Supervised Sequential Classification Under Budget Constraints: Supplementary Material

Kirill Trapeznikov
Boston University

Venkatesh Saligrama
Boston University

1 Proofs

Proof of Theorem 1 To simplify our derivations, we assume uniform class prior probability: $\mathrm{P}_y\left[y = \hat{y}\right] = \frac{1}{c},\ \hat{y} = 1, \ldots, C$. However, our results can be easily modified to account for a non-uniform prior. The expected conditional risk can be solved optimally by a dynamic program, where a DP recursion is,

$$J_K(\mathbf{x}^K, S^K) = \min_{f^K} \mathbf{E}_y\left[S^K(\mathbf{x}^K) R_k(y, \mathbf{x}^K, f^K)\right] \tag{1}$$

$$J_k(\mathbf{x}^k, S^k) = \min_{f^k} \Big\{ \mathbf{E}_y\left[S^k(\mathbf{x}^k) R_k(y, \mathbf{x}^k, f^k)\right] + \tag{2}$$

$$\mathbf{E}_{\mathbf{x}^{k+1}\ldots\mathbf{x}^K}\left[J_{k+1}(\mathbf{x}^{k+1}, S^{k+1}) \mid \mathbf{x}^k\right] \Big\} \tag{3}$$

Consider kth stage minimization, f^k can take $C+1$ possible values $\{1, 2, \ldots C,\ r\}$ and $J_k(\mathbf{x}^k, S^k)$ can be recast as an conditional expected risk minimization,

$$J_k(\mathbf{x}^k, S^k = 1) =$$

$$\min_{f^k} \Bigg\{ \underbrace{\mathrm{P}_y\left[y \neq \hat{y} \mid \mathbf{x}^k\right]}_{f^k(\mathbf{x}^k)=\hat{y}}, \underbrace{\delta^k + \mathbf{E}_{\mathbf{x}^{k+1}\ldots\mathbf{x}^K}\left[J_{k+1}(\mathbf{x}^{k+1}, 1) \mid \mathbf{x}^k\right]}_{f^k(\mathbf{x}^k)=r} \Bigg\} \tag{4}$$

Define,

$$\tilde{\delta}(x^k) = \delta^{k+1} + \mathbf{E}_{x^{k+1}\ldots x^K}\left[J_{k+1}(x^{k+1}, S^{k+1} = 1)\right]$$

and rewrite the conditional risk in 4,

$$f^k = \arg\min_f \Bigg\{ \underbrace{1 - \mathrm{P}_y\left[y = \hat{y} \mid \mathbf{x}^k\right]}_{f(\mathbf{x}^k)=\hat{y}}, \underbrace{\tilde{\delta}^k(\mathbf{x}^k)}_{f(\mathbf{x}^k)=r} \Bigg\} \tag{5}$$

Reject is the optimal decision if,

$$\min_{\hat{y}} \left\{1 - \mathrm{P}_y\left[y = \hat{y} \mid \mathbf{x}^k\right]\right\} \geq \tilde{\delta}^k(\mathbf{x}^k) \tag{6}$$

$$\max_{\hat{y}} \left\{\mathrm{P}_y\left[y = \hat{y} \mid \mathbf{x}^k\right]\right\} \leq 1 - \tilde{\delta}^k(\mathbf{x}^k) \tag{7}$$

If reject is not the optimal strategy then a class is chosen to maximize the posterior probability:

$$f^k(\mathbf{x}^k) = \arg\max_{\hat{y} \in \{1,\ldots,c\}} \left\{\mathrm{P}_y\left[y = \hat{y} \mid \mathbf{x}^k\right]\right\} \tag{8}$$

which is exactly our claim.

Proof of Lemma 2 Define an auxiliary variable corresponding to the error penalty term and absolute value of the maximizing codeword projection respectively:

$$e_i = \mathbf{1}_{\left[d^k(\mathbf{x}_i^k) \neq y_i\right]},\ z_i = \sigma_{d^k}(\mathbf{x}_i^k) \tag{9}$$

$$\begin{aligned}
\tilde{R}_k^i(\cdot) &= e_i \mathbf{1}_{[g(x^k)-z_i<0]} + \tilde{\delta}_i^k \mathbf{1}_{[g(x^k)-z_i\geq 0]} && (10)\\
&= e_i \mathbf{1}_{[g(x^k)-z_i<0]} + \tilde{\delta}_i^k \left\{1 - \mathbf{1}_{[g(x^k)-z_i<0]}\right\} && (11)\\
&= \left\{e_i - \tilde{\delta}_i^k\right\} \mathbf{1}_{[g(x^k)-z_i<0]} + \tilde{\delta}_i^k && (12)
\end{aligned}$$

Define weights $w_i = e_i - \tilde{\delta}_i^k$ and drop the $\tilde{\delta}_i^k$ term since it does not depend on $g(\cdot)$. Our goal is to minimize $\sum S_i^k \tilde{R}_k^i$ over g. We will split the summation into two sets:

$$\begin{aligned}
&= \sum_{w_i\geq 0} S_i^k w_i \mathbf{1}_{\left[\left(g(\mathbf{x}_i^k)-z_i\right)\leq 0\right]} + \sum_{w_i<0} S_i^k w_i \mathbf{1}_{\left[\left(g(x_i^k)-z_i\right)\leq 0\right]} && (13)\\
&= \sum_{w_i\geq 0} S_i^k w_i \mathbf{1}_{\left[\left(g(x_i^k)-z_i\right)\leq 0\right]} + \sum_{w_i<0} S_i^k w_i \left\{1 - \mathbf{1}_{\left[\left(g(x_i^k)-z_i\right)>0\right]}\right\} && (14)
\end{aligned}$$

If discard the constant term $\sum_{w_i<0} S_i^k w_i$ and introduce pseudo labels $b_i = \begin{cases} +1, & w_i \geq 0 \\ -1, & w_i < 0 \end{cases}$ then,

$$\arg\min_g \sum_{i=1}^N S_i^k \tilde{R}_k^i = \arg\min_g \sum_{i=1}^N S_i^k \left|w_i\right| \mathbf{1}_{\left[b_i\left(g(x_i^k)-z_i\right)\leq 0\right]} \tag{15}$$

Proof of Theorem 3 At each stage the reject decision can be expressed in terms of three boolean decisions:

$$\mathbf{1}_{[|h^k(\mathbf{x}^k)|-g^k(\mathbf{x}^k)\leq 0]} = \underbrace{\mathbf{1}_{[h^k(\mathbf{x}^k)>0]}}_{\text{Decision 1}} \underbrace{\mathbf{1}_{[h^k(\mathbf{x}^k)-g^k(\mathbf{x}^k)\leq 0]}}_{\text{Decision 2}} + \underbrace{\mathbf{1}_{[h^k(\mathbf{x}^k)\leq 0]}}_{\text{Not Decision 1}} \underbrace{\mathbf{1}_{[-h^k(\mathbf{x}^k)-g^k(\mathbf{x}^k)\leq 0]}}_{\text{Decision 3}} \tag{16}$$

If the rejectors ($g^k \in \mathcal{G}^k$) and stage classifiers ($h^k \in \mathcal{H}^k$) belong to families with finite VC dimensions then the complexity of Decision 2 and Decision 3 is $\mathcal{VC}[\mathcal{G}^k] + \mathcal{VC}[\mathcal{H}^k]$

The system classifier, F, is composed of K stages. Each of the first $K-1$ stages can be expressed as a boolean function of 3 boolean decisions. The last stage is a single boolean decision. So the output F can be expressed as a boolean function of $3(K-1)+1 = 3K-2$ functions. We know the VC dimension for each of the functions. Using this fact and Lemma 2 in [**?**] we obtain our result.

2 Implementation Details

For large datasets ($N > 1000$), we split them 50/10/40% into train, validation and test sets. The performance reported is on the test set. For smaller datasets ($N < 1000$), we perform 50 random 70/10/20% splits and report the average performance over the trials. Each subproblem reduces to minimizing a weighted binary error problem with respect to a logistic loss. Polynomial kernel classifier of degree q is parametrized by a vector $\mathbf{a}$:

$$h(x) = \sum_{i=1}^N a_i(\mathbf{x}_i^T \mathbf{x} + 1)^q$$

The optimization over the polynomial kernel classifier is performed using newton gradient descent method. Table 1 shows the degree of polynomial kernels used in our simulations.

Dataset	$\mathcal{H}^1$	$\mathcal{G}^1$	$\mathcal{H}^2$	$\mathcal{G}^2$	$\mathcal{H}^3$	$\mathcal{G}^3$	$\mathcal{H}^4$
synthetic	2	2	2				
mam	2	0	2				
pima	2	0	2	0	2		
threat	5	5	5	5	5		
covertype	1	1	1	1	1		
letter	7	2	7	2	7		
mnist	1	1	1	1	1	1	1
landsat	3	2	3	2	3	2	3

Table 1: Stage Complexity: we use polynomial kernel classifiers. This table displays the degree of the polynomial kernel used at each stage for the rejector and the stage classifier