Naive-Bayes Inspired Effective Pre-Conditioner for Speeding-up Logistic Regression

Nayyar A. Zaidi, Mark J. Carman
Faculty of Information Technology
Monash University
VIC 3800, Australia
{*nayyar.zaidi,mark.carman*}*@monash.edu*

Jesús Cerquides
IIIA-CSIC
Artificial Intelligence Research Institute
Spanish National Research Council
Campus UAB, 08193 Bellaterra, Spain
Email: cerquide@iiia.csic.es

Geoffrey I. Webb
Faculty of Information Technology
Monash University
VIC 3800, Australia
geoff.webb@monash.edu

***Abstract*—We propose an alternative parameterization of Logistic Regression (LR) for the categorical data, multi-class setting. LR optimizes the conditional log-likelihood over the training data and is based on an iterative optimization procedure to tune this objective function. The optimization procedure employed may be sensitive to scale and hence an effective pre-conditioning method is recommended. Many problems in machine learning involve arbitrary scales or categorical data (where simple standardization of features is not applicable). The problem can be alleviated by using optimization routines that are invariant to scale such as (second-order) Newton methods. However, computing and inverting the Hessian is a costly procedure and not feasible for big data. Thus one must often rely on first-order methods such as gradient descent (GD), stochastic gradient descent (SGD) or approximate second-order such as quasi-Newton (QN) routines, which are not invariant to scale. This paper proposes a simple yet effective pre-conditioner for speeding-up LR based on naive Bayes conditional probability estimates. The idea is to scale each attribute by the log of the conditional probability of that attribute given the class. This formulation substantially speeds-up LR's convergence. It also provides a weighted naive Bayes formulation which yields an effective framework for hybrid generative-discriminative classification.**

***Keywords*-classification, logistic regression, pre-conditioning, weighted naive Bayes, stochastic gradient descent, discriminative/generative learning.**

I. Introduction

Logistic Regression (LR) is a simple and yet highly effective linear classifier based on a linear log-odds assumption. It's use is almost ubiquitous throughout machine learning and statistics. In this paper, we develop an alternative parameterisation of LR for the multi-class categorical data setting, in which we rescale the parameters of the LR model by the log probabilities of the attribute given the class, i.e. by the logarithm of their corresponding Naive Bayes' parameters. This reparameterisation can also be interpreted as a form of over-parameterized weighted Naive Bayes with discriminatively trained weights. The reparameterisation provides a form of diagonal pre-conditioning for the parameter estimation procedure and leads to substantial speed-ups in the convergence of the Logistic Regression model. The accelerated convergence is observed for both Quasi-Newton (QN) and Stochastic Gradient Descent (SGD) based parameter estimation, two widely used techniques in the batch and online setting respectively.

The main contributions of this paper are summarized as follows:

- We show that the new parameterization, called WANBIA-C, has an important advantage relative to LR on categorical data. It has, as expected, near-identical generalization performance, but much faster convergence as compared to LR. Faster convergence is obviously desirable when learning from large quantities of data.
- We show better convergence for both batch Quasi Newton (QN) and online Stochastic Gradient Descent (SGD) based optimization. The latter is often used for learning with a single pass through the training data. In this case, the new parameterization effectively allows for different step sizes in different directions and thus provides for faster SGD convergence without needing to adaptively set the coordinate-wise step size.
- We believe the reason for the speed-up is that the information from the generatively learned parameters serves as an effective pre-conditioner for the discriminative parameter estimation process. We compare with other pre-conditioning methods and find the new parameterization to outperform them.
- Being a reparameterization of LR, it produces well-calibrated probability estimates and, therefore, generally results in more accurate classifiers than, for example Naive Bayes (NB), weighted Naive Bayes, and related methods, especially for large datasets.
- We demonstrate that adding a regularization term to the WANBIA-C objective function provides a framework to smoothly interpolates between generatively (NB) and discriminatively (LR) trained parameters.

II. Reparameterizing Logistic Regression

LR optimizes the Conditional Log Likelihood (CLL) objective function over all labeled training instances:

$$\mathrm{CLL} = \sum_{\langle \mathbf{x}, y\rangle \in \mathcal{D}} \log \hat{\mathrm{P}}(y|\mathbf{x}),$$

where

$$\begin{aligned}\hat{\mathrm{P}}(y|\mathbf{x}) &= \frac{\exp(\beta_{y0} + \sum_i \beta_{i,y} x_i)}{\sum_{y'} \exp(\beta_{y'0} + \sum_i \beta_{i,y'} x_i)}, \\ \log \hat{\mathrm{P}}(y|\mathbf{x}) &= \beta_{y0} + \sum_i \beta_{i,y} x_i - \\ & \log(\sum_{y'} \exp(\beta_{y'0} + \sum_i \beta_{i,y'} x_i)).\end{aligned}$$

If $\mathbf{x}$ is a vector of categorical feature values – the LR model must implicitly "binarize" each attribute i, effectively introducing one parameter β_{y,i,x_i} per attribute value x_i, per class y. Let us assume an extra attribute which has a value of one for each data point. Therefore, for data with categorical attributes, the last equation can be written as:

$$\log \hat{\mathrm{P}}(y|\mathbf{x}) = \sum_i \beta_{y,i,x_i} - \log(\sum_{y'} \exp(\sum_i \beta_{y',i,x_i})). \quad (1)$$

Based on Equation 1, LR gradient is defined as:

$$\frac{\partial \mathrm{CLL}}{\partial \beta_{k,i,j}} = \sum_{\langle \mathbf{x},y \rangle \in \mathcal{D}} \left(\mathbf{1}_{y=k} - \hat{\mathrm{P}}(k|\mathbf{x}) \right) \mathbf{1}_{x_i=j}, \quad (2)$$

where $\mathbf{1}_{a=b}$ is an indicator function.

Iterative optimization algorithms for LR are based on the update equation: $\beta_{t+1} = \beta_t + \alpha_t p_t$, where α_t is a positive scalar characterizing the step size and p_t takes the following form: $p_t = -B_t^{-1} \nabla f_t$, where ∇f_t is the gradient vector with entries $\frac{\partial \mathrm{CLL}}{\partial \beta_{k,i,j}}$ and B_t is a symmetric and non-singular matrix and takes the form of an identity matrix I in case of gradient descent and a Hessian matrix in case of Newton method.

Let us reparameterize β in Equation 1 by introducing parameter w's as:

$$\beta_{y,i,x_i} = w_{y,i,x_i} \log \theta_{x_i|i,y}. \quad (3)$$

where $\theta_{x_i|i,y}$ is the MAP estimates of $\hat{\mathrm{P}}(x_i|y)$ [1]. Now Equation 1 can be written as:

$$\begin{aligned}\log \hat{\mathrm{P}}(y|\mathbf{x}) &= \sum_i w_{y,i,x_i} \log \theta_{x_i|i,y} - \\ & \log(\sum_{y'} \exp(\sum_i w_{y',i,x_i} \log \theta_{x_i|i,y})). (4)\end{aligned}$$

The gradient w.r.t w can be computed with chain rule as:

$$\begin{aligned}\frac{\partial \mathrm{CLL}}{\partial w_{k,i,j}} &= \frac{\partial \mathrm{CLL}}{\partial \beta_{k,i,j}} \frac{\partial \beta_{k,i,j}}{\partial w_{k,i,j}} \\ &= \sum_{\langle \mathbf{x},y \rangle \in \mathcal{D}} \left(\mathbf{1}_{y=k} - \hat{\mathrm{P}}(k|\mathbf{x}) \right) \mathbf{1}_{x_i=j} \log \theta_{x_i|i,y}. (5)\end{aligned}$$

Note that this reparameterization will have the effect of scaling each element of the gradient vector by the log of the conditional probability of that attribute given the class. An analysis of WANBIA-C's Hessian will confirm that like LR, it too leads to a convex optimization problem. Now, Equation 4 can be decomposed as:

$$\begin{aligned}& \log \hat{\mathrm{P}}(y|\mathbf{x}) = w_y \log \pi_y + \sum_i w_{y,i,x_i} \log \theta_{x_i|i,y} - \\ & \log(\sum_{y'} \exp(w_{y'} \log \pi_{y'} + \sum_i w_{y',i,x_i} \log \theta_{x_i|i,y'})).\end{aligned}$$

where π_y is the MAP estimates of $\hat{\mathrm{P}}(y)$. This leads to following estimate of posterior probability $\hat{\mathrm{P}}(y|\mathbf{x})$ as:

$$\hat{\mathrm{P}}(y|\mathbf{x}; \pi, \Theta, \mathbf{w}) \propto \pi_y^{w_y} \prod_i \theta_{x_i|i,y}^{w_{y,i,x_i}}, \quad (6)$$

where Θ is a matrix constituting $\theta_{x_i|i,y}$ and π is a vector constituting π_y.

It can be seen that our reparameterization results in a weighted naive Bayes classifier. Weighting in naive Bayes has been investigated in some detail and leads to several options. Let us modify Equation 6 as $\hat{\mathrm{P}}(y|\mathbf{x}; \pi, \Theta, \mathbf{w}) \propto \pi_y^{w_y} \prod_i \theta_{x_i|i,y}^{w(y,i,x_i)}$. Now, one can place the same weight on all attributes: $w(x_i, i, y) = w$ [1], learn a different weight for each attribute: $w(x_i, i, y) = w_i$ [2], or learn a weight for each attribute value: $w(x_i, i, y) = w_{i,x_i}$ [3]. It has been argued that these weighting schemes improve the calibration of the naive Bayes probability estimates and provide a mechanism to control the bias-variance trade-off of the NB classifier. NB is a high bias, low variance classifier and the introduction of weight parameters results in reducing its bias at the expense of increasing variance. It can be seen that our parameterization of LR results in a novel NB weighting scheme that introduces a weight per-attribute-value-per-class-value. We argue that this results in an even lower-bias variant of NB classifier which is desirable for big data. Inspired from [2], we name this parameterization WANBIA-C for *Weighting to Alleviate the Naive Bayes Independence Assumption on a per Class basis*. A subtle distinction between WANBIA-C reparameterization of LR and previous weighted NB formulations is that the parameters are unconstrained – whereas most weighted NB methods enforce a constraint $0 < \mathbf{w} < 1$.

An alternative interpretation of Equation 6 is a two stage classification. The first stage involves learning generative-parameters (π, Θ) and the second stage is the learning of discriminative parameters ($\mathbf{w}$).

III. Related Work

Pre-conditioning is a preprocessing step used in many optimization methods, which aims to accelerate convergence to the optimum by reducing the condition number of the problem. It involves a linear transformation of the variables that can correct for variables that have wildly different scales

[1] Assuming symmetric Dirichlet priors

$$\hat{\theta}_{x_i|i,y} = \frac{N_{x_i,y} + \frac{m}{|\mathcal{X}_i|}}{(N_y - N_{i,?}) + m},$$

where $N_{x_i y}$ is the count of data points with attribute value x_i and class label y, N_y is the count just for y, $N_{i,?}$ is the number of missing values of attribute i and $|\mathcal{X}_i|$ is the cardinality of the attribute.

or functions that vary differently in different directions. A number of pre-conditioners have been proposed for numeric data but few are applicable to categorical data. For example, the scale-based pre-conditioner requires specification of scale of each variable. Hessian-based pre-conditioners, on the other hand, require calculating the Hessian matrix or specifying the diagonal elements of the Hessian matrix for the optimization routine. Because we are considering only categorical datasets, we concentrate on (diagonal) Hessian-based pre-conditioners in this study.

There is a significant body of work that investigates combining generative and discriminative models. Generally, a function given by convex combination of the two models is maximized. For example: $\alpha \log \hat{P}_{\text{Disc}}(y|\mathbf{x}) + (1-\alpha)\log \hat{P}_{\text{Gen}}(y, \mathbf{x})$. However, this equation does not maximizes any well-defined model. WANBIA-C, on the hand, maximizes CLL over the data and hence achieves similar goals of combining the two models in a very elegant manner. We will show that regularization in WANBIA-C gives a meta-parameter λ that can be used to control the generative-discriminative component of the classifier. On one extreme, one obtains naive Bayes and on the other hand un-regularized LR.

There has been a significant amount of research into weighting attributes for naive Bayes [3], [4]. Most of these works were primarily motivated from the point of view of increasing the influence of those attributes that are highly correlated with the class.

IV. Experiments - Batch Optimization

In this section, we compare the performance of two parameterizations of LR in terms of training time. For the sake of completeness, we also show a comparison between the two parameterizations (expecting to see near identical performance) in terms of 0-1 loss and root mean square error (RMSE) i.e., $\frac{1}{2}\sum_{\mathbf{x}\in\mathcal{D}}\sum_{y}(\mathrm{P}(y|\mathbf{x})-\hat{\mathrm{P}}(y|\mathbf{x}))^2$, bias and variance on 73 natural domains from the UCI repository (Table I). Quantitative attributes are discretized by using Minimum Description Length (MDL) discretization. Each algorithm is tested on each dataset using 20 rounds of 2-fold cross validation. Optimization settings are exactly the same when comparing the two parameterization.

A. Quasi-Newton Method

For L-BFGS quasi-Newton methods [5], the matrix B is an approximation to the Hessian that is updated at every iteration by means of a low-rank formula [2] [3]. QN are extremely popular for unconstrained optimization and are widely used for optimizing LR. A comparative study of various optimization routines for LR are done in [7], where quasi-Newton performs better than most alternative routines.

	WANBIA-C vs. LR		**WANBIA-C vs. NB**	
	W-D-L	p	W-D-L	p
Bias	29/24/20	0.252	62/3/8	**<0.001**
Variance	28/23/22	0.479	17/3/53	**<0.001**
		All Datasets (73)		
0-1 Loss	32/18/23	0.280	45/4/24	**0.015**
RMSE	29/19/25	0.683	41/3/29	0.188
		Top Size Datasets (12)		
0-1 Loss	2/9/1	1.000	12/0/0	**<0.001**
RMSE	2/7/3	1.000	12/0/0	**<0.001**

Table II
Win-Draw-Loss: WANBIA-C vs LR and NB. p is two-tail binomial sign test. Results are significant if $p \leq 0.05$.

WANBIA-C is compared with LR and NB in terms of W-D-L on 73 datasets in Table II. The results are shown separately for 12 biggest datasets. Comparing with NB, it can be seen that WANBIA-C significantly improves NB's Bias and 0-1 loss. It also improves RMSE, though non-significantly. As expected, variance is significantly worst than NB. On bigger datasets, WANBIA-C wins on all 12 in terms of 0-1 Loss and RMSE. Comparing with LR, WANBIA-C has similar bias-variance profile. The 0-1 loss and RMSE results are not significantly different even on larger datasets.

The scatter plots of WANBIA-C and LR showing RMSE, training time and number of iterations taken by each algorithm to converge is shown in Figure 1. These results are after convergence of each of the two parameterizations. One can see that the RMSE profile of the two algorithms is the same but WANBIA-C is greatly advantaged due to faster training time resulting from fewer iterations. The convergence analysis on sample datasets is shown in Figure 2 where variation in conditional log likelihood is plotted against QN iterations. One can see that WANBIA-C has steeper ascent as compared to LR.

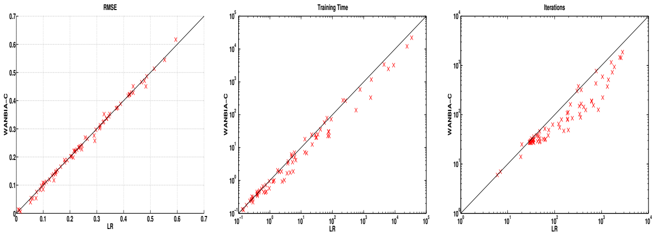

Figure 1. Comparison of RMSE (Left), training time (Middle) and number of iterations (Right) of WANBIA-C and LR on 73 datasets using QN. Training time and number of iterations are on log-scale.

Until now, parameters of LR (β) are initialized to zero. A common approach is to set β to a small random set of values. Yet, another approach is to begin with the MAP

[2] The algorithm terminates when relative improvement in the objective function, given by $\frac{(f_t - f_{t+1})}{\max\{|f_t|,|f_{t+1}|,1\}}$, drops below 10^{-32}, or the number of iterations exceeds 10000.

[3] The original L-BFGS implementation of [6] from http://users.eecs.northwestern.edu/~nocedal/lbfgsb.html is used.

Domain	Case	Att	Class	Domain	Case	Att	Class	Domain	Case	Att	Class	Domain	Case	Att	Class	Domain	Case	Att	Class
Activity	3850500	45	19	PenDigits	10992	17	10	Splice-junctionGeneSequences	3190	62	3	Syncon	600	61	6	SonarClassification	208	61	2
USCensus1990	2458300	67	4	Thyroid	9169	30	20	Segment	2310	20	7	Chess	551	40	2	AutoImports	205	26	7
Poker-hand	1175067	11	10	Pioneer	9150	37	57	CarEvaluation	1728	8	4	Cylinder	540	40	2	WineRecognition	178	14	3
MITFaceSetC	839300	361	2	Mushrooms	8124	23	2	Volcanoes	1520	4	4	Musk1	476	167	2	Hepatitis	155	20	2
Covertype	581012	55	7	Musk2	6598	167	2	Yeast	1484	9	10	HouseVotes84	435	17	2	TeachingAssistantEvaluation	151	6	3
MSDYearPrediction	515300	90	90	Satellite	6435	37	6	ContraceptiveMethodChoice	1473	10	3	HorseColic	368	22	2	IrisClassification	150	5	3
MITFaceSetB	489400	361	2	OpticalDigits	5620	49	10	German	1000	21	2	Dermatology	366	35	6	Lymphography	148	19	4
Census-Income(KDD)	299285	40	2	PageBlocksClassification	5473	11	5	LED	1000	8	10	Ionosphere	351	35	2	Echocardiogram	131	7	2
Localization	164860	7	3	Wall-following	5456	25	4	Vowel	990	14	11	LiverDisorders(Bupa)	345	7	2	PromoterGeneSequences	106	58	2
Connect-4Opening	67557	43	3	Nettalk(Phoneme)	5438	8	52	Tic-Tac-ToeEndgame	958	10	2	PrimaryTumor	339	18	22	Zoo	101	17	7
Statlog(Shuttle)	58000	10	7	Waveform-5000	5000	41	3	Annealing	898	39	6	Haberman'sSurvival	306	4	2	PostoperativePatient	90	9	3
Adult	48842	15	2	Spambase	4601	58	2	Vehicle	846	19	4	HeartDisease(Cleveland)	303	14	2	LaborNegotiations	57	17	2
LetterRecognition	20000	17	26	Abalone	4177	9	3	PimaIndiansDiabetes	768	9	2	Hungarian	294	14	2	LungCancer	32	57	3
MAGICGammaTelescope	19020	11	2	Hypothyroid(Garavan)	3772	30	4	BreastCancer(Wisconsin)	699	10	2	Audiology	226	70	24	Contact-lenses	24	5	3
Nursery	12960	9	5	Sick-euthyroid	3772	30	2	CreditScreening	690	16	2	New-Thyroid	215	6	3				
Sign	12546	9	3	King-rook-vs-king-pawn	3196	37	2	BalanceScale	625	5	3	GlassIdentification	214	10	3				

Table I
DATA SETS

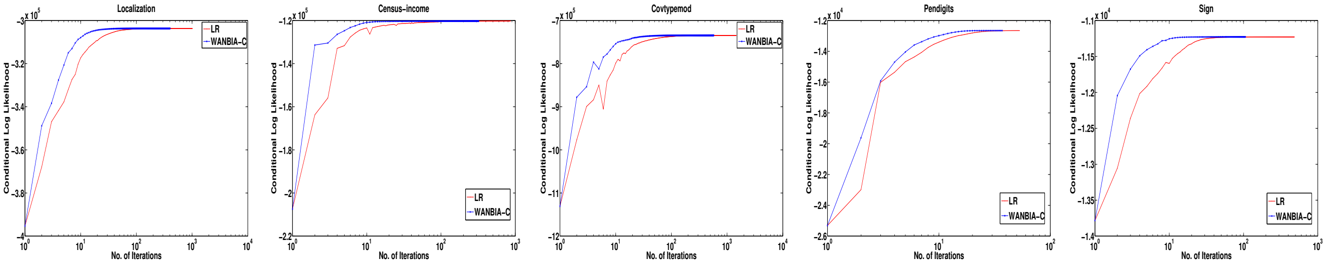

Figure 2. Objective function's convergence comparison of WANBIA-C and LR on `Localization`, `Census-income`, `Covtypemod`, `Pendigits`, `Sign` datasets using QN optimization. Number of iterations are on log-scale.

(or MLE) probability estimates computed through generative process. WANBIA-C is compared with LR where parameters are initialized to be naive Bayes' probability estimates in Figure 3. One can see that even though starting LR with NB estimates results in some improvements, WANBIA-C still converges in far fewer iterations.

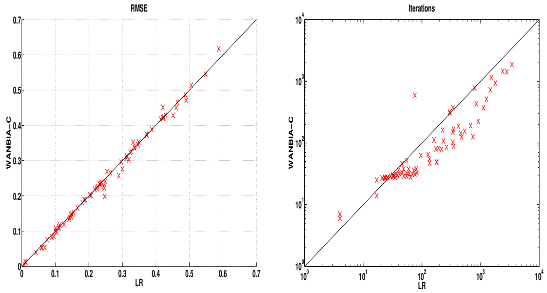

Figure 3. Comparison of RMSE (Left) and number of iterations (Right) of WANBIA-C and LR on 73 datasets using quasi-Newton optimization. LR parameters are initialized to NB MAP estimates. Number of iterations are on log-scale.

B. Pre-conditioners for Optimization

The goal of experiments in this section is to compare WANBIA-C with existing pre-conditioners that are used with L-BFGS. We use the ALGLIB library implementation of L-BFGS [8], which allows for diagonal scale-based and diagonal Hessian-based pre-conditioners. Our preliminary experiments suggested that simple diagonal Hessian-based pre-conditioning was ineffective, and so an adaptive pre-conditioning strategy was investigated whereby the diagonal elements of the Hessian were recomputed on each iteration. Results on 10 datasets comparing WANBIA-C, LR and diagonally pre-conditioned LR (LR+diag-precond) are shown in Figure 4. One can see that the classification performance of three algorithms is exactly the same with the exception of one dataset: `wall-following`. Training time for both LR alternatives is worse than WANBIA-C, with the adaptive pre-conditioning strategy appears taking even longer than vanilla LR. And while pre-conditioning helped reduce the number of iterations to convergence for LR on some datasets, they are still greater than those of WANBIA-C.

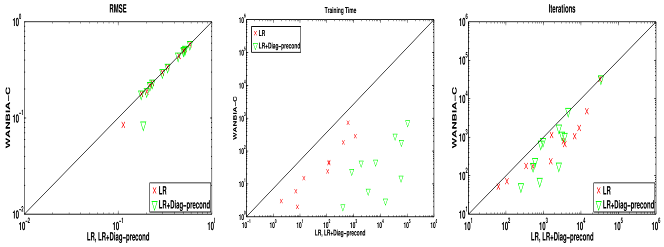

Figure 4. Comparison of RMSE (Left), Training time (Middle) and Number of iterations (Right) of WANBIA-C with LR and LR+Diagonal-Pre-conditioned on 10 datasets using quasi-Newton optimization. Training time and number of iterations are plotted on log-scale.

V. Experiments - Online Optimization

Stochastic Gradient Descent (SGD) methods update the parameters after analyzing a data point. That is, the gradient vector $\mathbf{g}$ is calculated only on a single data point which is then used (scaled by the step-size) to update the parameters. This is in contrast to batch optimization where the parameters are updated after calculating $\mathbf{g}$ over the whole dataset. SGD learners are highly sort after due to the emergence of big data. We use a learning rate of $\eta = \frac{\eta_0}{1+\lambda t}$, where λ is the regularization parameter set to 10^{-2} and the constant η_0 is determined through cross-validation, by searching the best value from the set: $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{-0}, 10^{1}, 10^{2}\}$.

The performance of the two parameterizations is compared by plotting the learning curves in terms of RMSE with time step t. Following is the procedure for generating the learning curves. Randomize the order of instances in a dataset. Pick the first data point and use it to learn a model. Next, pick the second data point and classify it using the learned model. Compute the RMSE and record the value. Next, update the model using the second data point. The same procedure is applied to subsequent data points and continues until the end of records. In our experiments, we ran this procedure twice with different randomization of datasets. To plot the curves, we took the average of the values across two runs and plotted the results by a moving average filter of size 10000. A comparison of RMSE learning curves across various datasets is shown in Figure 5. It can be seen that WANBIA-C has better learning curves, i.e, starts from a lower value and asymptotes to a smaller value than LR. Note that in these experiments, WANBIA-C is advantaged as it avails two passes through the data. In the first pass, it computes the probabilities and in the second pass, it learns the weights – whereas, LR relies on single pass through the data. However, it is extraordinary to see how prior information in the form of naive Bayes estimates of probabilities can result in drastic improvement in the classification performance of a single-pass learner. Single pass WANBIA-C has been left as a future work.

VI. Interpolation Framework

The objective function for L_2 regularized LR takes the form: $\sum_{\langle \mathbf{x},y\rangle \in \mathcal{D}} \log \hat{\mathrm{P}}(y|\mathbf{x}) + C\|\beta\|^2$. For WANBIA-C, one can introduce an identical regularization term as follows: $\sum_{\langle \mathbf{x},y\rangle \in \mathcal{D}} \log \hat{\mathrm{P}}(y|\mathbf{x}) + C\|\mathbf{w}\|^2$. The new term will penalize large (and heterogeneous) parameter values, such that larger C values will cause the classifier to progressively ignore the data and assign more uniform class probabilities. Alternatively one could penalize deviations from the NB conditional independence assumption by centering the regularization term at one rather than zero: $\sum_{\langle \mathbf{x},y\rangle \in \mathcal{D}} \log \hat{\mathrm{P}}(y|\mathbf{x}) + C\|\mathbf{w} - \mathbf{1}\|^2$. Doing so allows the regularization parameter C to be used to interpolate between the generative NB model and the discriminative model by controlling the amount of deviation from the conditional independence assumption.

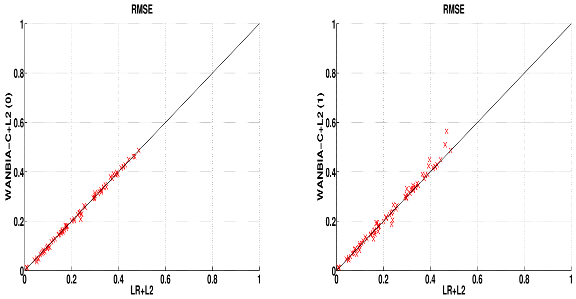

Figure 6. Performance comparison for the two versions of regularized WANBIA-C versus regularized LR in terms of RMSE. (Left) WANBIA-C regularized towards zero. (Right) WANBIA-C regularized towards one.

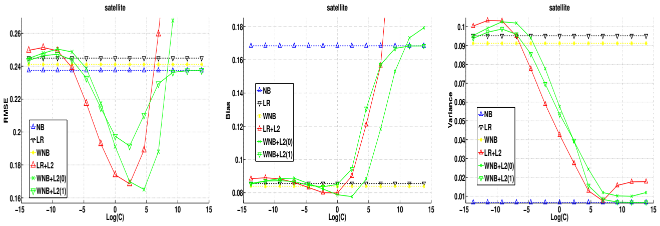

Figure 7. Variation in RMSE, bias and variance of LR+L2, WANBIA-C+L2(0) and WANBIA-C+L2(1) with regularization parameter C. Horizontal lines represent the RMSE, bias and variance of NB, unregularized LR and unregularized WANBIA-C on sample dataset (`Satellite`).

We first compare regularized LR with regularized WANBIA-C in Figure 6. Results were computed by setting the regularization parameter $C = 10^x$ for $x \in \{-6, -4, -2, 0, 2, 4, 6\}$, and then selecting the best performing value for C in terms of RMSE. We see that WANBIA-C+L2(0), with parameters regularized toward zero, has a near identical performance profile to that of regularized LR. The alternative model WANBIA-C+L2(1), where the parameters are drawn towards one, has more varied performance with respect to regularized LR.

In order to better understand the effects regularization has on classifier performance it is instructive to graph estimates of bias and variance as a function of the level of regularization. Figure 7 plots RMSE as well as bias and variance against the regularization parameter C for the `Satellite` dataset. The plots also show un-regularized NB, LR and WANBIA-C values (as horizontal lines) indicating that LR and WANBIA-C have low bias but high variance with respect to NB on this dataset. As mentioned previously, the WANBIA-C+L2(1) classifier allows us to interpolate between WANBIA-C estimates (for small values of C) and NB estimates (for large values). This mechanism for blending the two algorithms allows us to directly trade-off the bias and variance for the classifier. The best value for C in this case (10^2) results in much better RMSE

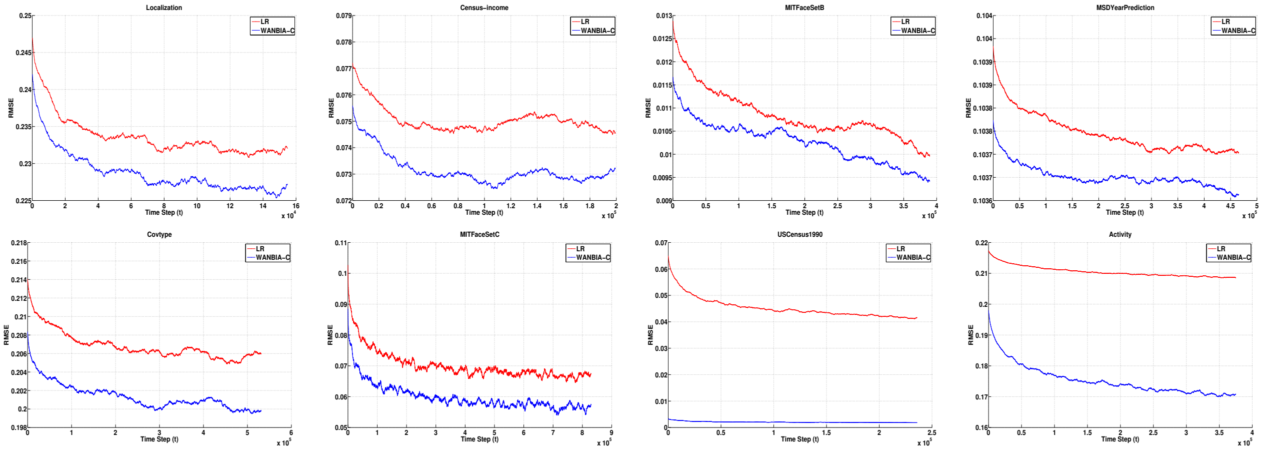

Figure 5. RMSE comparison of LR and WANBIA-C with SGD on Localization, Census-income, MITFaceSetB, MSDYearPrediction, Covtype, MITFaceSetC, USCensus1990, and Activity datasets.

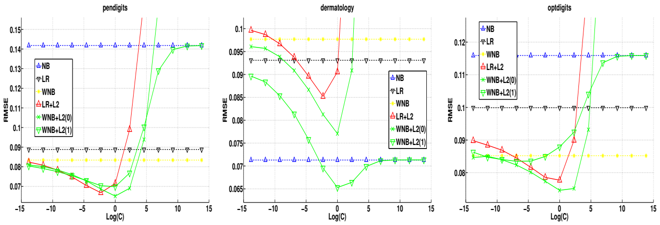

Figure 8. Variation in RMSE of LR+L2, WANBIA-C+L2(0) and WANBIA-C+L2(1) with regularization parameter C on representative datasets (Pendigits, Dermatology and Optdigits. Horizontal lines represent the RMSE of NB, unregularized LR and unregularized WANBIA-C.

performance than either of the constituent classifiers (NB and WANBIA-C) as shown in the left-most sub-figure in Figure 7. In Figure 8 we plot RMSE performance versus the amount of regularization on sample representative datasets. We see that in some cases regularization results in significant performance improvements over the un-regularized techniques, while in other cases the performance improvements are negligible at best.

VII. Conclusion and Future Work

In this work, we proposed an effective alternative parameterization of logistic regression – WANBIA-C. Parameters in LR are scaled by NB probability estimates in pursuit of better convergence. Our reparameterization resulted in a weighted naive Bayes model. We showed that WANBIA-C converges much faster than LR when using quasi-Newton and most importantly stochastic gradient descent optimization. We also showed that regularizing the parameters of WANBIA-C provides an elegant way to mix generatively and discriminatively learned parameters. This also leads to a framework for controlling the bias-variance of the classifier. We showed that the information provided by the initial MAP estimates in WANBIA-C substantially benefits the gradient-based optimization methods employed for discriminative training of weights. In future, we seek to develop a deeper theoretical understanding of the reasons why the initial MAP estimates so substantially improve the discriminative search.

VIII. Acknowledgments

This research has been supported by the Australian Research Council (ARC) under grant DP140100087 and Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research under contract FA23861214030.

References

[1] J. Hilden and B. Bjerregaard, "Computer-aided diagnosis and the atypical case," in *In Decision Making and Medical Care: Can Information Science Help*. North-Holland Publishing Company, 1976, pp. 365–378.

[2] N. A. Zaidi, J. Cerquides, M. J. Carman, and G. I. Webb, "Alleviating naive Bayes attribute independence assumption by attribute weighting," *Journal of Machine Learning Research*, vol. 14, pp. 1947–1988, 2013.

[3] J. T. A. S. Ferreira, D. G. T. Denison, and D. J. Hand, "Weighted naive Bayes modelling for data mining," 2001.

[4] M. A. Hall, "A decision tree-based attribute weighting filter for naive Bayes," *Knowledge-Based Systems*, vol. 20, pp. 120–126, March 2007.

[5] C. Zhu, R. H. Byrd, and J. Nocedal, "L-bfgs-b: Algorithm 778: L-bfgs-b, fortran routines for large scale bound constrained optimization," *ACM Transactions on Mathematical Software*, vol. 23, no. 4, pp. 550–560, 1997.

[6] R. Byrd, P. Lu, and J. Nocedal, "A limited memory algorithm for bound constrained optimization," *SIAM Journal on Scientific and Statistical Computing*, vol. 16, no. 5, pp. 1190–1208, 1995.

[7] T. P. Minka, "A comparison of numerical optimizers for logistic regression," 2003.

[8] S. Bochkanov and V. Bystritsky. [Online]. Available: http://www.alglib.net