The Power of Log-Sum-Exp: Sequential Density Ratio Matrix Estimation for Speed-Accuracy Optimization

Taiki Miyagawa ¹ Akinori F. Ebihara ¹

Abstract

We propose a model for multiclass classification of time series to make a prediction as early and as accurate as possible. The matrix sequential probability ratio test (MSPRT) is known to be asymptotically optimal for this setting, but contains a critical assumption that hinders broad real-world applications; the MSPRT requires the underlying probability density. To address this problem, we propose to solve density ratio matrix estimation (DRME), a novel type of density ratio estimation that consists of estimating matrices of multiple density ratios with constraints and thus is more challenging than the conventional density ratio estimation. We propose a log-sum-exp-type loss function (LSEL) for solving DRME and prove the following: (i) the LSEL provides the true density ratio matrix as the sample size of the training set increases (consistency); (ii) it assigns larger gradients to harder classes (hard class weighting effect); and (iii) it provides discriminative scores even on class-imbalanced datasets (guess-aversion). Our overall architecture for early classification, MSPRT-TANDEM, statistically significantly outperforms baseline models on four datasets including action recognition, especially in the early stage of sequential observations. Our code and datasets are publicly available¹.

1. Introduction

Classifying an incoming time series as early and as accurately as possible is challenging yet crucial, especially when the sampling cost is high or when a delay results in serious consequences (Xing et al., 2009; 2012; Mori et al., 2015; Mori et al., 2018). Early classification of time series is a multi-objective optimization problem, and there is usually no ground truth indicating when to stop observation and classify a sequence.

The MSPRT is a provably optimal algorithm for early multiclass classification and has been developed in mathematical statistics (Armitage, 1950; Chernoff, 1959; Kiefer & Sacks, 1963; Lorden, 1967; 1977; Pavlov, 1984; Dragalin, 1987; Pavlov, 1991; Baum & Veeravalli, 1994; Dragalin & Novikov, 1999). The MSPRT uses a matrix of log-likelihood ratios (LLRs), the $(k,l)$-entry of which is the LLR of hypothesis $H_k$ to hypothesis $H_l$ and depends on the current time $t$ through consecutive observations of sequential data $X^{(1:t)}$ (Figure 1). A notable property of the MSPRT is that it is asymptotically optimal (Tartakovsky, 1998): It achieves the minimum stopping time among all the algorithms with bounded error probabilities as the thresholds go to infinity, or equivalently, as the error probabilities go to zero or the stopping time goes to infinity (Appendix A). Therefore, the MSPRT is a promising approach to early multiclass classification with strong theoretical support.

However, the MSPRT has a critical drawback that hinders its real-world applications in that it requires the true LLR matrix, which is generally inaccessible. To address this problem, we propose to solve density ratio matrix estimation (DRME); i.e., we attempt to estimate the LLR matrix from a dataset. DRME has yet to be explored in the literature but can be regarded as a generalization of the conventional density ratio estimation (DRE), which usually focuses on only two densities (Sugiyama et al., 2012). The difficulties with DRME come from simultaneous optimization of multiple density ratios; the training easily diverges when the denominator of only one density ratio is small. In fact, a naive application of conventional binary DRE-based loss functions does not generalize well in this setting, and sometimes causes instability and divergence of the training (Figure 2 Top).

Therefore, we propose a novel loss function for solving DRME, the log-sum-exp loss (LSEL). We prove three properties of the LSEL, all of which contribute to enhancing the performance of the MSPRT. (i) The LSEL is consistent; i.e., by minimizing the LSEL, we can obtain the true LLR

¹NEC Corporation, Kanagawa, Japan. Correspondence to: Taiki Miyagawa <miyagawataik@nec.com>.

¹https://github.com/TaikiMiyagawa/MSPRT-TANDEM
matrix as the sample size of the training set increases. (ii) The LSEL has the hard class weighting effect; i.e., it assigns larger gradients to harder classes, accelerating convergence of neural network training. Our proof also explains why log-sum-exp-type losses, e.g., (Song et al., 2016; Wang et al., 2019; Sun et al., 2020), have performed better than sum-log-exp-type losses. (iii) We propose the cost-sensitive LSEL for class-imbalanced datasets and prove that it is guess-averse (Beijbom et al., 2014). Cost-sensitive learning (Elkan, 2001), or loss re-weighting, is a typical and simple solution to the class imbalance problem (Kubat & Matwin, 1997; Japkowicz & Stephen, 2002; He & Garcia, 2009; Buda et al., 2018). Although the consistency does not necessarily hold for the cost-sensitive LSEL, we show that the cost-sensitive LSEL nevertheless provides discriminative “LLRs” (scores) by proving its guess-aversion.

Along with the novel loss function, we propose the first DRE-based model for early multiclass classification in deep learning, MSPRT-TANDEM, enabling the MSPRT’s practi-
Our contributions are summarized as follows.

1. We formulate a novel problem setting, DRME, to enable the MSPRT on real-world datasets.

2. We propose a loss function, LSEL, and prove its consistency, hard class weighting effect, and guess-aversion.

3. We propose MSPRT-TANDEM: the first DRE-based model for early multiclass classification in deep learning. We show that it outperforms baseline models statistically significantly.

2. Related Work

Early classification of time series. Early classification of time series aims to make a prediction as early and as accurately as possible (Xing et al., 2009; Mori et al., 2015; 2016; Mori et al., 2018). An increasing number of real-world problems require earliness as well as accuracy, especially when a sampling cost is high or when a delay results in serious consequences; e.g., early detection of human actions for video surveillance and health care (Vats & Chan, 2016), early detection of patient deterioration on real-time sensor data (Mao et al., 2012), early warning of power system dynamics (Zhang et al., 2017), and autonomous driving for early and safe action selection (Donà et al., 2019). In addition, early classification saves computational costs (Ghodrati et al., 2021).

SPRT. Sequential multihypothesis testing has been developed in (Sobel & Wald, 1949; Armitage, 1950; Paulson, 1963; Simons, 1967). The extension of the binary SPRT to multihypothesis testing for i.i.d. data was conducted in (Armitage, 1950; Chernoff, 1959; Kiefer & Sacks, 1963; Lorden, 1967; 1977; Pavlov, 1984; Draganal, 1987; Pavlov, 1991; Baum & Veevarelli, 1994; Draganal & Novikov, 1999). The MSPRT for non-i.i.d. distributions was discussed in (Lai, 1981; Tartakovsky, 1998; Draganal et al., 1999; Tartakovsky et al., 2014). The asymptotic optimality of the MSPRT was proven in (Tartakovsky, 1998).

Density ratio estimation. DRE consists of estimating a ratio of two densities from their samples without separately estimating the numerator and denominator (Sugiyama et al., 2012). DRE has been widely used for, e.g., covariate shift adaptation (Sugiyama et al., 2008), representation learning (Oord et al., 2018; Hjelm et al., 2019), mutual information estimation (Belghazi et al., 2018), and off-policy reward estimation in reinforcement learning (Liu et al., 2018). Our proof of the consistency of the LSEL is based on (Gutmann & Hyvärinen, 2012).

We provide more extensive references in Appendix B. To the best of our knowledge, only (Ebihara et al., 2021) and (Moustakides & Basioti, 2019) combine the SPRT with DRE. Both restrict the number of classes to only two. The loss function proposed in (Ebihara et al., 2021) has not been proven to be unbiased; there is no guarantee for the estimated LLR to converge to the true one. (Moustakides & Basioti, 2019) does not provide empirical validation for the SPRT.

3. Density Ratio Matrix Estimation for MSPRT

3.1. Log-Likelihood Ratio Matrix

Let \( p \) be a probability density over \( (X^{(1:T)}, y) \), \( X^{(1:T)} = \{x^{(t)}\}_{t=1}^T \in \mathcal{X} \) is an example of sequential data, where \( T \in \mathbb{N} \) is the sequence length. \( x^{(t)} \in \mathbb{R}^d \) is a feature vector at timestamp \( t \); e.g., an image at the \( t \)-th frame in a video \( X^{(1:T)} \). \( y \in \mathcal{Y} = [K] := \{1, 2, ..., K\} \) is a multiclass label, where \( K \in \mathbb{N} \) is the number of classes. The LLR matrix is defined as \( \lambda(X^{(1:T)}) := (\lambda_{kl}(X^{(1:T)}))_{k,l \in [K]} := (\log p(X^{(1:T)}|y = k)/p(X^{(1:T)}|y = l))_{k,l \in [K]} \), where \( p(X^{(1:T)}|y) \) is a conditional probability density. \( \lambda(X^{(1:T)}) \) is an anti-symmetric matrix by definition; thus the diagonal entries are 0. Also, \( \lambda \) satisfies \( \lambda_{kl} + \lambda_{lm} = \lambda_{km} \) (\( \forall k,l,m \in [K] \)). Let \( \lambda(X^{(1:T)}, \theta) := (\lambda_{kl}(X^{(1:T)}, \theta))_{k,l \in [K]} := (\log \frac{p_{\theta}(X^{(1:T)}|y = k)\lambda}{p_{\theta}(X^{(1:T)}|y = l)}\lambda)_{k,l \in [K]} \) be an estimator of the true LLR matrix \( \lambda(X^{(1:T)}) \), where \( \theta \in \mathbb{R}^{d_{\theta}} \) (\( d_{\theta} \in \mathbb{N} \)) denotes trainable parameters, e.g., weight parameters of a neural network. We use the hat symbol (\( \hat{\cdot} \)) to highlight that the quantity is an estimated value. The \( \lambda \) should be anti-symmetric and satisfy \( \lambda_{kl} + \lambda_{lm} = \lambda_{km} \) (\( \forall k,l,m \in [K] \)). To satisfy these constraints, one may introduce additional regularization terms to the objective loss function, which can cause learning instability. Instead, we use specific combinations of the posterior density ratios \( \hat{p}_{\theta}(y = k|X^{(1:T)})/\hat{p}_{\theta}(y = l|X^{(1:T)}) \), which explicitly satisfy the aforementioned constraints (see the following M-TANDEM and M-TANDEMwO formulae).

3.2. MSPRT

Formally, the MSPRT is defined as follows (see Appendix A for more details):
A crucial property of the LSEL is consistency. Let \( P \) and \( P_k (k \in [K]) \) be probability distributions. Define a threshold matrix \( a_{kl} \in \mathbb{R} (k, l \in [K]) \), where the diagonal elements are immaterial and arbitrary, e.g., 0. The MSPRT of multihypothesis \( H_k : P = P_k (k \in [K]) \) is defined as \( \delta^* := (d^*, \tau^*) \), where \( d^* := k \) if \( \tau^* = \tau_k (k \in [K]) \), \( \tau^* := \min \{ \tau_k | k \in [K] \} \), and \( \tau_k := \inf \{ t \geq 1 \} \min_{l(\neq k) \in [K]} \{ \lambda_{kl}(X(1,t)) - a_{lk} \} \geq 0 \} \).

In other words, the MSPRT terminates at the smallest timestamp \( t \) such that for a class of \( k \in [K] \), \( \lambda_{kl}(t) \) is greater than or equal to the threshold \( a_{kl} \) for all \( l(\neq k) \) (Figure 1). By definition, we must know the true LLR matrix \( \lambda(X(1,t)) \) of the incoming time series \( X(1,t) \); therefore, we estimate \( \lambda \) with the help of the LSEL defined in the next section. For simplicity, we use single-valued threshold matrices \( a_{kl} = a_{kl}^' \) for all \( k, l, k', l' \in [K] \) in our experiment.

### 3.3. LSEL for DRME

To estimate the LLR matrix, we propose the log-sum-exp loss (LSEL):

\[
L_{\text{LSEL}}(\lambda) := \frac{1}{KT} \sum_{k \in [K]} \sum_{l \in [T]} \int dx(1,t)p(x(1,t)|k) \log(1 + \sum_{l(\neq k)} e^{-\lambda_{kl}(x(1,t))}). \tag{1}
\]

Let \( S := \{(x_i^{(1,T)}, y_i)\}_{i=1}^M \sim p(x(1,T), y)^M \) be a training dataset, where \( M \in \mathbb{N} \) is the sample size. The empirical approximation of the LSEL is

\[
\hat{L}_{\text{LSEL}}(\theta; S) := \frac{1}{KT} \sum_{k \in [K]} \sum_{l \in [T]} \frac{1}{M_k} \sum_{i \in I_k} \log(1 + \sum_{l(\neq k)} e^{-\hat{\lambda}_{kl}(x_i^{(1,t)}, \theta)}). \tag{2}
\]

\( M_k \) and \( I_k \) denote the sample size and index set of class \( k \), respectively; i.e., \( M_k = |\{ i \in [M] | y_i = k \}| = |I_k| \) and \( \sum_k M_k = M \).

#### 3.3.1. CONSISTENCY

A crucial property of the LSEL is consistency; therefore, by minimizing (2), the estimated LLR matrix \( \hat{\lambda} \) approaches the true LLR matrix \( \lambda \) as the sample size increases. The formal statement is given as follows:

**Theorem 3.1 (Consistency of the LSEL).** Let \( L(\theta) \) and \( \hat{L}_S(\theta) \) denote \( L_{\text{LSEL}}[\lambda(\cdot; \theta)] \) and \( \hat{L}_{\text{LSEL}}(\theta; S) \) respectively. Let \( \Theta_S \) be the empirical risk minimizer of \( L_S \); namely, \( \Theta_S := \arg \min L_S(\theta) \). Let \( \Theta^* := \{ \theta^* \in \mathbb{R}^{d_\theta} | \lambda(X(1,t); \theta^*) = \lambda(X(1,t)) \ (\forall t \in [T]) \} \) be the target parameter set. Assume, for simplicity of proof, that each \( \theta^* \) is separated in \( \Theta^* \); i.e., \( \exists \delta > 0 \) such that \( B(\theta^*; \delta) \cap B(\theta^*; \delta) = \emptyset \) for arbitrary \( \theta \) and \( \theta^* \in \Theta^* \), where \( B(\theta; \delta) \) denotes an open ball at center \( \theta \) with radius \( \delta \). Assume the following three conditions:

(a) \( \forall k, l \in [K], \forall t \in [T], \ p(X(1,t)|k) = 0 \iff p(X(1,t)|l) = 0 \).

(b) \( \sup_{\theta} |\hat{L}_S(\theta) - L(\theta)| \xrightarrow{M \to \infty} 0 \); i.e., \( \hat{L}_S(\theta) \) converges in probability uniformly over \( \theta \) to \( L(\theta) \).

(c) For all \( \theta^* \in \Theta^* \), there exist \( t \in [T], k \in [K] \) and \( l \in [K] \), such that the following \( d_\theta \times d_\theta \) matrix is full-rank:

\[
\int dx(1,t)p(x(1,t)|k) \times \nabla_{\theta^*} \hat{\lambda}_{kl}(X(1,t); \theta^*) \nabla_{\theta^*} \hat{\lambda}_{kl}(X(1,t); \theta^*)^T. \tag{3}
\]

Then, \( P(\hat{\Theta}_S \notin \Theta^*) \xrightarrow{M \to \infty} 0 \); i.e., \( \hat{\Theta}_S \) converges in probability into \( \Theta^* \).

Assumption (a) ensures that \( \lambda(X(1,t)) \) exists and is finite. Assumption (b) can be satisfied under the standard assumptions of the uniform law of large numbers (compactness, continuity, measurability, and dominance) (Jennrich, 1969; Newey & McFadden, 1986). Assumption (c) is a technical requirement, often assumed in the literature (Gutmann & Hyvärinen, 2012). The complete proof is given in Appendix C.

The critical hurdle of the MSPRT to practical applications (availability to the true LLR matrix) is now relaxed by virtue of the LSEL, which is provably consistent and enables a precise estimation of the LLR matrix. We emphasize that the MSPRT is the earliest and most accurate algorithm for early classification of time series, at least asymptotically (Theorem A.1, A.2, and A.3).

#### 3.3.2. HARD CLASS WEIGHTING EFFECT

We further discuss the LSEL by focusing on a connection with hard negative mining (Song et al., 2016). It is empirically known that designing a loss function to emphasize hard classes improves model performance (Lin et al., 2017). The LSEL has this mechanism.

Let us consider a multiclass classification problem to obtain a high-performance discriminative model. To emphasize hard classes, let us minimize \( \hat{L} := \frac{1}{KT} \sum_{k \in [K]} \sum_{l \in [T]} \frac{1}{M_k} \sum_{i \in I_k} \max_{l(\neq k)} \{ e^{-\hat{\lambda}_{kl}(x_i^{(1,t)}, \theta)} \} \); however, mining the single hardest class with the max function induces a bias and causes the network to converge to a bad local minimum. Instead of \( \hat{L} \), we can use the LSEL because it is not only provably consistent but is a smooth upper bound of \( \hat{L} \): Because
max \{ e^{-\hat{\lambda}_{yk}(X^{(1,t)})} \} < \sum_{l \neq y} e^{-\hat{\lambda}_{yl}(X^{(1,t)})},
we obtain \( \hat{L} < \hat{L}_{LSEL} \) by summing up both sides with respect to \( i \in I_k \) and then \( k \in [K] \) and \( t \in [T] \). Therefore, a small \( \hat{L}_{LSEL} \) indicates a small \( \hat{L} \). In addition, the gradients of the LSEL are dominated by the hardest class \( k^* \in \arg\max_{k \neq y} \{ e^{-\hat{\lambda}_{yk}(X^{(1,t)})} \} \), because for all \( k \neq y, k^* \),
\[
\left| \frac{\partial\hat{L}_{LSEL}}{\partial \lambda_{yk}} \right| \propto \frac{e^{-\hat{\lambda}_{yk}}}{\sum_{l \in [K]} e^{-\hat{\lambda}_{yl}}} \leq \frac{e^{-\hat{\lambda}_{yk^*}}}{\sum_{l \in [K]} e^{-\hat{\lambda}_{yl}}},
\]
meaning that the LSEL assigns large gradients to the hardest class during training, which accelerates convergence.

Let us compare the hard class weighting effect of the LSEL with that of the logistic loss (a sum-log-exp-type loss extensively used in machine learning). For notational convenience, let us define \( \ell_{LSEL} := \log(1 + \sum_{k \neq y} e^{a_k}) \) and \( \ell_{\text{logistic}} := \sum_{k \neq y} \log(1 + e^{a_k}) \), where \( a_k := -\hat{\lambda}_{yk}(X^{(1,t)}, \theta) \), and compare their gradient scales. The gradients for \( k \neq y \) are:
\[
\frac{\partial \ell_{\text{logistic}}}{\partial \lambda_{yk}} = -\frac{e^{-\hat{\lambda}_{yk}}}{1 + e^{-\hat{\lambda}_{yk}}}, \quad b_k,
\]
\[
\frac{\partial \hat{L}_{LSEL}}{\partial \lambda_{yk}} = -\frac{e^{-\hat{\lambda}_{yk}}}{\sum_{l \in [K]} e^{-\hat{\lambda}_{yl}}}, \quad c_k.
\]
The relative gradient scales of the hardest class to the easiest class are:
\[
R_{\text{logistic}} := \max_{k \neq y} \frac{b_k}{\min_{k \neq y} b_k} = \frac{e^{a_{k^*}} e^{1+a_{k^*}}}{e^{a_k^*} e^{1+a_{k^*}}},
\]
\[
R_{\text{LSEL}} := \max_{k \neq y} \frac{c_k}{\min_{k \neq y} c_k} = \frac{e^{a_{k^*}}}{e^{a_k^*}},
\]
where \( k^* := \arg\min_{k \neq y} \{ a_k \} \). Since \( R_{\text{logistic}} \leq R_{\text{LSEL}} \), we conclude that the LSEL weighs hard classes more than the logistic loss. Note that our discussion above also explains why log-sum-exp-type losses (e.g., (Song et al., 2016; Wang et al., 2019; Sun et al., 2020)) perform better than sum-log-exp-type losses. In addition, Figure 2 (Top and Bottom) shows that the LSEL performs better than the logistic loss—a result that supports the discussion above. See Appendix E for more empirical results.

3.3.3. Cost-Sensitive LSEL and Guess-Aversion

Furthermore, we prove that the cost-sensitive LSEL provides discriminative scores even on imbalanced datasets. Conventional research for cost-sensitive learning has been mainly focused on binary classification problems (Fan et al., 1999; Elkan, 2001; Viola & Jones, 2002; Masnadi-Shirazi & Vasconcelos, 2010). However, in multiclass cost-sensitive learning, (Beijbom et al., 2014) proved that random score functions (a “random guess”) can lead to even smaller values of the loss function. Therefore, we should investigate whether our loss function is averse (robust) to such random guesses, i.e., guess-averse.

Definitions Let \( s : X \to \mathbb{R}^K \) be a score vector function; i.e., \( s_k(X^{(1,t)}) \) represents how likely it is that \( X^{(1,t)} \) is sampled from class \( k \). In the LSEL, we can regard \( \log \hat{p}_0(X^{(1,t)})/k \) as \( s_k(X^{(1,t)}) \). A cost matrix \( C \) is a matrix on \( \mathbb{R}^K \times K \) such that \( C_{kl} \geq 0 \) (\( \forall k,l \in [K] \), \( C_{kk} = 0 \) (\( \forall k \in [K] \)), \( \sum_{l \in [K]} C_{kl} 
eq 0 \) (\( \forall k \in [K] \)). \( C \) represents a misclassification cost, or a weight for the loss function, when the true label is \( k \) and the prediction is \( l \). The support set of class \( k \) is defined as \( S_k := \{ v \in \mathbb{R}^K \mid \forall l \neq k, v_l > v_k \} \). Ideally, discriminative score vectors should be in \( S_k \) when the label is \( k \). In contrast, the arbitrary guess set is defined as \( A := \{ v \in \mathbb{R}^K \mid v_1 = v_2 = ... = v_K \} \). If \( s(X^{(1,t)}) \notin A \), we cannot gain any information from \( X^{(1,t)} \); therefore, well-trained discriminative models should avoid such an arbitrary guess of \( s \). We consider a class of loss functions such that \( \ell(s(X^{(1,t)}), y; C) \): It depends on \( X^{(1,t)} \) through the score function \( s \). The loss \( \ell(s(X^{(1,t)}), y; C) \) is guess-averse, if for any \( k \in [K] \), any \( s \in S_k \), any \( s' \in A \), and any cost matrix \( C \), \( \ell(s, k; C) < \ell(s', k; C) \); thus, the guess-averse loss can provide discriminative scores by minimizing it. The empirical loss \( \hat{L} = \frac{1}{MT} \sum_{i=1}^{M} \sum_{t=1}^{T} \ell(s(X^{(1,t)}), y_t; C) \) is said to be guess-averse, if \( \ell \) is guess-averse. The guess-aversion trivially holds for most binary and multiclass loss functions but does not generally hold for cost-sensitive multiclass loss functions due to the complexity of multiclass decision boundaries (Beijbom et al., 2014).

Cost-sensitive LSEL is guess-averse. We define a cost-sensitive LSEL:
\[
\hat{L}_{\text{LSEL}}(\theta, C, S) := \frac{1}{MT} \sum_{i=1}^{M} \sum_{t=1}^{T} C_{yi} \log(1 + \sum_{l \neq y} e^{-\hat{\lambda}_{yl}(X^{(1,t)}, \theta)}),
\]
where \( C_{kl} = C_k \) (\( \forall k,l \in [K] \)). Note that \( \hat{\lambda} \) is no longer an unbiased estimator of the true LLR matrix; i.e., \( \hat{\lambda} \) does not necessarily converge to \( \lambda \) as \( M \to \infty \), except when \( C_k = M/|M_k| (K - 1) \) (\( \hat{L}_{\text{LSEL}} \) reduces to \( \hat{L}_{\text{LSEL}} \)). Nonetheless, the following theorem shows that \( \hat{L}_{\text{LSEL}} \) is guess-averse. The proof is given in Appendix G.1.

Theorem 3.2. \( \hat{L}_{\text{LSEL}} \) is guess-averse, provided that the log-likelihood vector
\[
(\log \hat{p}_0(X^{(1,t)}|y = 1), \log \hat{p}_0(X^{(1,t)}|y = 2),
..., \log \hat{p}_0(X^{(1,t)}|y = K)) \in \mathbb{R}^K
\]
Figure 3. Top: Relative Loss v.s. Training Iteration of LSEL and NGA-LSEL with Two Cost Matrices for Each. Bottom: Averaged Per-Class Error Rate of Last Frame v.s. Training Iteration. Although all the loss curves decrease and converge (top), the error rates of the NGA-LSEL converge slowly and show a large gap depending on the cost matrix, while the error rates of the LSEL converge rapidly, and the gap is small (bottom). “unif.” means \( C_{kl} = 1 \) and “inv. freq.” means \( C_{kl} = 1/M_k \). The dataset is UCF101 (Soomro et al., 2012).

is regarded as the score vector \( s(X^{(1,t)}) \).

Figure 4. MSPRT-TANDEM \((N = 2)\). \( x^{(t)} \) is an input vector; e.g., a video frame. FE is a feature extractor. TI is a temporal integrator, which allows two inputs: the feature vector and a hidden state vector, which encodes the information of the past frames. We use ResNet and LSTM for FE and TI, respectively, but are not limited to them in general. The output posterior densities are highlighted with pink circles. By aggregating the posterior densities, the multiplet loss is calculated. Also, the estimated LLR matrix \( \lambda \) is constructed using the M-TANDEM or M-TANDEMwO formulae. Finally, \( \lambda \) is input to the LSEL. \( L_{\text{LSEL}} + L_{\text{mult}} \) is optimized with gradient descent. In the test phase, \( \hat{\lambda}(X^{(1,t)}) \) is used to execute the MSPRT (Figure 1 and Definition 3.1).

the output of the network \((\hat{p}(y|X^{(1,t)}))\) to the likelihood \(\hat{p}(X^{(1,t)}|y)\) under the \(N\)-th order Markov approximation, which avoids the gradient vanishing of recurrent neural networks (Ebihara et al., 2021):

\[
\hat{\lambda}_{kl}(X^{(1,t)}) = \sum_{s=N+2}^t \log \frac{\hat{p}_0(k|X^{(s-N,s-1)})}{\hat{p}_0(l|X^{(s-N,s-1)})},
\]

where we do not use the prior ratio term \(-\log(\hat{p}(k)/\hat{p}(l)) = -\log(M_k/M_l)\) in our experiments because it plays a similar role to the cost matrix (Menon et al., 2021). Note that (5) is a generalization of the original to DRME, and thus we call it the \(M\)-TANDEM formula.

However, we find that the \(M\)-TANDEM formula contains contradictory gradient updates caused by the middle minus sign. Let us consider an example \(z_i := (X_1^{(1,t)}, y_i)\). The posterior \(\hat{p}_0(y = y_i|X_i^{(s-N,s-1)})\) (appears in (5)) should take a large value for \(z_i\) because the posterior density represents the probability that the label is \(y_i\). For the same reason, \(\hat{\lambda}_{kl}(X^{(1,t)})\) should take a high value; thus \(\hat{p}_0(y = y_i|X^{(1,t)})\) should take a small value in accordance with (5) — an apparent contradiction. These contradictory updates may cause a conflict of gradients and slow the convergence of training, leading to performance deterioration. Therefore, in the experiments, we use either (5) or another approximation formula: \(\hat{\lambda}_{kl}(X^{(1,t)}; \theta) \approx \log(\hat{p}_0(k|X^{(t-N,t)})/\hat{p}_0(l|X^{(t-N,t)}))\), which we call the \(M\)-TANDEM with Oblivion (\(M\)-TANDEMwO) formula. Clearly, the gradient does
We use four datasets: two are new simulated datasets with dense random noise, which is gradually removed (10
we can change the threshold of MSPRT-TANDEM with-
(Dachraoui et al., 2015; Mori et al., 2015; Tavenard & Mali-
ticlass action recognition (UCF101 (Soomro et al., 2012)
and HMDB51 (Kuehne et al., 2011)). A sequence in
made from MNIST (LeCun et al., 2010) (NMNIST-H and
in Figure 4. MSPRT-TANDEM can be used for arbitrary
sequential data and thus has a wide variety of potential
applications, such as computer vision, natural language
processing, and signal processing. We focus on vision tasks in
our experiments.

Note that in the training phase, MSPRT-TANDEM does not require a hyperparameter that controls the speed-accuracy tradeoff. A common strategy in early classification of time series is to construct a model that optimizes two cost functions: one for earliness and the other for accuracy (Dachraoui et al., 2015; Mori et al., 2015; Tavenard & Malinowski, 2016; Mori et al., 2018; Martinez et al., 2020). This approach typically requires a hyperparameter that controls earliness and accuracy (Achenchabe et al., 2020). The tradeoff hyperparameter is determined by heuristics and cannot be changed after training. However, MSPRT-TANDEM does not require such a hyperparameter and enables us to control the speed-accuracy tradeoff after training because we can change the threshold of MSPRT-TANDEM without retraining. This flexibility is an advantage for efficient deployment (Cai et al., 2020).

4. Experiment

To evaluate the performance of MSPRT-TANDEM, we use averaged per-class error rate and mean hitting time: Both measures are necessary because early classification of time series is a multi-objective optimization problem. The averaged per-class error rate, or balanced error, is defined as
\[
1 - \frac{1}{|K|} \sum_{k=1}^{K} \frac{[i \in |M| | h_i = y_i = k]}{|i \in |M| | y_i = k|},
\]
where \( h_i \in [K] \) is the prediction of the model for \( i \in [M] \) in the dataset. The mean hitting time is defined as the arithmetic mean of the stopping times of all sequences.

We use four datasets: two are new simulated datasets made from MNIST (LeCun et al., 2010) (NMNIST-H and NMNIST-100f), and two real-world public datasets for multiclass action recognition (UCF101 (Soomro et al., 2012) and HMDB51 (Kuehne et al., 2011)). A sequence in NMNIST-H consists of 20 frames of an MNIST image filled with dense random noise, which is gradually removed (10 pixels per frame), while a sequence in NMNIST-100f consists of 100 frames of an MNIST image filled with random noise that is so dense that humans cannot classify any video (Appendix K); only 15 of 28 x 28 pixels maintain the original image. The noise changes temporally and randomly and is not removed, unlike in NMNIST-H.

4.1. Models

We compare the performance of MSPRT-TANDEM with four other models: LSTM-s, LSTM-m (Ma et al., 2016), EARLIEST (Hartvigsen et al., 2019), and the Neyman-Pearson (NP) test (Neyman & Pearson, 1933). LSTM-s and LSTM-m, proposed in a pioneering work in deep learning-based early detection of human action (Ma et al., 2016), use loss functions that enhance monotonicity of class probabilities (LSTM-s) and margins of class probabilities (LSTM-m). Note that LSTM-s/m support only the fixed-length test; i.e., the stopping time is fixed, unlike MSPRT-TANDEM. EARLIEST is a reinforcement learning algorithm based on recurrent neural networks (RNNs). The base RNN of EARLIEST calculates a current state vector from an incoming time series. The state vector is then used to generate a stopping probability in accordance with the binary action sampled: Halt or Continue. EARLIEST has two objective functions in the total loss: one for classification error and one for earliness. The balance between them cannot change after training. The NP test is known to be the most powerful, Bayes optimal, and minimax optimal test (Borovkov, 1998; Lehmann & Romano, 2006). The NP test uses the LLR to make a decision in a similar manner to the MSPRT, but the decision time is fixed. The decision rule is
\[
d_{NP}(X^{(1:t)}) := \arg\max_{k \in [K]} \min_{i \in [K]} \lambda_{kl}(X^{(1:t)})
\]
with a fixed \( t \in [T] \). In summary, LSTM-s/m have different loss functions from MSPRT-TANDEM, and the stopping time is fixed. EARLIEST is based on reinforcement learning, and its stopping rule is stochastic. The only difference between the NP test and MSPRT-TANDEM is whether the stopping time is fixed.

We first train the feature extractor (ResNet (He et al., 2016abc)) by solving multiclass classification with the softmax loss and extract the bottleneck features, which are then used to train LSTM-s/m, EARLIEST, and the temporal integrator for MSPRT-TANDEM and NP test. Note that all models use the same feature vectors for the training. For a fair comparison, hyperparameter tuning is carried out with the default algorithm of Optuna (Akiba et al., 2019) with an equal number of tuning trials for all models. Also, all models have the same order of trainable parameters. After fixing the hyperparameters, we repeatedly train the models with different random seeds to consider statistical fluctuation due to random initialization and stochastic optimizers. Finally, we test the statistical significance of the models with the two-way ANOVA followed by the Tukey-Kramer multi-comparison test. More detailed settings are given in Appendix I.
The Power of Log-Sum-Exp: Sequential Density Ratio Matrix Estimation for Speed-Accuracy Optimization

Figure 5. Speed-Accuracy Tradeoff (SAT) Curves. The vertical axis represents the averaged per-class error rate, and the horizontal axis represents the mean hitting time. Early and accurate models come in the lower-left area. The vertical error bars are the standard error of mean (SEM); however, some of the error bars are too small and are collapsed. **Upper left:** NMNIST-H. The Neyman-Pearson test, LSTM-s, and LSTM-m almost completely overlap. **Upper right:** NMNIST-100F. LSTM-s and LSTM-m completely overlap. **Lower left:** UCF101. Lower right: HMDB51.

### 4.2. Results

The performances of all the models are summarized in Figure 5 (The lower left area is preferable). We can see that MSPRT-TANDEM outperforms all the other models by a large margin, especially in the early stage of sequential observations. We confirm that the results have statistical significance; i.e., our results are reproducible (Appendix L). The loss functions of LSTM-s/m force the prediction score to be monotonic, even when noisy data are temporally observed, leading to a suboptimal prediction. In addition, LSTM-s/m have to make a decision, even when the prediction score is too small to make a confident prediction. However, MSPRT-TANDEM can wait until a sufficient amount of evidence is accumulated. A potential weakness of EARLIEST is that reinforcement learning is generally unstable during training, as pointed out in (Nikishin et al., 2018; Kumar et al., 2020). The NP test requires more observations to attain a comparable error rate to that of MSPRT-TANDEM, as expected from the theoretical perspective (Tartakovsky et al., 2014): In fact, the SPRT was originally developed to outperform the NP test in sequential testing (Wald, 1945; 1947).

### 5. Conclusion

We propose the LSEL for DRME, which has yet to be explored in the literature. The LSEL relaxes the crucial assumption of the MSPRT and enables its real-world applications. We prove that the LSEL has a theoretically strong background: consistency, hard class weighting, and guess-aversion. We also propose MSPRT-TANDEM, the first DRE-based model for early multiclass classification in deep learning. The experiment shows that the LSEL and MSPRT-TANDEM outperform other baseline models statistically significantly.

### Acknowledgments

The authors thank the anonymous reviewers for their careful reading to improve the manuscript. We would like to thank Jiro Abe, Genki Kusano, Yuta Hatakeyama, Kazuma Shimizu, and Natsuhiko Sato for helpful comments on the
proof of the LSEL’s consistency.

**References**


He, H. and Garcia, E. A. Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering, 21(9):1263–1284, 2009. doi: 10.1109/TKDE.2008.239.


The Power of Log-Sum-Exp: Sequential Density Ratio Matrix Estimation for Speed-Accuracy Optimization


