Feature Import Vector Machine (FIVM): A General Classifier with Flexible Feature Selection

Dr. Samiran Ghosh
sghos@med.wayne.edu

This is a joint work with Y. Wang
University of Wisconsin, Madison

This work is partly supported by: NIH P30-ES020957, R01-NS079429
What is Classification Problem:

“Suppose, we have a clinical study entailing genetic and other clinical profiles of 100 \((n)\) subjects which can be either classified as Bipolar or Unipolar Disorder. Our task is to identify a subset of these profiles as a marker to these Disease”

- This is a supervised learning problem, with the outcomes as the class variable (disease type). It is also called a classification problem
- In case the true disease type is not known, this becomes unsupervised learning problem or clustering
- Number of disease types not necessarily dichotomous \((p)\)
Classification in General

- Classification is a supervised learning problem
- Preliminary task is to construct classification rule (some functional form) from the training data
- For $p<<n$, many methods are available in classical statistics,
  - Linear (LDA, LR)
  - Non-Linear (QDA, KLR)
- However when $n<<p$, we face estimability problem
- Some kind of data compression/transformation is inevitable.
- Well known techniques for $n<<p$, PCR, SVM etc.
Classification in High Dimension \( (n<<p) \)

- We will concentrate on \( n<<p \), domain.
- Application domains: Many, but primarily Bioinformatics

**Few points to note,**

- Support Vector Machine is a very successful non-parametric technique based on RKHS principle
- Our proposed method is based on RKHS principle
- In High dimension it is often believed that all dimensions are not carrying *useful* information
- In short our methodology will employ dimension filtering based on the RKHS principle
Suppose our training data set, \( D = \{x_i, y_i\}_{i=1}^n \), \( x_i \in \mathbb{R}^p \), \( y_i \in \{-1,1\} \)

A general class of regularization problem is given by

\[
\min_{f \in \mathcal{H}} \left[ \sum_{i=1}^n L(y_i, f(x_i)) + \lambda J(f) \right]
\]

Where \( \lambda > 0 \) is a regularization parameter and \( \mathcal{H} \) is a space of function in which \( J(f) \) is defined.

By the Representer theorem of Kimeldorf and Wahba the solution to the above problem is finite dimensional,

\[
f(x) = \sum_{i=1}^n \alpha_i K(x, x_i)
\]

\( \{ K(x, x') : X \times X \to \mathbb{R} \) is a kernel function

\( J(f) = \| f \|^2_{H_k} \to \) is the second order norm
Choice of Kernel

$K(x, x')$ is a suitable symmetric, positive (semi-)definite function.

$d$th deg. poly.: $K(x, x') = (1 + \langle x, x' \rangle)^d$

radial basis: $K(x, x') = \exp(-\|x - x'\|^2 / c)$

RKHS or $\mathcal{H}$ will be the vector space spanned by $K(., x)$

Due to the inner product $K(x_i, x_j) = \langle K(., x_i), K(., x_j) \rangle$

This is also known as reproducing property of the kernel

SVM is a special case of the above RKHS setup, which aims at maximizing margin

$$\max_{\|\beta\| = 1} C \quad \text{Subject to} \quad y_i f(x_i) \geq C \text{ for } i = 1, 2, \ldots, n$$
SVM based Classification

In SVM we have a special loss and roughness penalty,

\[
\min_{f \in \mathcal{H}_k} \frac{1}{n} \sum_{i=1}^{n} [1 - y_i f(x_i)]_+ + \lambda \|f\|_{\mathcal{H}_k}^2
\]

By the Representer theorem of Kimeldorf and Wahba the optimal solution to the above problem,

\[
f(x) = \sum_{i=1}^{n} \alpha_i K(x, x_i)
\]

However for SVM most of the \(\alpha_i\) are zero, resulting huge data compression.

In short, kernel based SVM perform classification by representing the original function as a linear combination of the basis functions in the higher dimension space.
Key Features of SVM

- Achieves huge data compression as most $\alpha_i$ are zero
- However this compression is only in terms of $n$
- Hence in estimation of $f(x)$ it uses only those observations that are close to classification boundary

Few Points,

- In high dimension ($n<<p$), compression in terms of $p$ is more meaningful than that of $n$
- Standard SVM is only applicable for two class classification problem
- Results have no probabilistic interpretation as we cannot estimate $p(x)[= P(y=1|X=x)]$ rather only $\text{sign}\left(p(x) - \frac{1}{2}\right)$
To overcome drawbacks of SVM, Zhu & Hastie (2005) introduced IVM (import vector machine) based on KLR.

In IVM we replace hinge loss with $\ln[1 + e^{-yf}]$, the NLL of binomial distribution. Then we get natural estimate of classification as,

$$P(Y = 1 | X = x) = \frac{1}{1 + e^{-f(x)}}, \text{and } y \in \{-1, 1\} \text{ and } x \in \mathbb{R}$$

The advantages are crucial:

1. Exact classification probability can be computed
2. Multi-class extension of the above is straight forward
However ...

Previous advantages come at a cost,

- It destroys the sparse representation of SVM, i.e. all $\alpha_i$ are non zero, & hence no compression (neither in $n$ nor in $p$)

They employ an algorithm to filter out only few *significant* observations ($n$) which will *help* the classification, most.

- These selected observations are called Import points.
- Hence it serves both data compression ($n\downarrow$) and probabilistic classification ($p(x)$)

However for $n<<p$

It is much more meaningful if compression is in $p$. (why?)
Why Bother About $p$?

- Obviously $n<<p$ and in practical bioinformatics application, $n$ is not a quantity to be reduced much.
- *Physically* $p$’s are what? Depending upon domain they are Gene, Protein, Metabonome etc.
- If a dimension selection scheme within classification can be implemented, it will also generate possible candidate list of biomarkers.

Essentially we are talking about simultaneous variable selection and classification in high dimension.

Are there existing methods which already do that

What about $L_1$ penalty and LASSO?
Least Absolute Selection and Shrinkage Operator

LASSO is a popular $L_1$ penalized least square method proposed by Tibshirani (1997) in regression context. Lasso minimizes,

$$\arg\min_{\beta} \left\{ \left\| y - \sum_{j=1}^{p} x_j \beta_j \right\|^2 + \lambda_n \sum_{j=1}^{p} |\beta_j| \right\} \quad \text{subject to} \quad \sum_{j} |\beta_j| \leq t.$$ 

Due to the nature of the penalty and choice of $t(\geq 0)$, LASSO produces threshold rule by making many small $\beta$’s zero.

- Replacing squared error loss by NLL of binomial distribution, LASSO can do probabilistic classification.
- Roth (2004) proposed KLASSO (Kernelized)

Nonzero $\beta$’s are selected dimensions ($p$).
Disadvantage of LASSO

- LASSO does variable selection through $L_1$ penalty
- If there are high correlations between variables, LASSO tend to select only one of them.
- Owing to the nature of convex optimization problem, it can select at most $n$ out of the $p$ variables.

The last one is a severe restriction.

We are going to propose a method based on KLR and IVM which does not suffer from this drawback.

We will essentially change the role of $n$ and $p$ in IVM problem to achieve compression in terms of $p$. 
Goal of the Proposed Method

- Use Kernel machine to do classification
- Produce non-linear classification boundary in the kernel transformed space
- Feature/variable selection will be done in original input space not in the kernel transformed space
- The result will have straightforward probabilistic interpretation
- Extension from two-class to multi-class classification should be natural
Framework for Dimension Selection

For high dimensional problem many dimensions are just noise hence filtering them makes sense. (but how ?)

- Best classifier lies in a much lower dimensional space
- We start with a dimension and then try to add more dimensions sequentially to improve classification
- We choose Gaussian Kernel,

$$K(x_i, x_j | \theta) = \exp \left\{-\frac{||x_i - x_j||^2}{2\theta^2}\right\}$$

**Theorem 1** : If the training data is separable in $S$ then it will be separable in any $\mathcal{S} (\supseteq S)$. For completely separable case,

- Classification performance cannot degrade with inclusion of more dimensions.
- Separating hyperplane in $S$ is also a separating hyperplane in $\mathcal{S} (\supseteq S)$. 
Rough Sketch of Proof

Margin = \( yf(x) \)

Maximal Separating hyperplane in only one dimension

Maximal Separating hyperplane in two dimension

For non-linear kernel based transformation this is not so obvious and the proof is little technical.

**Theorem 2**: Distance (and so does the margin) between any two points is a non-decreasing function of the dimensions.

Proof is straight forward
Problem Formulation

Essentially we are hypothesizing for \( S \subseteq \mathcal{L} = \{1, 2, \ldots, p\} \)

\[
\arg\min_{f_S \in \mathcal{H}_k} \left[ \sum_{i=1}^{n} L(y_i, f_S(x_i)) + \lambda \|f_S\|^2_{\mathcal{H}_k} \right] \leq \arg\min_{f_{\mathcal{L}} \in \mathcal{H}_k} \left[ \sum_{i=1}^{n} L(y_i, f_{\mathcal{L}}(x_i)) + \lambda \|f_{\mathcal{L}}\|^2_{\mathcal{H}_k} \right]
\]

where \( L \) is the binomial deviance (negative log-likelihood).

For an arbitrary \( \Lambda \) dimensional space we may define Gaussian Kernel as,

\[
K_{\Lambda}(x_i, x_j | \theta) = \exp \left\{ - \frac{\sum_{t \in \Lambda} (x_{it} - x_{jt})^2}{2\theta^2} \right\}.
\]

Our objective optimization problem for dimension selection,

\[
\arg\min_{\alpha \in \mathbb{R}^s} \left[ \sum_{i=1}^{n} L_S(y_i, \alpha_i, x_i) + \lambda \sum_{i,j=1}^{n} \alpha_i \alpha_j K_S(x_i, x_j) \right].
\]

Starting from \( S = \emptyset \), we go towards \( S \rightarrow \mathcal{L} \) until desired accuracy is obtained.
Problem Formulation …

In the heart of FIVM lies KLR, so more specifically

\[
H = \frac{1}{n} \sum_{i=1}^{n} \ln \left[ 1 + e^{-y_i f(x_i)} \right] + \lambda \|f\|^2_{H_k} = \frac{1}{n} \ln \left( 1 + e^{-y \cdot K_a} \right) + \lambda a' K a.
\]

To find optimum value of \(a\) we may adopt any optimization method (e.g. NR) until some convergence criteria is satisfied (used by Zhu et al. \(\frac{H_k - H_{k-\Delta k}}{H_k} < \epsilon\))

**Optimality Theorem:** If training data is separable in \(S(\subseteq \mathcal{L})\) and the solution for the equivalent KLR problem in \(S\) and \(\mathcal{L}\) are respectively \(\hat{\beta}^*(s)\) and \(\tilde{\beta}(s)\), then as \(s \to \infty\)

\[
\hat{\beta}^*(s)/s - \tilde{\beta}^*(s)/s \to 0.
\]

Note:- To show optimality of submodel, we are assuming the kernel \(K(x, x')\) is rich enough to completely separate the training data.
To prove optimality theorem we need following two propositions. Under compete severability in $S(\subseteq \mathcal{I})$ the margin maximizing hyperplane in $S(|S|=q)$ can be written as

$$Q(q, x^*, S) = \max_{\beta_0, \beta^*, \|\beta^*\|=1} D^* \text{ subject to } y_i (\beta_0 + h(x_i^*)^T \beta^*) \geq D^*, \ i = 1, \cdots, n$$

Similarly for $\mathcal{I}(|\mathcal{I}|=p>q)$

$$Q(p, x, \mathcal{L}) = \max_{\beta_0, \beta, \|\beta\|=1} D, \text{ subject to } y_i (\beta_0 + h(x_i)^T \beta) \geq D, \ i = 1, \cdots, n$$

Proposition 1: $Q(p, x, \mathcal{L}) \geq Q(q, x^*, S)$.

Proposition 2: $Q(x^*, \mathcal{L}) = Q(q, x^*, S)$, and thus $\hat{\beta}^* = \tilde{\beta}^*$.

We assumed only $q$ many dimensions are true features. Combining above two we obtain the optimality theorem.
FIVM Algorithm

1. Let $\mathcal{S} = \emptyset$, $\mathcal{L} = \{1, 2, \ldots, p\}$ and $k = 1$.

2. For each $l \in \mathcal{L}$, let
   $$f_l(x) = \sum_{i=1}^{n} a_i K_{\mathcal{S}\cup l}(x, x_i),$$ where $K_{\mathcal{S}\cup l}(x_i, x_j) = \exp \left\{ -\frac{\sum_{t \in \mathcal{S}\cup l} (x_{it} - x_{jt})^2}{2\theta^2} \right\}$. Define $\mathbf{K}_l = (K_{\mathcal{S}\cup l}(x_i, x_j))_{i,j=1}^{n}$. Use Newton-Raphson or other function minimization method to find $a$ which minimizes,
   $$H_l = \frac{1}{n} \log \left( 1 + e^{-y^l \mathbf{K}_l^a} \right) + \lambda a'K a.$$

3. Find $l^*$ such that $l^* = \arg\min_{l \in \mathcal{L}} H_l$.

   Let $\mathcal{S} = \mathcal{S} \cup \{l^*\}$, $\mathcal{L} = \mathcal{L} \setminus \{l^*\}$, $H_k = H(x_{l^*})$ and $k = k + 1$.

4. Repeat Steps 2 and 3 until convergence criteria are satisfied.

The dimensions in $\mathcal{S}$ are called imported dimensions.
Convergence Criteria and choice of λ

Convergence criteria used in IVM is not suitable for our purpose.

For $k$-th iteration define $p_k = \text{the proportion of correctly classified training observations with } k \text{ many imported dimensions.}$

The algorithm stops if the ratio $\left| \frac{p_k - p_{k-\Delta k}}{p_k} \right| < \varepsilon$ (a prechosen small number 0.001 ) and $\Delta k = 1$

We choose optimal value of $\lambda$ (regularization parameter) by decreasing it from a larger value to a smaller value until we hit optimum (smallest) misclassification error rate in the training set via grid serach.

We have tested our algorithm for three data sets

- Synthetic data set (two original and eight noisy dimensions)
- Breast Cancer Data of West et al. (2001)
- Colon cancer data set of Alon et al.(1999)
Exploration With Synthetic Data

- Generate 10 means from $\mathcal{N}\left(\left(1.5, 0\right)^\prime, 2I\right)$ and label them +1
- Generate 10 means from $\mathcal{N}\left(\left(0, 1.5\right)^\prime, 2I\right)$ and label them -1
- From each class we generate 100 observations by selecting a mean $\mu_k$ randomly with probability $1/10$ and then generate $\mathcal{N}(\mu_k, I/5)$.

We deliberately add eight more dimensions and filled them with white noise.
Training Results

Note \( n = 200, \theta = 1, \Delta k = 1 \). The stopping criterion is satisfied when \( |S| = 2 \).

With increasing testing sample size classification accuracy of FIVM does not degrade.
Testing Results

We choose $\varepsilon=0.05$ and For $\theta$ we searched over $[2^{-6},2^{6}]$, for $\lambda$ we searched over $[2^{-10},2^{10}]$.

Only those dimensions (i.e. first two) selected by FIVM are used for final classification of test data.

<table>
<thead>
<tr>
<th>SVM Performance</th>
<th>FIVM Performance</th>
<th>$L_1$ Logistic Regression</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta = 2, \lambda = 4$, Train error=0.125</td>
<td>$\theta = 1, \lambda = 0.5$, Train error=0.105</td>
<td>$\lambda = 0.1$, Train error=0.12</td>
</tr>
<tr>
<td>Test error= 0.18, 36 Support points</td>
<td>Test error= 0.125</td>
<td>Test error= 0.145</td>
</tr>
<tr>
<td>Run Time=55 sec</td>
<td>Run Time=51 sec</td>
<td>Run Time=31 sec</td>
</tr>
</tbody>
</table>

FIVM correctly select two informative dimensions.
Exploration With Breast Cancer Data

- Studied earlier by West et al. (2001).
- Tumors were either positive for both the estrogen & progesterone receptors or negative for both receptors.
- Final collection of tumors consisted of 13 estrogen receptor (ER) + lymph node (LN)+tumors, 12 ER-LN+tumors, 12 ER+LN-tumors, and 12 ER- LN-tumors.
- Out of 49, 35 samples are selected randomly as training data.
- Each sample consists of 7129 gene probes.
- Two separate analysis is done 1>ER status, 2>LN status.
- Two convergence parameters are selected $\varepsilon = 0.05$ and $\varepsilon = 0.001$. to study the performance of FIVM.
Breast Cancer Result (ER)
### Breast Cancer Result (ER)

<table>
<thead>
<tr>
<th>$\epsilon = 0.05$</th>
<th>$\epsilon = 0.001$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Four Genes Selected by FIVM</strong></td>
<td><strong>Six Genes Selected by FIVM</strong></td>
</tr>
<tr>
<td>Z84721_cds2, Contains alpha and zeta globin genes and ESTs</td>
<td>Z84721_cds2, Contains alpha and zeta globin genes and ESTs</td>
</tr>
<tr>
<td>X83425, Homo Sapiens LU gene for Lutheran blood group glycoprotein</td>
<td>X83425, Homo Sapiens LU gene for Lutheran blood group glycoprotein</td>
</tr>
<tr>
<td>X55037, Homo Sapiens GATA-3 mRNA</td>
<td>X55037, Homo Sapiens GATA-3 mRNA</td>
</tr>
<tr>
<td>U33147, Human mammaglobin mRNA, complete cds</td>
<td>U33147, Human mammaglobin mRNA, complete cds</td>
</tr>
<tr>
<td>HG1205-HT1205, Collagen, Type IV, Alpha 2, N-Terminus</td>
<td>X03635, Human mRNA for oestrogen receptor</td>
</tr>
</tbody>
</table>

**Testing error:**
- $0.344(\pm 0.032)$ for $\epsilon = 0.05$
- $0.416(\pm 0.027)$ for $\epsilon = 0.001$

**SVM Testing Performance & Tuning Parameters**
- $\theta = 0.5, \lambda = 1$
- $\theta = 1, \lambda = 1$
- Error $= 0.406(\pm 0.052), 12(\pm 4)$ support points
- Error $= 0.41(\pm 0.054), 20(\pm 3)$ support points

Table 2: Estrogen Receptor (ER) Classification Result for Breast Cancer Data
Breast Cancer Result (LN)

3D Plot based on First set of Three selected genes

3D Plot based on Second set of Three selected genes

3D Plot based on First set of Three selected genes

3D Plot based on Second set of Three selected genes
### Breast Cancer Result (LN)

<table>
<thead>
<tr>
<th>$\epsilon = 0.05$</th>
<th>$\epsilon = 0.001$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Three Genes Selected by FIVM</td>
<td>Five Genes Selected by FIVM</td>
</tr>
<tr>
<td>HG3638-HT3849_s at Amyloid Beta (A4) Precursor Protein, Alt. Splice 2</td>
<td>HG3638-HT3849_s at Amyloid Beta (A4) Precursor Protein, Alt. Splice 2</td>
</tr>
<tr>
<td>U02493, Human 54 kDa protein mRNA, complete cds</td>
<td>U02493, Human 54 kDa protein mRNA, complete cds</td>
</tr>
<tr>
<td>U51678, Human small acidic protein mRNA, complete cds</td>
<td>U51678, Human small acidic protein mRNA, complete cds</td>
</tr>
<tr>
<td>HG3432-HT3620_s at Fibroblast Growth Receptor K-Sam, Alt. Splice 3</td>
<td>X80692, Homo Sapiens ERK3 mRNA</td>
</tr>
</tbody>
</table>

Testing error: $0.304(\pm 0.017)$ \hspace{2cm} Testing error: $0.411(\pm 0.022)$

SVM Testing Performance & Tuning Parameters

| $\theta = 0.25$, $\lambda = 2$ | $\theta = 0.5$, $\lambda = 4$ |
| Error $= 0.475(\pm 0.012)$, 25(±4) support points | Error $= 0.475(\pm 0.012)$, 25(±4) support points |

Table 3: Lymph Node (LN) Classification Result for Breast Cancer Data
Alon et al. (1999) described a Gene expression profile 40 tumor and 22 normal colon tissue samples, analyzed with an Affymetrix oligonucleotide array. Final data set contains intensities of 2,000 genes. This data set is heavily benchmarked in classification. We divide it in a training set containing 40 observations and the testing data set having 22 observations randomly. Convergence parameter selected as $\varepsilon = 0.001$. 
FIVM performs better than SVM in all occasions.
Multiclass Extension of FIVM

Recall

\[ H = \frac{1}{n} \sum_{i=1}^{n} \ln \left[ 1 + e^{-y_i f(x_i)} \right] + \lambda \| f \|^2_{\mathcal{H}_k} = \frac{1}{n} \ln \left( 1 + e^{-y \cdot K a} \right) + \lambda a' K a. \]

- This is straight forward if we replace the NLL of bionomial by that of multinomial.
- For M-class classification through kernel multi-logit regression:
  \[ P_r(Y = j | x) = \frac{e^{f_j(x)}}{e^{f_1(x)} + \ldots + e^{f_M(x)}} \text{ with } \sum_m f_m(x) = 0. \]
- We need to minimize regularized NLL as,

\[
H = -\frac{1}{n} \sum_{i=1}^{n} \log p_{y_i}(x_i) + \lambda \| f \|^2_{\mathcal{H}_k} \\
= -\frac{1}{n} \sum_{i=1}^{n} \left[ y_i f(x_i) + \log \left( e^{f_1(x_i)} \ldots + e^{f_C(x_i)} \right) \right] + \lambda \| f \|^2_{\mathcal{H}_k}
\]
Multiclass Extension of FIVM

- Kernel trick works here too, so extension is straightforward.
- Additional Complexity of Multiclass FIVM is proportional to the number of class.

Key Features of FIVM

- Select dimensions decreases regularized NLL the most
- Imported dimensions are the most important candidate biomarkers having highest differential capability
- Unlike other methods (e.g. PCR), our method FIVM achieves data compression in the original feature space
- Dual purpose: Probabilistic classification & data compression
- Multiclass extension of FIVM is straightforward
Open Questions

● What about simultaneous reduction of dimension and observation? (both \( n \) and \( p \))

● How to augment dimensions when dimensions are correlated, with some known (or unknown) correlation structure?

● FIVM like algorithm's selection of \( p \)'s can be compared with other methods. (e.g. Elastic Net, Fussed Lasso)

● Effect of FIVM type dimension selection for doubly penalized methods. (two penalties instead of one)

● Theoretical question: Does FIVM has Oracle inequality?

● Effect of FIVM on the hard/soft thresholding rule?
Some References

4. West et al. Predicting the clinical status of human breast cancer by using gene expression profiles. PNAS
5. Alon et al. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. PNAS
Thank you