Learning with Mean-Variance Filtering, SVM and Gradient-based Optimization [⋆]

Vladimir Nikulin

Mathematical Sciences Institute, Australian National University, Canberra

Summary

◆ 1) Gradient-based method as a core optimization tool with such examples as quadratic minimization and logit model (base models).

◆ 2) Distance-based clustering and non-linear classifier (ENV-2006, PAKDD-2006).

◆ 3) An ensemble system as a sequence of several different base models (ADA).

◆ 4) Effectiveness of the linear and non-linear SVM (GINA and NOVA).

◆ 5) One of the base models plus several association rules (SYLVA).

◆ 6) Feature selection using mean-variance filtering model (HIVA).

◆ 7) Concluding remarks and further developments.

[⋆] This work was supported by a grant of the Australian Research Council.

The formulation of the problem

Let $\mathbf{X} = (\mathbf{x}_t, y_t), t = 1..n$, be a training sample of observations where $\mathbf{x}_t$ is ℓ-dimensional vector of features, and y_t is binary label: $y_t \in \{-1, 1\}$.

In practical situation the label y_t may be hidden, and the task is to estimate it using vector of features. Let us consider the most simple linear decision function

$$u_t = \sum_{j=1}^{\ell} w_j \cdot x_{tj} + b \tag{1}$$

where w_i are weight coefficients and b is a bias term.

We will make decision $\widehat{y}_t = 1$ if $u_t \geq \lambda$, alternatively, we will conclude that $\widehat{y}_t = -1$ where λ is a threshold regulation parameter.

The optimization criterion is to minimize the balanced error rate (*BER*):

$$Q(\lambda) = \frac{1}{2}\left(\frac{q_{12}}{q_{11} + q_{12}} + \frac{q_{21}}{q_{21} + q_{22}}\right) \tag{2}$$

where value of q_{ij} equal to the number of j-predictions in the true cases of $i = 1..2$. Unfortunately, the target function (2) can not be optimized directly. Respectively, we will consider several alternative (differentiable) target functions assuming that the corresponding models will produce good solutions in the sense of (2).

Quadratic Minimization (*QM*) model

Let us consider the most basic quadratic minimization model with the following target function:

$$L(\mathbf{w}) = \sum_{t=1}^{n} (y_t - u_t)^2 . \tag{3}$$

The direction of the steepest decent is defined by the gradient vector

$$g(\mathbf{w}) = \{g_j(\mathbf{w}), j = 1..\ell\},$$

where

$$g_j(\mathbf{w}) = \frac{\partial L(\mathbf{w})}{\partial w_j} = -2 \sum_{t=1}^{n} x_{tj} (y_t - u_t) .$$

Initial values of the linear coefficients w_i and bias parameter b may be arbitrary. Then, we recompute the coefficients

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} + \varDelta_k \cdot g(\mathbf{w}^{(k)}), \tag{4a}$$

$$b^{(k+1)} = b^{(k)} + \frac{1}{n} \sum_{t=1}^{n} (y_t - u_t) \tag{4b}$$

where k is a sequential number of iteration. Minimizing (3) we find size of the step according to the formula

$$\varDelta = \frac{L_1 - L_2}{\sum_{t=1}^{n} s_t^2} \tag{5}$$

where

$$L_1 = \sum_{t=1}^{n} s_t y_t, \quad L_2 = \sum_{t=1}^{n} s_t u_t, \quad s_t = \sum_{j=1}^{\ell} x_{tj} g_j .$$

Selection of the threshold parameter

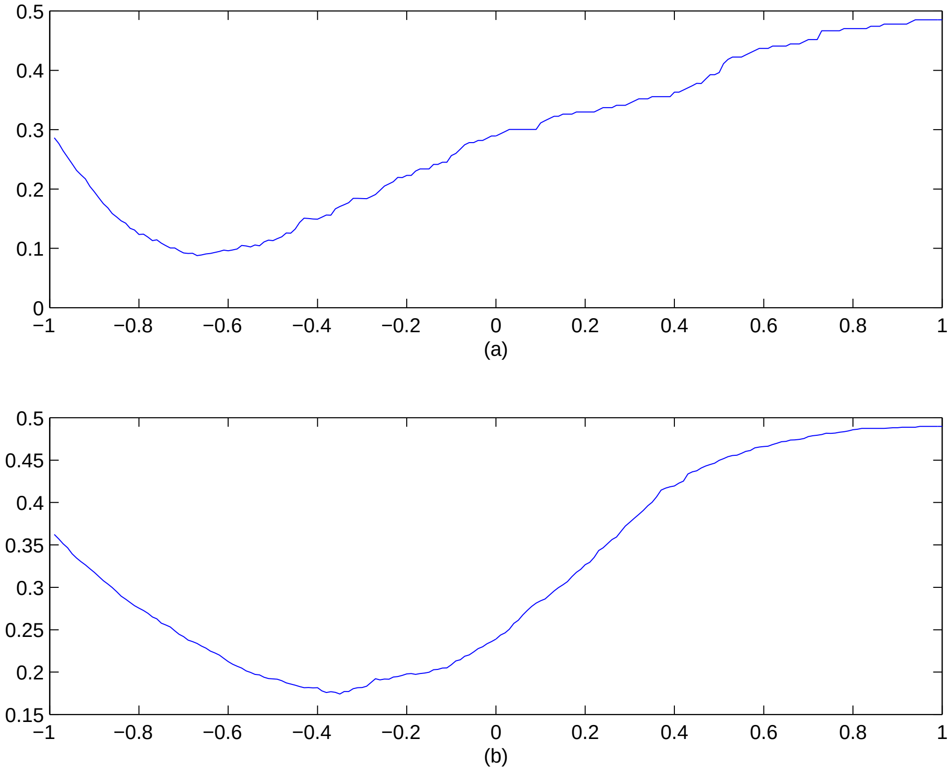


Fig. 1. Behavior of the *BER* (2) as a function of the parameter λ (a): *HIVA*, (b): *ADA*.

Logit Model

Let us consider a non-linear modification of the target function

$$L(\mathbf{w}) = \sum_{t=1}^{n} (y_t - f(u_t))^2 \qquad (6)$$

where $f(t) = \frac{1-e^{-t}}{1+e^{-t}}$.

Newton's Method

The objective of the Newton's method is to find a solution for the equation

$$g(\mathbf{w}) = 0,$$

which represents necessary condition of an optimum.

The following equation may be used as a base for the iterative algorithm and is called Newton's step

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - \nabla^2 L(\mathbf{w}^{(k)})^{-1} g(\mathbf{w}^{(k)})$$

where $\nabla^2 L$ is a matrix of second derivatives of the target function.

As a target function we used log-likelihood function

$$L(\mathbf{w}) = \sum_{t=1}^{n} \left[z_t \log\{P_t\} + (1 - z_t) \log\{1 - P_t\}\right] \qquad (7)$$

where $z_t = 0.5 \cdot (1 + y_t)$,

$$P_t = \frac{e^{u_t}}{1 + e^{u_t}}$$

where $u_t = \sum_{j=0}^{\ell} w_j \cdot x_{tj}$, and in order to simplify notations a constant is regarded as one of the features.

Distance-based clustering and non-linear classifier

Using k-means algorithm we can split the whole training dataset into several subsets/clusters where any cluster is represented by centroid. Then, we can compute vector of coefficients specifically for any particular cluster.

The complex of two matrices 1) centroids and 2) coefficients may be used as a nonlinear classifier, which works as follows:

♦ 1) for any data-instance we find the nearest centroid;
♦ 2) compute decision function according to the corresponding vector of regression coefficients.

Above method has been proved to be very effective in application to the KDD-99 intrusion detection database. Also, it produced good results for the PAKDD-2006 and for the Environmental Modelling Competition.

An Ensemble System

The solutions as an outcome of linear and logit models are far from identical. Respectively, these models may be used together. For example, assuming that linear model is leading, we can make decision if

$$|u_t - \lambda| \geq \delta > 0 : \widehat{y}_t = 1 \text{ if } u_t \geq \lambda + \delta, \widehat{y}_t = -1 \text{ if } u_t \leq \lambda - \delta.$$

Suppose that $|u_t - \lambda| < \delta$. In this case, we can employ, for example, logit model, and will make decision if $|f(\tilde{u}_t) - \lambda_1| \geq \delta_1 > 0$ (using similar technique as above).

The case $|u_t - \lambda| \leq \delta$ and $|f(\tilde{u}_t) - \lambda_1| \leq \delta_1$ may be regarded as a disputable and will require some additional investigation, which may be conducted using third model or we can return to the first model, for example.

The threshold parameters λ and λ_1 may be selected as a result of the separate or combined optimizations for linear and logit models against the training set. The parameters δ and δ_1 represent levels of confidence in relation to the linear and logit models.

Support Vector Machines: (a): behavior of the target function in the case of *GINA*-set; (b): coefficients α were sorted in a decreasing order

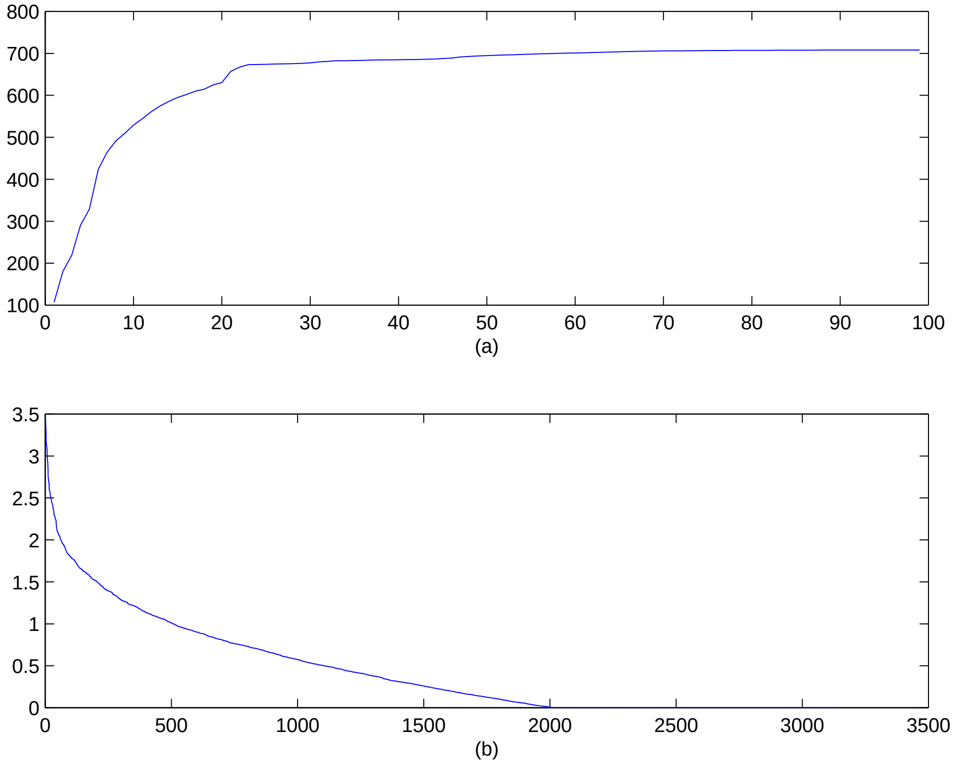


Fig. 2.

Behavior of the of the *BER* as a function of the parameter λ (a): *GINA*, (b): *NOVA*

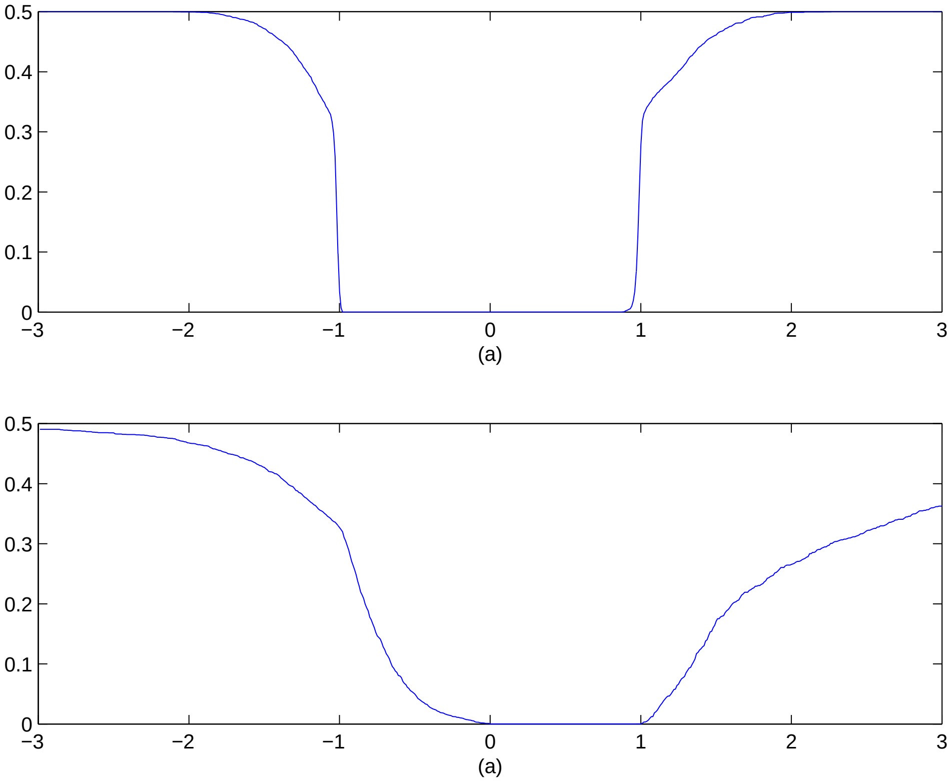


Fig. 3.

Onelevel DTrees

OneLevel pattern:

$$A_j := \{t : x_{tj} \geq 1\}, \qquad n_j = \# A_j;$$

Outcome:

$$B_j := \{t \in A_j : \mathbb{C}\}, \qquad m_j = \# B_j;$$

$$\mathbb{C} = \{y_t = -1\}.$$

Criterions:

$$1) n_j \geq \alpha \ \text{(confidence)}; \qquad 2) \frac{m_j}{n_j} \geq \beta \ \text{(support)}$$

where $\alpha = 200, \beta = 0.999$ (28 patterns in the case of *SYLVA*).

SecondLevel DT

$$A_{ij} := \{t : x_{ti} \geq 1, x_{tj} \geq 1\}, \qquad n_{ij} = \# A_{ij};$$

Outcome:

$$B_{ij} := \{t \in A_{ij} : \mathbb{C}\}, \qquad m_{ij} = \# B_{ij}.$$

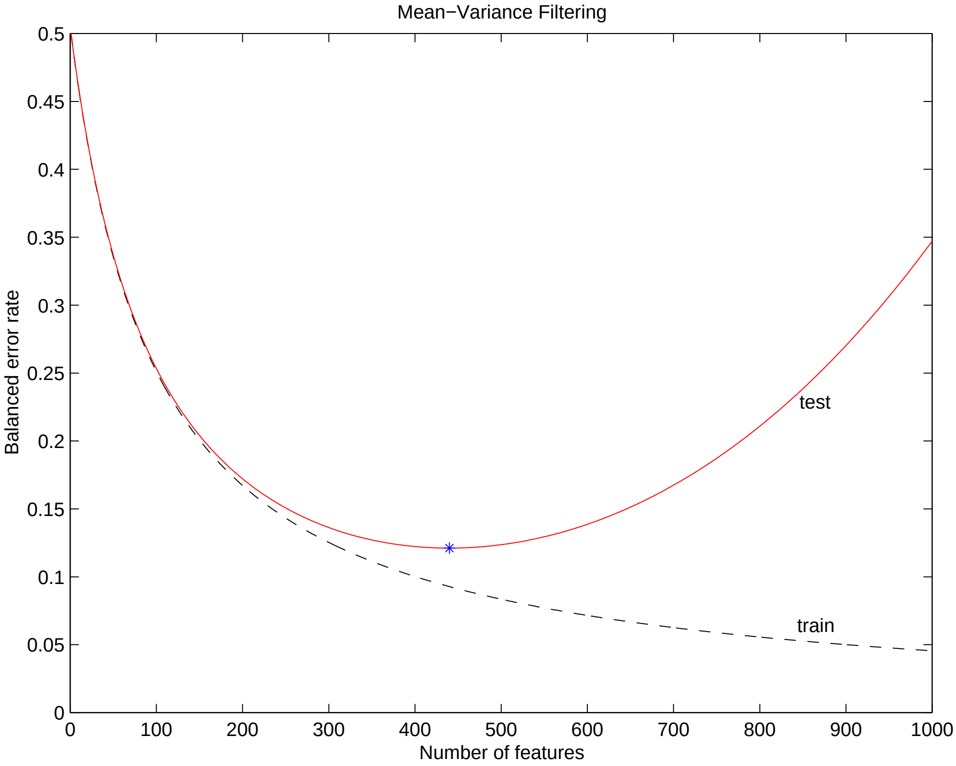
Criterions:

$$1) n_{ij} \geq \alpha \ \text{(confidence)}; \qquad 2) \frac{m_{ij}}{n_{ij}} \geq \beta \ \text{(support)}.$$

Continue until $A_I = \emptyset$.

Mean-Variance: main idea

Using one of the base models we can compute vector of coefficients $\mathbf{w}_0$ for the whole training set with an excellent simulation result. An application of $\mathbf{w}_0$ to another set may produce inconsistently poor results. In this situation it will be good to investigate stability of the particular coefficients as a components of the vector $\mathbf{w}_0$. In other words, it is very important to clarify consistency of the influence of different features depending on the different parts of training set.



Mean-Variance model

Using as a base the whole training set $\mathbf{X}$ we can form several subsets $\mathbf{X}_j$ with approximately equal size: $\mathbf{X} = \cup_{j=1}^{m} \mathbf{X}_j$. Note, that subsets $\mathbf{X}_j$ must be sufficiently large. Respectively, they may have not empty intersections.

$$\mathbb{C} = \{y_t = -1\}.$$

As a next step we compute matrix of coefficients where any row corresponds to the particular subset $\mathbf{X}_j$.

We removed features according to the condition

$$r_i = \frac{M_i - m_i}{\min\limits_{t \in [m_i, M_i]} |t|} \geq H, i = 1..\ell, \tag{8}$$

where $H > 0$ is a threshold parameter, m_i and M_i are minimal and maximal values of the coefficient w_i.

Remark 1. Experiments with

$$r_i = \frac{s_i}{\mu_i} \tag{9}$$

produced slightly worse results where μ_i is a sample mean and s_i is a standard deviation of the coefficient with index i.

Cross-Validation

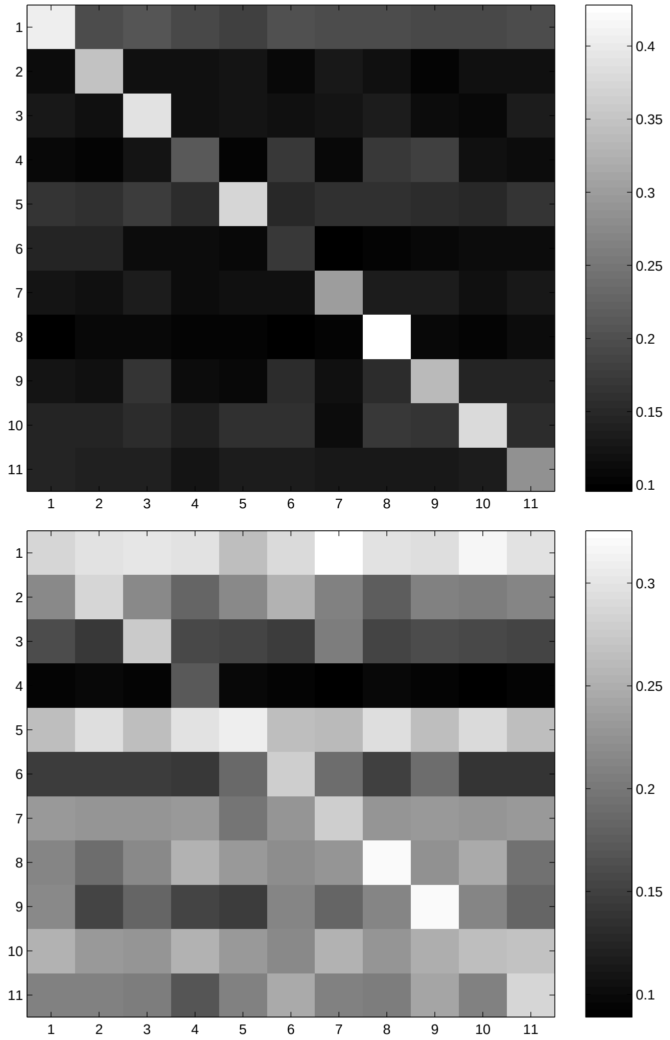


Fig. 5. Matrix of *BERs* before and after *MVF*.

Concluding remarks

Table 1. Best test entries

Dataset	*AUC*	*BER*	Guess *BER*	Guess error
ADA	0.8225	0.1851	0.1650	0.0201
GINA	0.9348	0.0566	0.0600	0.0034
HIVA	0.6605	0.3515	0.2500	0.1015
NOVA	0.9474	0.0507	0.0500	0.0007
SYLVA	0.9913	0.0122	0.0100	0.0022
Overall	0.8713	0.1312	0.1070	0.0261

Further developments

⧫ 1) ICA & MVF.

⧫ 2) Smoothed MVF.

⧫ 3) MultiLevel DTrees and RF.