

- 1 Standard Optimization
- 2 Fixed-Share
- 3 Tracking Source
- 4 Weight-Push

Problem Definition

An s -space δ -delay R -capacity-channel coding scheme is defined by the encode/decode functions f, g .

The source symbols are $x_1, \dots, x_n \in X$, channel signals b_1, \dots where $b_i \in 1, \dots, 2^R$, and reproduction symbols \hat{x}_1, \dots .

Let x^j be x_1, \dots, x_j , similarly for b^j and \hat{x}^j . Assume loss $l(x, \hat{x}) \in [0, B]$.

Then $b_i = f(x^{i+\delta})$, and $\hat{x}_i = g(b^i)$

Hedge Review

Hedge: policy is usually a deterministic normalized weight \vec{p} .

In this case, policy is a distribution over experts, and we evaluate expected payoff. The objective is identical.

Recall: $\sqrt{(T/2\ln(n))}$ regret bound for hedge, after tuning η

Fixed-Share Review

For N the number of experts Fixed share:

$$w_{t+1,i} = \frac{\alpha W_{t+1}}{N} + (1 - \alpha) e^{-\eta I(y_t, y_t^{(i)})} w_{t,i}$$

Then sample $\hat{y}_t = y_t^{(i)}$ with probability $w_{t,i} / W_t$.

Problem: if $N = T^\gamma$ for $\gamma > 0$, the updates could be computationally expensive.

Efficient Fixed-Share

The inefficiency is in the updates to \vec{w} . The solution they give: expand the recursion.

$$\text{So } W_t = \frac{\alpha}{N} \sum_{t'=2}^{t-1} (1-\alpha)^{t-1-t'} W_{t'} Z_{t',t-1}$$

and

$$\frac{w_t^{(i)}}{W_t} = \frac{(1-\alpha)^{t-1}}{NW_t} e^{-\eta L([1,t-1],i)} + \frac{\alpha}{NW_t} \sum_{t'=2}^{t-1} (1-\alpha)^{t-t'} W_{t'} e^{-\eta L([t',t-1],i)} + \frac{\alpha}{N}$$

Where $Z_{t',t-1} = \sum_{i=1}^N e^{-\eta L([t',t-1],i)}$

Efficient Fixed-Share

Then to sample \hat{y}_t , first sample the time step τ_t .

$$P(\tau_t = t') = \frac{\alpha}{N W_t} (1 - \alpha)^{t-t'} W_{t'} \text{ for } t' \geq 2$$

Then sample $P(\hat{y}_t | \tau_t) = e^{-\eta L([t', t-1], i)} / Z'_t, t - 1$

We assume Z and $P(\hat{y}_t | \tau_t)$ can be computed efficiently from prior work on exponentially weighted averages. Then W_t can be computed using the recursion on prior slide.

Tracking Source Code

A reference code is an encoder/decoder pair (f, g) .

Define F_s^δ as the set of reference codes using s -space with delay δ .

Select some finite subset $F \subseteq F_s^\delta$. $F_{m,n}$ is a code which changes codes in F m times over n examples.

Then minimum possible cumulative distortion is

$$\frac{1}{n} \min_{1 \leq i_1 < \dots < i_n} \sum_{j=0}^m \min_{(f,g) \in F} \sum d(x_{i_j}, g_i(f_{i-s}(x^{i+\delta-s}, \dots f_i(x^{i+\delta}))))$$

Tracking Source Code

Algorithm: partition sequence of n examples into subsequences of length l . At the beginning of each subsequence, $\log|F|/R$ bits are used to communicate the coding method being used.

During the period where the reference code choice is being sent, random is predicted.

We can view the choice of an expert for each l -length-sequence as a single prediction. We've then mapped this to the standard switching-experts problem.

Thus, the minimum loss of any algorithm that is restricted to by at changes at the segment boundaries may exceed for each occasion the change in the optimal idealized by at most $l-1$.

Motivation

- Want a kernel function (like RBF, polynomial, ...) that takes into account graph structure relating features.
- Each path in the graph is a feature.
- Value along a path is the product of the edges (like probability in routing)

Definition

Path Kernel Definition

A graph $G = (V, E)$ and a node $u \in V$ defines a path kernel function K . Let $n = |E|$.

Then $\forall x, y \in \mathbb{R}^n$, $K(x, y) = \sum_{p \in P} \prod_{e \in p} x_e y_e$

Path-weighting-properties:

- $w_P = \prod_{e \in P} v_e$
- $\sum_{u': (u, u') \in E(G)} v_{(u, u')} = 1$
- $\sum_P w_P = 1$

Weight Push Update

We want to be able to perform multiplicative updates of the weight vector over edges, but maintain the above three properties of the assignments to edges.

For original edge weights v_e satisfying 3 properties, and weights b_e to multiply, update edges: $\tilde{v}_e = \frac{v_e b_e K_{u'}(v, b)}{K_u(v, b)}$

Maintains 3 Properties

Product property above holds for updated weights:

$$\begin{aligned}
 \prod_{e \in P} \tilde{v}_e &= \prod_{i=1}^k \frac{v_{(u_{i-1}, u_i)} b_{(u_{i-1}, u_i)} K_{u_i}(v, b)}{K_{u_{i-1}}(v, b)} \\
 &= \frac{\prod_{e \in P} v_e b_e K_{\text{sink}}(v, b)}{K_{\text{source}}(v, b)} \\
 &= \frac{w_P \prod_{e \in P} b_e}{K_{\text{source}}(v, b)} \\
 &= \frac{w_P \prod_{e \in P} b_e}{\sum_P \prod_{e \in P} b_e}
 \end{aligned}$$

Other two properties are easy to verify

Scalar Quantization

$$f_i : [0, 1]^i \times [0, 1]^i \rightarrow 1, \dots, M \quad g_i : 1, \dots, M \rightarrow [0, 1]$$

The class of reference codes is not finite. Solution: restrict to a class Q_K of reference codes.