# ONLINE LOGISTIC REGRESSION ON MANIFOLDS

*Yao Xie and Rebecca Willett*

Electrical and Computer Engineering, Duke University

## ABSTRACT

This paper describes a new method for online logistic regression when the feature vectors lie close to a low-dimensional manifold and when observations of the feature vectors may be noisy or have missing elements. The new method exploits the low-dimensional structure of the feature vector, finds a multi-scale union of linear subsets that approximates the manifold, and performs online logistic regression separately on each subset. The union of subsets enables better performance in the face of noisy and missing data, and offsets challenges associated with the curse of dimensionality. The effectiveness of the proposed method in predicting correct labels of the data and in adapting to slowly time-varying manifolds are demonstrated using numerical examples and real data.

***Index Terms*—** Online learning, manifold learning, subspace tracking, logistic regression, big data

## 1. INTRODUCTION

Logistic regression [1] is a highly effective technique for supervised learning of classifiers. It is widely used in data mining, cancer classification using microarray data [2], computational advertising, phishing email scams, and social network analysis [3]. In recent years, online algorithms have become a popular approach to logistic regression for large-scale problems [4, 5], due to its simple implementation and effectiveness in processing massive quantities of data.

In many scenarios, we have to process streaming data $(x_t, y_t)$, where $x_t \in \mathbb{R}^D$ is a "feature vector" with large $D$ and $y_t \in \{0, 1\}$ is a label. The goal is to learn a classifier via updates based on training pairs, so that when only $x_t$ is available we may predict the label $\hat{y}_t$. When the feature dimension $D$ is large, this task can be particularly challenging due to the well-known curse of dimensionality. We can side-step this problem by exploiting low-dimensional structure underlying $x_t$. In many settings, however, this low-dimensional structure must be updated from streaming data and may be slowly changing over time. We must update the model to track the dynamic structure. Moreover, missing data is frequently encountered in high-dimensional data analysis: it is often impossible or too costly to obtain complete data, as in internet tomography [6]. The primary challenges are to find this (possibly dynamic) low-dimensional structure within a high-dimensional feature space and exploit this in the logistic regression framework using streaming data.

In this paper, we develop a new method for multiscale online logistic regression for data on dynamic manifolds, which we call *MOUSSE-Logit*. It regresses on low-dimensional linear subsets that are approximations of the manifold; moreover, it is able to handle noisy and missing elements in the feature vectors. Our recent work on Multiscale Online Union-of-Subsets Estimation (MOUSSE) algorithm enables online estimation and tracking of a set of linear approximations to the manifold. Based on this algorithm, we project data onto the local linear approximations, and perform logistic regression in the lower dimensional projected space. We show that when the feature vector has an intrinsic manifold structure, MOUSSE-Logit can: (1) regress on the low-dimensional space with much lower computational cost; (2) achieves better performance working in a low-dimensional space than logistic regression in the ambient space; and (3) achieves better performance than logistic regression based on projecting feature vectors on a single subspace. Numerical examples and real-data experiments demonstrate the strong performance of MOUSSE-Logit.

Related work includes sparse logistic regression [7], which imposes an $\ell_1$ penalty on the logistic parameter in regression, so that the number of features (entries in feature vector) used in regression is small. In contrast, MOUSSE regresses on learned projections of the feature vectors. In this sense, MOUSSE-Logit is closer to logistic regression based on the batch (not online) approach of sufficient dimension reduction [8].

## 2. FORMULATION

### 2.1. Model

Suppose we are given training data $\{(x_{\Omega_t}, y_t, \Omega_t)\}$, $t = 1, 2, \cdots$. The observed feature vector $x_{\Omega_t}$ is defined as follows. Assume the "ideal" or true feature vector $v_t$ lies on submanifold $\mathcal{M}_t \subset \mathbb{R}^D$, and let $x_t$ be a noisy realization of $v_t$:

$$x_t = v_t + w_t, \tag{1}$$

where $w_t$ is a Gaussian noise with zero mean and variance $\sigma^2$. Here $D$ denotes the *ambient dimension*. The *intrinsic dimension* of the submanifold $\mathcal{M}_t$ is $d$; we assume $d \ll D$. The underlying submanifold $\mathcal{M}_t$ may vary slowly with time. At each time $t$ we only observe $x_{\Omega_t}$ on a subset of indices $\Omega_t$. Let $\mathcal{P}_{\Omega_t}$ represent the $|\Omega_t| \times D$ matrix that selects the axes of $\mathbb{R}^D$ indexed by $\Omega_t$; we observe

$$x_{\Omega_t} \triangleq \mathcal{P}_{\Omega_t} x_t. \tag{2}$$

The label $y_t \in \{0, 1\}$ may not be observed for test data.

Assume the labels $\{y_t\}$ are generated according to a Logistic model described as follows [9]. Define the logistic function as

$$h(w; p, b) = \frac{1}{1 + e^{-p^T w - b}}, \tag{3}$$

This work was supported by DARPA award # FA8650-11-1-7150, # HR0011-10-C-0073, AFOSR award # FA9550-10-1-0390, AFOSR award # FA9550-11-1-0028, NGA # HM01773-1-0006.

where $w \in \mathbb{R}^m$ is a vector of features, $p \in \mathbb{R}^m$ denotes the weight vector, $b \in \mathbb{R}$ denotes an intercept or offset, and the notation $^\top$ denotes transpose of a matrix or vector. We assume the labels $\{y_t\}$ are Bernoulli with probability $p$ of being 1 that depends on $v_t$:

$$\mathbb{P}(y_t = 1) = h(v_t; \theta, b), \tag{4}$$

where $\theta \in \mathbb{R}^D$ is the weight vector on the feature vector, and $b \in \mathbb{R}$ is the intercept.

Our goal is to use online logistic regression to design a classifier based on streaming training data $\{(x_{\Omega_t}, y_t, \Omega_t)\}$ which exploits the low-dimensional submanifold structure of $v_t$.

### 2.2. Attribute and relational data

Our formulation is motivated by the following two types of problems. In the first scenario, we may want to infer the label of data according to its attribute. For example, in a recommendation system, $x_t$ is the feature vector of a customer, $y_t$ indicates whether a customer likes a given product. In a reputation system (say credit scoring system), $x_t$ are the features of a user, and $y_t$ is a label indicating whether the user is reliable or not.

In the second scenario, we may have relational data for a pair of users $(n_t, m_t)$, which consist of their feature vectors $z_{n_t} \in \mathcal{X}$ in space $\mathcal{X}$, $z_{m_t} \in \mathcal{X}$ and a label $y_t$ that takes value 1 if user $n_t$ and user $m_t$ are related (e.g., they are friends) at time $t$. Then we would like to infer whether two people for whom we have not observed relationship indicators are likely to become friends. This scenario can be mapped to the first scenario. We can define a metric that compares two feature vectors (for example, take entry-wise difference): $\text{dist}(z_{n_t}, z_{m_t}) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^D$. Then we may use $x_t = \text{dist}(z_{n_t}, z_{m_t})$ as features in logistic regression.

## 3. LOGISTIC REGRESSION

In the following, we will present two approaches for logistic regression: regression in the original $\mathbb{R}^D$ feature space, and regression in the $\mathbb{R}^d$ projected feature domain.

### 3.1. Logistic regression in $\mathbb{R}^D$ feature space

With training data $(x_{\Omega_t}, y_t, \Omega_t)_{t=1}^N$, we can find parameters $(\theta, b)$ to the logistic regression model (4) by maximizing the log-likelihood function with observed data entries, which is given by

$$\begin{aligned}\ell(\theta, b; \{x_{\Omega_t}, y_t, \Omega_t\}) \triangleq \sum_{t=1}^{N} \{ & y_t \log h(x_{\Omega_t}; \theta_{\Omega_t}, b) \\ & + (1 - y_t) \log[1 - h(x_{\Omega_t}; \theta_{\Omega_t}, b)]\},\end{aligned} \tag{5}$$

where $\theta_{\Omega_t} = \mathcal{P}_{\Omega_t}\theta$.

In the online setting, we discuss how to update estimates for parameters $(\theta, b)$. Let the estimated parameters at time $t$ be $(\theta_t, b_t)$. At time $t+1$, when there is new data with feature vector $x_{t+1}$ and unknown label, we can predict its label using

$$\hat{y}_{t+1} = \begin{cases} 1, & h(x_{\Omega_{t+1}}; \theta_{\Omega_{t+1}}, b_t) \geq 1/2; \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

When both the feature vector $x_{t+1}$ and the label $y_{t+1}$ are available, we can update $(\theta_t, b_t)$ by stochastic gradient descent. Consider the following one-sample log-likelihood function with data $(x, y)$:

$$\ell(\theta, b; x, y) \triangleq y \log h(x; \theta, b) + (1 - y) \log(1 - h(x; \theta, b)). \tag{7}$$

Let $[w]_m$ denote the $m$-th element of vector $w$. The gradient of (7) is given by

$$\frac{\partial \ell(\theta, b; x, y)}{\partial [\theta]_m} = -(h(x; \theta, b) - y)[\theta]_m, \quad m = 1, \ldots, D, \tag{8}$$

$$\frac{\partial \ell(\theta, b; x, y)}{\partial b} = -(h(x; \theta, b) - y) \tag{9}$$

With missing data, we can update the parameter $\theta_t$ on the dimensions where the feature vector are observed:

$$\begin{aligned} \theta_{t+1} &= \theta_t + \eta(h(x_{\Omega_{t+1}}; \theta_{\Omega_{t+1}}, b_t) - y_{t+1})[x_{t+1}]_m, \quad m \in \Omega_{t+1}, \\ b_{t+1} &= b_t + \eta(h(x_{\Omega_{t+1}}; \theta_{\Omega_{t+1}}, b_t) - y_{t+1}), \end{aligned} \tag{10}$$

where $\eta > 0$ is the step-size.

### 3.2. Logistic regression in $\mathbb{R}^d$ projection space

When feature vectors lie in a low-dimensional space, we can exploit this structure to reduce computational complexity and achieve better predication performance. Note that label $y_t$ depends on $x_t$ through the logistic function with argument $\theta^\top x_t$, so in the following we focus on this inner product $\theta^\top x_t$.

First we consider a special case, and demonstrate that when there is no noise in the observed feature vector $x_{\Omega_t} = v_t$ (i.e., $\Omega_t = \{1, \ldots, D\}$ so there is no missing data), and the feature vector $v_t$ lies in a $d$-dimensional subspace embedded in $\mathbb{R}^D$, then logistic regression in the ambient space and the projected space are identical. In particular, assume $v_t = U\beta$, for some orthogonal basis $U \in \mathbb{R}^{D \times d}$, where $\beta = U^\top v_t \in \mathbb{R}^d$ is the coefficient of $v_t$. Let $a = U^\top \theta \in \mathbb{R}^d$ be the projection of $\theta$ on the linear subspace. In the following, we will show

$$\theta^\top v_t = a^\top \beta. \tag{11}$$

To see this, we decompose $\theta$ into a component that lies in $U$, $\theta_\parallel$, and a component that is orthogonal to $U$, $\theta_\perp$, with $\theta = \theta_\parallel + \theta_\perp$. Note that $\theta_\parallel = Ua$, $\theta_\perp = (I - UU^\top)\theta$. By this decomposition, $\theta^\top v_t = {\theta_\parallel}^\top v_t + \theta_\perp^\top v_t$, with

$$\theta_\parallel^\top v_t = a^\top U^\top U \beta = a^\top \beta, \tag{12}$$

$$\theta_\perp^\top v_t = \theta^\top (I - UU^\top) U\beta = \theta^\top (U - UU^\top U)\beta = 0, \tag{13}$$

since $U^\top U = I$. Hence, we have (11).

Since $y_t$ depends on $v_t$ through the logistic function with argument $\theta^\top v_t$, the above argument demonstrates that with knowledge of the subspace $U$, we can first project $v_t$ onto the subspace, and then perform logistic regression in the projection space.

The benefit of performing logistic regression in the projection space is two-fold: first, when the observed feature vector is noisy ($x_{\Omega_t} = \mathcal{P}_{\Omega_t}(v_t + w_t)$), knowledge of $U$ helps with denoising and imputing missing elements; second, projection significantly lowers computational complexity: instead of estimating $D+1$ parameters $(\theta, b)$ in ambient space, in the projection space we only need to estimate $d + 1$ parameters $(a, b)$.

When $v_t$ lies on a low-dimensional manifold rather than a linear subspace, we can use a union of linear subsets to approximate the manifold, and project feature vectors onto the union of subsets. The above arguments motivate our new algorithm described in Section 4.

The main computational cost of MOUSSE-Logit comes from computing the subspace tracking, which is on the order of $\mathcal{O}(dD)$ [10], higher than computing standard online logistic regression (complexity $\mathcal{O}(D)$). The memory requirement for MOUSSE-Logit, dominated by storing the subspace, is on the order of $\mathcal{O}(Dd)$, higher than online logistic regression (where the memory requirement dominated by storing the weight vector $\mathcal{O}(D)$). However, the benefits of MOUSSE-Logit include imputing the missing data using subspace projection and achieving better performance by denoising, as demonstrate in Section 5.

## 4. MOUSSE-LOGIT ALGORITHM

In this section, we present the new algorithm, MOUSSE-Logit, which approximates the manifold in the feature vector space by union of linear subsets that are organized in a tree structure, projects the feature vectors onto the linear subsets, and then fits a logistic model using projected feature vectors. To obtain the approximation to the manifold, and update the estimate using streaming data, we use our recently developed Multiscale Online Union of Subspace Estimation (MOUSSE) algorithm [10]. To update logistic regression parameters, we develop a stochastic gradient descent method based on projected feature vectors. The MOUSSE-Logit algorithm is summarized in Algorithm 1.

### 4.1. Union-of-subset approximation

MOUSSE uses a union of low-dimensional subsets $\widehat{\mathcal{M}}_t$ to approximate $\mathcal{M}_t$, and organizes these subsets using a tree structure. The idea for a multiscale tree structure is drawn from the multiscale harmonic analysis literature [11]. The leaves of the tree are subsets that are used for the submanifold approximation. Each node in the tree represents a local approximation to the submanifold at one scale. The parent nodes are subsets that contain coarser approximations to the submanifold than their children.

More specifically, the approximation at each time $t$ consists of a union of subspaces $\mathcal{S}_{j,k,t}$ that is organized using a tree structure. Here $j \in \{1, \ldots, J_t\}$ denotes the scale or level of the subset in the tree, where $J_t$ is the tree depth at time $t$, and $k \in \{1, \ldots, 2^j\}$ denotes the index of the subset for that level. The approximation $\widehat{\mathcal{M}}_t$ at time $t$ is given by:

$$\widehat{\mathcal{M}}_t = \bigcup_{(j,k)\in\mathcal{A}_t} \mathcal{S}_{j,k,t}, \tag{14}$$

where $\mathcal{A}_t$ contains the indices of all leaf nodes used for approximation at time $t$. Also define $\mathcal{T}_t$ to be the set of indices of all nodes in the tree at time $t$, with $\mathcal{A}_t \subset \mathcal{T}_t$. Each of these subsets lies on a low-dimensional hyperplane with dimension $d$ and is parameterized as

$$\begin{aligned}\mathcal{S}_{j,k,t} = \{v \in \mathbb{R}^D : \quad & v = U_{j,k,t} z + c_{j,k,t}, \\ & z^\top \Lambda_{j,k,t}^{-1} z \le 1, \quad z \in \mathbb{R}^d\}.\end{aligned} \tag{15}$$

The matrix $U_{j,k,t} \in \mathbb{R}^{D\times d}$ is the subspace basis, and $c_{j,k,t} \in \mathbb{R}^D$ is the offset of the subset from the origin. The diagonal matrix

$$\Lambda_{j,k,t} \triangleq \text{diag}\{\lambda_{j,k,t}^{(1)}, \ldots, \lambda_{j,k,t}^{(d)}\} \in \mathbb{R}^{d\times d},$$

with $\lambda_{j,k,t}^{(1)} \ge \ldots \ge \lambda_{j,k,t}^{(d)} \ge 0$, contains eigenvalues of the covariance matrix of the projected data onto each subset. This parameter specifies the shape of the ellipsoid by capturing the spread of the data within the subset. Each node also has a parameter $\delta_{j,k,t}$ measuring the approximation error associated with each subset and is used in evaluating the distance of a sample to the subset.

MOUSSE estimates this tree of subsets by balancing the approximation errors and the number of subsets used for approximation to prevent data-overfitting (more details in [10]).

### 4.2. Online update of parameters and tree structure

MOUSSE-Logit uses the same tree structure as MOUSSE, and it also includes two more parameters for each node: $a_{j,k,t}$ and $b_{j,k,t}$, which are the slope and intercept of the logistic model fitted using data associated and projected onto the subset $\mathcal{S}_{j,k,t}$. In summary, the parameters used in MOUSSE-Logit are

$$\{U_{j,k,t}, c_{j,k,t}, \Lambda_{j,k,t}, \delta_{j,k,t}, a_{j,k,t}, b_{j,k,t}\}_{(j,k)\in\mathcal{T}_t}. \tag{16}$$

For initialization of these parameters $\{U_{j,k,t}, c_{j,k,t}, \Lambda_{j,k,t}, \delta_{j,k,t}\}$ in (16) and the tree structure, we apply MOUSSE initialization on the training feature vectors $\{(x_t, \Omega_t)\}$. To initialize the logistic regression parameters $(a_{j,k}, b_{j,k})$, we use projected data $\{(\beta_t, y_t)\}$ to fit a logistic model.

When a new sample $(x_{t+1}, y_{t+1}, \Omega_{t+1})$ becomes available, MOUSSE-Logit updates the parameters, and the tree structure if necessary. There are three main steps: (a) find the subset in the $\widehat{\mathcal{M}}_t$ which is closest to $x_{t+1}$; (b) use MOUSSE to update $\widehat{\mathcal{M}}_t$ by updating parameters and the tree structure; (c) using stochastic gradient descent to update logistic parameter $(a_{j,k,t}, b_{j,k,t})$ of the closest subset and all its ancestor.

More details about the first two steps can be found in [10]. In the following we focus on updating of the logistic regression parameters, which can be achieved by performing stochastic gradient descent using the log-likelihood function (19) as the cost function. Given the union of subsets approximation estimated by MOUSSE, to update logistic parameters or predict label, we first project the feature vector with missing entries on to a subset with minimum distance. The distance metric used is the approximate Mahalanobis distance [10], since it captures the low-dimensional structure of the data. Denote the pseudoinverse operator that computes the coefficients of a vector in the subspace spanned by $V$ as $V^\# \triangleq (V^\top V)^{-1} V^\top$. When $U$ is an orthogonal matrix, we have $U^\# \equiv U^\top$. Given a subset with parameters $\{U, c, \Lambda, \delta\}$, we compute the projection coefficient of $x_t$ with support $\Omega_t$ onto the subset as

$$\beta_t = U_{\Omega_t}^\# (x_{\Omega_t} - c_{\Omega_t}). \tag{17}$$

This projection essentially imputes the missing entries using the subspace rather than filling in zeros. Then the orthogonal component is given by $\beta_t^\perp = (I - U_{\Omega_t} U_{\Omega_t}^\#)(x_{\Omega_t} - c_{\Omega_t})$. Using the projections, the approximate Mahalanobis distance of $x_t$ to the subset is defined as

$$\rho_\delta(x_t, \mathcal{S}) \triangleq \beta_t^\top \Lambda^{-1} \beta_t + \delta^{-1} \|\beta_t^\perp\|^2, \tag{18}$$

which combines Mahalanobis distance (commonly used in classification) and the Euclidean distance.

In the following, let $\beta_t$ denote the projection coefficients of $x_{\Omega_t}$ projected onto the corresponding minimum distance subset. Using the projected training feature vectors, we can estimate the

logistic parameters by maximizing the following log-likelihood function

$$\ell(a,b;\{\beta_t,y_t\}) \triangleq \sum_{t=1}^{N} y_t \log h(\beta_t;a,b) + (1-y_t)\log[1-h(\beta_t;a,b)]. \tag{19}$$

In the online setting, we discuss how to update estimates for parameters $(a,b)$. Let the estimated parameters at time $t$ be $(a_t,b_t)$. At time $t+1$, when there is new data with feature vector $x_{t+1}$ and unknown label, we can predict its label using

$$\hat{y}_{t+1} = \begin{cases} 1, & h(\beta_{t+1};a_t,b_t) \geq 1/2; \\ 0, & \text{otherwise.} \end{cases} \tag{20}$$

When both the feature vector $x_{\Omega_{t+1}}$ and the label $y_{t+1}$ are available, we can update $(a_t,b_t)$ by stochastic gradient descent, by finding the minimum distance subset, computing the projection $\beta_{t+1}$, and updating the logistic regression parameters using stochastic gradient descent to maximize the log likelihood function. Let $(a_t,b_t)$ denote the parameter at time $t$, we can update these parameters by

$$\begin{aligned} [a_{t+1}]_m &= [a_t]_m - \eta(h(\beta_{t+1};a_t,b_t)-y_t)[\beta_{t+1}]_m, m=1,\ldots,d \\ b_{t+1} &= b_t - \eta(h(\beta_{t+1};a_t,b_t)-y_t), \end{aligned} \tag{21}$$

where $\eta > 0$ is the step-size.

**Algorithm 1** MOUSSE-Logit

1: Initialize parameters (16) and tree structure
2: **for** $t = 1,2,\ldots$ **do**
3: Given data $x_{\Omega_t}$, its support $\Omega_t$ (and for training, label $y_t$)
4: Find the set with minimum approximate Mahalanobis distance distance (18) to $x_{\Omega_t}$
5: Compute projection $\beta_t$ (17) using parameters of minimum distance set $(U^*,c^*,\Lambda^*,\delta^*)$
6: If label is not available, predict label via (20) using $(a^*,b^*)$ of the minimum distance set
7: Update parameters $(U^*,c^*,\Lambda^*,\delta^*)$ using MOUSSE
8: If label is available, update logistic regression parameters of the minimum distance set and all its ancestors using (21).
9: **end for**

## 5. NUMERICAL EXAMPLES

We compare the average error in predicting labels using three methods: (1) Online logistic regression in $\mathbb{R}^D$ (described in Section 3.1); (2) Online logistic regression in $\mathbb{R}^d$ by projecting feature vectors on an estimated single subspace, which corresponds to restricting $K = 1$ in MOUSSE; (3) MOUSSE-Logit in $\mathbb{R}^d$.

The manifold we generate is a chirp-signal, with ambient dimension $D = 100$ and intrinsic dimension $d = 2$. The two-dimensional parameters $[f_0,\Phi]$ are uniformly random generated: frequency $f_0 \in [1,100]$, and phase $\Phi \in [0,1]$. Define $v_t \in \mathbb{R}^D$ with its $n$-th element being

$$[v_t]_n = \sin\left[2\pi(f_0 z_n + \frac{k_t^2}{2} z_n^2 + \Phi)\right], \tag{22}$$

where the indices, $z_n = 10^{-4}n, n = 1,2,\ldots,100$, corresponds to regularly spaced samples between 0 and 0.01. The parameter $k_t$ controls how fast the submanifold changes:

$$k_t = \begin{cases} 0.1t & t = 1,2,\ldots,1000, \\ 0.1(t-1000) & t = 1001,\ldots,2000, \\ 200-0.1(t-1000) & t = 2001,\ldots,3000. \end{cases}$$

For MOUSSE, we use PETRELS-FO for tracking [10], and parameter values are $\epsilon = 0.3$, $\mu = 0.02$, $\alpha = 0.9$.

We use the first 1000 samples generated by (22) for training, and the remaining 2000 samples for test. We perform training with a batch method using a variant of $k$-means, as detailed in [10], followed by batch logistic regression on the learned submanifold approximation. Then for the test procedure, we first use each of the three algorithms to predict label $\hat{y}_t$, then reveal the true label and use $(x_{\Omega_t},y_t,\Omega_t)$ to update the classifier.

The performance metric is given by average prediction error on the test samples: $P_e = T^{-1}\sum_{t=1001}^{T+1000}\mathbb{I}\{\hat{y}_t \neq y_t\}$, $T = 2000$. In the following, we average $P_e$ over 50 independent trials; each trial is a sequence generated by (22).

We also compare these methods on real data for credit card fraud detection: the user E-commerce transaction history data set in 2008 UCSD data mining competition[1]. We choose an arbitrary sequence from the data set, which corresponds to the spending pattern of one user over time. The series has 144 samples, and each sample is a transaction record that consists of a $D = 41$ dimensional feature vector and a label that indicating if the transaction is normal ($y_t = 0$) or a fraud ($y_t = 1$). We randomly permute the order of these samples for each of our 50 trials, and use 72 samples for training and the remaining 72 samples for test. The parameters for MOUSSE-Logit are $d = 3$, $\epsilon = 15$, $\mu = 0.05$, $\alpha = 0.9$. We obtain $P_e$ for each of 50 random permutations, and then average $P_e$ across these trials. The comparison results are shown in Table 1, where in the first column $\sigma$ is the noise variance and the second number is the percentage of missing entries in the feature vector. The credit card data does not have missing entries or known noise statistics. In all cases, MOUSSE-Logit outperforms the other two methods, especially in the real data experiment.

**Table 1**: Comparison of average $P_e \pm$ std

| Dataset | Online Logistic | Single Subspace Online Logistic | MOUSSE-Logit |
|---|---|---|---|
| $\sigma = 0$, 60% | 0.0559 ±0.0323 | 0.2053 ±0.0730 | **0.0410** ±0.0182 |
| $\sigma = 0.1$, 40% | 0.0455 ±0.0275 | 0.1716±0.0616 | **0.0343** ±0.0179 |
| credit data | 0.3883 ±0.1706 | 0.1389 ±0.0995 | **0.1107**±0.0615 |

## 6. CONCLUSIONS

We have presented a new method, MOUSSE-Logit, for online logistic regression when the feature vectors lie close to a manifold with intrinsic dimension $d$ much smaller than the ambient dimension $D$. MOUSSE-Logit has low complexity (proportional to $d$) and good performance, as demonstrated using numerical examples and a real-data experiment.

**Acknowledgement**

We thank Jiaji Huang for his help on part of numerical examples.

[1] Data available at `http://www.cs.purdue.edu/commugrate/data_access/all_data_sets_more.php?search_fd0=20` More detail about the data can be found in [10].

## 7. REFERENCES


[1] D. W. Hosmer and S. Lemeshow, *Applied Logistic Regression*, Wiley, 2000.

[2] L. Shen and E. C. Tan, "Dimension redunction-based penalized logistic regression for cancer classification using microarray data", *IEEE Trans. Computational Biology and Bioimformatics*, vol. 2, no. 2, pp. 166 – 175, 2005.

[3] R. Xi, N. Lin, and Y. Chen, "Compression and aggregation for logistic regression analysis in data cubes", *IEEE Trans. on Knowledge and data engineering*, vol. 1, no. 1, pp. 1 – 14, Aug. 2008.

[4] W. D. Penny and S. J. Roberts, "Dynamic logistic regression", in *Neural Networks, 1999. IJCNN '99. International Joint Conference on*, 1999, vol. 3, pp. 1562– 1567.

[5] H. B. McMahan and M. Streeter, "Open problem: Better bounds for online logistic regression", in *COLT/ICML Joint Open Problem Session, JMLR: Workshop and Conference Proceedings*, 2012.

[6] L. Balzano, B. Eriksson, and R. Nowak, "High rank matrix completion and subspace clustering with missing data", in *Proc. of Conf. Artificial Intelligence and Stats (AIStats)*, 2012.

[7] K. Koh, S.-J. Kim, and S. Boyd, "An interior-point method for large scale $\ell_1$-regularized logistic regression", *J. of Machine Learing Res.*, vol. 8, pp. 1519 – 1555, 2007.

[8] J. Nilsson, F. Sha, and M. Jordan, "Regression on manifolds using kernel dimension reduction", in *Proc. Inter. Conf. Machine Learning (ICML)*, 2007, pp. 697 – 704.

[9] A. Agresti, *Categorical data analysis*, Wiley-Interscience, 2002.

[10] Y. Xie, J. Huang, and R. Willett, "Changepoint detection for high-dimensional time series with missing data", *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 1, pp. 12–27, Feb. 2013, http://people.ee.duke.edu/~yx44/MOUSSE.pdf.

[11] D. Donoho, "Cart and best-ortho-basis selection: A connection", *Annals of Stat.*, vol. 25, pp. 1870–1911, 1997.