoost}\left(x_i\right), y_i\right)\right]$$

$$=\frac{1}{ml}\sum_{i=1}^{m}\sum_{k=1}^{l} 1_{\tilde{h}^k(x_i,y_i)-\max_{y\neq y_i}\tilde{h}^k(x_i,y)<0}$$

$$\leq\frac{1}{ml}\sum_{i=1}^{m}\sum_{k=1}^{l}\exp\left\{-\sum_{t=1}^{T}\alpha_t\rho\left(\tilde{h}_t^k,\boldsymbol{x}_i,\boldsymbol{y}_i\right)\right\} := F\left(\boldsymbol{\alpha}\right)$$

- Margin: $\rho\left(\tilde{h}^k,\boldsymbol{x}_i,\boldsymbol{y}_i\right)=\tilde{h}^k\left(\boldsymbol{x}_i,y_i^k\right)-\arg\max_{y^k\neq y_i^k}\tilde{h}^k\left(\boldsymbol{x}_i,y^k\right)$

ESPBoost

- Hypothesis space: the set of combined predictions

$$\mathcal{H}_{ESPBoost}(\boldsymbol{x}) = \left\{ h(\boldsymbol{x}) : \arg\max_{\boldsymbol{y} \in \mathcal{Y}} \tilde{h}(\boldsymbol{x}, \boldsymbol{y}) \right\}$$

- Objective Function $F : \mathbb{R}_+^T \to \mathbb{R}$

$$F(\boldsymbol{\alpha}) = \frac{1}{ml} \sum_{i=1}^{m} \sum_{k=1}^{l} \exp\left\{ -\sum_{t=1}^{T} \alpha_t \rho\left(\tilde{h}_t^k, \boldsymbol{x}_i, \boldsymbol{y}_i \right) \right\}$$

- Apply coordinate descent.

ESPBoost: algorithm

Algorithm 2 ESPBoost Algorithm.

Inputs: $S = ((\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_m, \mathbf{y}_m))$; set of experts $\{h_1, \ldots, h_p\}$.

for $i = 1$ **to** m **and** $k = 1$ **to** l **do**

$\quad \mathcal{D}_1(i, k) \leftarrow \frac{1}{ml}$

end for

for $t = 1$ **to** T **do**

$\quad \mathsf{h}_t \leftarrow \operatorname{argmin}_{\mathsf{h} \in \mathsf{H}} \mathbb{E}_{(i,k) \sim \mathcal{D}_t} [\mathbf{1}_{\mathsf{h}^k(\mathbf{x}_i) \neq y_i^k}]$

$\quad \epsilon_t \leftarrow \mathbb{E}_{(i,k) \sim \mathcal{D}_t} [\mathbf{1}_{\mathsf{h}_t^k(\mathbf{x}_i) \neq y_i^k}]$

$\quad \alpha_t \leftarrow \frac{1}{2} \log \frac{1-\epsilon_t}{\epsilon_t}$

$\quad Z_t \leftarrow 2\sqrt{\epsilon_t(1 - \epsilon_t)}$

$\quad$ **for** $i = 1$ **to** m **and** $k = 1$ **to** l **do**

$\quad\quad \mathcal{D}_{t+1}(i, k) \leftarrow \frac{\exp(-\alpha_t \rho(\widetilde{\mathsf{h}}_t^k, \mathbf{x}_i, \mathbf{y}_i)) \mathcal{D}_t(i,k)}{Z_t}$

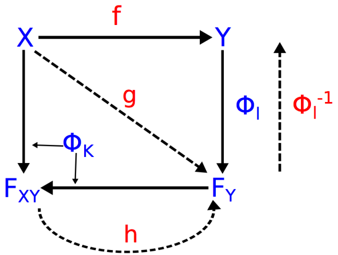
$\quad$ **end for**

end for

Return $\widetilde{\mathsf{h}} = \sum_{t=1}^{T} \alpha_t \widetilde{\mathsf{h}}_t$

Generalized Kernel Approach

- Problem Setting: Given $(x_i, y_i)_{i=1}^n \in \mathcal{X} \times \mathcal{Y}$, we want to learn a mapping f from $\mathcal{X}$ to $\mathcal{Y}$.



- $l : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ a kernel function
- $\mathcal{F}_{\mathcal{Y}}$: RKHS associated with l
- Φ_l: the mapping from $\mathcal{Y}$ to $\mathcal{F}_{\mathcal{Y}}$

- Step 1: Learn the mapping g from $\mathcal{X}$ to $\mathcal{F}_{\mathcal{Y}}$
- Step 2: Find the pre-image of $g(x)$

Generalized Kernel Approach

- Step 1: learn the mapping $g : \mathcal{X} \to \mathcal{F}_{\mathcal{Y}}$
- Preliminaries
  - Operator-valued kernels
  - Function-valued RKHS

Operator-valued Kernels

- $\mathcal{L}(\mathcal{F}_{\mathcal{Y}})$ be the set the bounded operators $\mathcal{T} : \mathcal{F}_{\mathcal{Y}} \rightarrow \mathcal{F}_{\mathcal{Y}}$
- Non-negative $\mathcal{L}(\mathcal{F}_{\mathcal{Y}}) - \textit{valued}$ kernel $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{L}(\mathcal{F}_{\mathcal{Y}})$ such that:
  - $\forall x_i, x_j \in \mathcal{X},\ K(x_i, x_j) = K(x_j, x_i)^*$.
  - For any $m > 0$, $\left\{(x_i, \varphi_i)_{i=1,\ldots,m}\right\} \subseteq \mathcal{X} \times \mathcal{F}_{\mathcal{Y}}$, $\sum_{i,j=1}^{m} \langle K(x_i, x_j)\varphi_j, \varphi_i\rangle_{\mathcal{F}_{\mathcal{Y}}} \geq 0$.
- Note: * denotes adjoint operator, i.e. $\forall \varphi_1, \varphi_2 \in \mathcal{F}_{\mathcal{Y}}$, $\langle K(\varphi_1), \varphi_2\rangle_{\mathcal{F}_{\mathcal{Y}}} = \langle \varphi_1, K^*(\varphi_2)\rangle_{\mathcal{F}_{\mathcal{Y}}}$

Function-valued RKHS

- A Hilbert space $\mathcal{F}_{\mathcal{X}\mathcal{Y}}$ of functions $g : \mathcal{X} \to \mathcal{F}_{\mathcal{Y}}$ is a $\mathcal{F}_{\mathcal{Y}} - valued$ RKHS if there is a non-negative $\mathcal{L}\left(\mathcal{F}_{\mathcal{Y}}\right) - valued$ kernel K with the following properties:
    - $\forall x \in \mathcal{X}, \forall \varphi \in \mathcal{F}_{\mathcal{Y}}$, the function $K\left(x, \cdot\right)\varphi \in \mathcal{F}_{\mathcal{X}\mathcal{Y}}$
    - $\forall g \in \mathcal{F}_{\mathcal{X}\mathcal{Y}}, \forall x \in \mathcal{X}, \forall \varphi \in \mathcal{F}_{\mathcal{Y}}$,
    $\langle g, K\left(x, \cdot\right)\varphi\rangle_{\mathcal{F}_{\mathcal{X}\mathcal{Y}}} = \langle g\left(x\right), \varphi\rangle_{\mathcal{F}_{\mathcal{Y}}}$
- Theorem: Bijection between $\mathcal{F}_{\mathcal{X}\mathcal{Y}}$ and K, as long as K is non-negative.

Kernel Ridge Regression

- Step 1: Learning $g : \mathcal{X} \to \mathcal{F}_{\mathcal{Y}}$
- Kernel Ridge Regression, with closed form solution:

$$\arg\min_{g \in \mathcal{F}_{\mathcal{X}\mathcal{Y}}} \sum_{i=1}^{n} \| g(x_i) - \Phi_l(y_i) \|_{\mathcal{F}_{\mathcal{Y}}}^2 + \lambda \| g \|_{\mathcal{F}_{\mathcal{X}\mathcal{Y}}}^2$$

$$g(x) = \boldsymbol{K}_x (\boldsymbol{K} + \lambda I)^{-1} \boldsymbol{\Phi}_{\boldsymbol{l}}$$

- $\boldsymbol{K}_x$: a row vector of operators, $[K(\cdot, x_i) \in \mathcal{L}(\mathcal{F}_{\mathcal{Y}})]_{i=1}^n$
- $\boldsymbol{K}$: a matrix of operators, $[K(x_i, x_j) \in \mathcal{L}(\mathcal{F}_{\mathcal{Y}})]_{i,j=1}^n$
- $\boldsymbol{\Phi}_{\boldsymbol{l}}$: a column vector of functions $[\Phi_l(y_i) \in \mathcal{F}_{\mathcal{Y}}]_{i=1}^n$

Find Pre-image

- Step 2: Find pre-image of $g(x)$

$$f(x) = \arg\min_{y \in \mathcal{Y}} \|g(x) - \Phi_l(y)\|^2_{\mathcal{F}_\mathcal{Y}}$$
$$= \arg\min_{y \in \mathcal{Y}} \left\| \boldsymbol{K}_x (\boldsymbol{K} + \lambda I)^{-1} \boldsymbol{\Phi_l} - \Phi_l(y) \right\|^2_{\mathcal{F}_\mathcal{Y}}$$
$$= \arg\min_{y \in \mathcal{Y}} l(y,y) - 2 \left\langle \boldsymbol{K}_x (\boldsymbol{K} + \lambda I)^{-1} \boldsymbol{\Phi_l}, \Phi_l(y) \right\rangle_{\mathcal{F}_\mathcal{Y}}$$

- $\Phi_l(y)$ unknown, use a generalized kernel trick:

$$\langle \mathcal{T}\Phi_l(y_i), \Phi_l(y) \rangle = [\mathcal{T} l(y_i, \cdot)](y)$$

- Express $f(x)$ using only kernel functions:

$$f(x) = \arg\min_{y \in \mathcal{Y}} l(y,y) - 2 \left[\boldsymbol{K}_x (\boldsymbol{K} + \lambda I)^{-1} \boldsymbol{L_\cdot} \right](y)$$

where $\boldsymbol{L_\cdot}$ is a column vector of $[l(y_i, \cdot)]_{i=1}^n$

Covariance-based Operator-valued Kernels

- Covariance-based operator-valued kernels:

$$K(x_i, x_j) = k(x_i, x_j)\, C_{YY}$$

  - $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ a scalar-valued kernel;
  - $C_{YY} : \mathcal{F}_{\mathcal{Y}} \to \mathcal{F}_{\mathcal{Y}}$ a covariance operator defined by a random variable $Y \in \mathcal{Y}$:

$$\langle \varphi_1, C_{YY}\varphi_2 \rangle_{\mathcal{F}_{\mathcal{Y}}} = \mathbb{E}\left[\varphi_1(Y)\,\varphi_2(Y)\right]$$

  - Empirical covariance operator

$$\hat{C}_{YY}(\varphi) = \frac{1}{n}\sum_{i=1}^{n} \varphi(y_i)\, l(\cdot, y_i)$$

- To account for the effects of input, we could also use conditional covariance operator

$$C_{YY|X} = C_{YY} - C_{YX} C_{XX}^{-1} C_{XY}$$

Conclusion

- Ensemble methods: ensemble learning with expended 'path experts'.
  - On-line algorithm (WMWP): efficient for learning and inference by exploiting the output structure.
  - On-line-to-batch-conversion: randomized and deterministic algorithms with learning guarantees.
  - Boosting: efficient for output structure.
- Kernel method:
  - Use a joint feature space.
  - Covariance-based operator-valued kernel to encode interactions between outputs.
  - Conditional Covariance-based operator to correlate input with 'interaction between outputs'.
  - Express the final hypothesis with only kernel functions.

Reference

Corinna Cortes, Vitaly Kuznetsov, and Mehryar Mohri.
Ensemble methods for structured prediction.
In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1134–1142, 2014.

Hachem Kadri, Mohammad Ghavamzadeh, and Philippe Preux.
A Generalized Kernel Approach to Structured Output Learning.
In *International Conference on Machine Learning (ICML)*, Atlanta, United States, June 2013.