Some other techniques

Antanas Verikas
antanas.verikas@hh.se

IDE, Halmstad University

2013

SVM summary

SVM is one of the most successful classifiers. Predictions are based on a function of the form:

$$y(\mathbf{x}, \mathbf{w}) = \sum_{i=1}^{N} w_i K(\mathbf{x}, \mathbf{x}_i) + w_0 \tag{1}$$

where w_i are the model parameters, and $K(\mathbf{x}, \mathbf{x}_i)$ is a kernel function defining one basis function for each sample in the training set.

Although SVM is one of the most successful classifiers, a number of significant disadvantages can be identified.

Drawbacks of SVM

- Although SVMs are relatively sparse, the number of SVs typically grows linearly with the size of the training set, thus, basis functions are unnecessarily liberally used.
- Predictions are not probabilistic. However, the posterior probabilities of class membership are necessary, in many applications.
- In SVM, it is required to estimate the error/margin tradeoff parameter C, which usually entails a cross-validation procedure leading to a waste of data.
- The kernel function must satisfy Mercer's condition, hence, it must be a continuous symmetric kernel of a positive integral operator.

Drawbacks of SVM

- Although SVMs are relatively sparse, the number of SVs typically grows linearly with the size of the training set, thus, basis functions are unnecessarily liberally used.
- Predictions are not probabilistic. However, the posterior probabilities of class membership are necessary, in many applications.
- In SVM, it is required to estimate the error/margin tradeoff parameter C, which usually entails a cross-validation procedure leading to a waste of data.
- The kernel function must satisfy Mercer's condition, hence, it must be a continuous symmetric kernel of a positive integral operator.

Drawbacks of SVM

- Although SVMs are relatively sparse, the number of SVs typically grows linearly with the size of the training set, thus, basis functions are unnecessarily liberally used.
- Predictions are not probabilistic. However, the posterior probabilities of class membership are necessary, in many applications.
- In SVM, it is required to estimate the error/margin tradeoff parameter C, which usually entails a cross-validation procedure leading to a waste of data.
- The kernel function must satisfy Mercer's condition, hence, it must be a continuous symmetric kernel of a positive integral operator.

Drawbacks of SVM

- Although SVMs are relatively sparse, the number of SVs typically grows linearly with the size of the training set, thus, basis functions are unnecessarily liberally used.
- Predictions are not probabilistic. However, the posterior probabilities of class membership are necessary, in many applications.
- In SVM, it is required to estimate the error/margin tradeoff parameter C, which usually entails a cross-validation procedure leading to a waste of data.
- The kernel function must satisfy Mercer's condition, hence, it must be a continuous symmetric kernel of a positive integral operator.

RVM (1)

- RVM is a Bayesian treatment alternative to the SVM. RVM does not suffer from the aforementioned limitations.
- RVM introduces a prior over the model parameters governed by a set of hyper-parameters.
- One hyper-parameter is associated with each parameter, and the most probable values are iteratively estimated from the training data.
- The most compelling feature of the RVM is that it typically utilizes significantly fewer kernel functions compared to the SVM, while providing a similar performance.

RVM (1)

- RVM is a Bayesian treatment alternative to the SVM. RVM does not suffer from the aforementioned limitations.
- RVM introduces a prior over the model parameters governed by a set of hyper-parameters.
- One hyper-parameter is associated with each parameter, and the most probable values are iteratively estimated from the training data.
- The most compelling feature of the RVM is that it typically utilizes significantly fewer kernel functions compared to the SVM, while providing a similar performance.

RVM (1)

- RVM is a Bayesian treatment alternative to the SVM. RVM does not suffer from the aforementioned limitations.
- RVM introduces a prior over the model parameters governed by a set of hyper-parameters.
- One hyper-parameter is associated with each parameter, and the most probable values are iteratively estimated from the training data.
- The most compelling feature of the RVM is that it typically utilizes significantly fewer kernel functions compared to the SVM, while providing a similar performance.

RVM (1)

- RVM is a Bayesian treatment alternative to the SVM. RVM does not suffer from the aforementioned limitations.
- RVM introduces a prior over the model parameters governed by a set of hyper-parameters.
- One hyper-parameter is associated with each parameter, and the most probable values are iteratively estimated from the training data.
- The most compelling feature of the RVM is that it typically utilizes significantly fewer kernel functions compared to the SVM, while providing a similar performance.

RVM (2)

- For two-class classification, a target can be chosen such that $t_n \in \{0, 1\}$.
- A Bernoulli distribution can be adopted for $p(t|\mathbf{x})$, since only two values (0 and 1) are possible.
- The logistic sigmoid function $\sigma(y) = 1/(1 + e^{-y})$ is applied to $y(\mathbf{x})$ to generalize the linear model.

RVM (3)

Following the definition of the Bernoulli distribution, the likelihood is written

$$p(\mathbf{t}|\mathbf{w}) = \prod_{n=1}^{N} \sigma\{y(\mathbf{x}_n; \mathbf{w})\}^{t_n}[1 - \sigma\{y(\mathbf{x}_n; \mathbf{w})\}]^{1-t_n} \qquad (2)$$

A Gaussian prior over model parameters is used:

$$p(\mathbf{w}|\boldsymbol{\alpha}) = \prod_{n=1}^{N} \frac{\sqrt{\alpha_n}}{\sqrt{2\pi}} \exp\left(-\frac{\alpha_n w_n^2}{2}\right) \qquad (3)$$

where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_N)^T$ are the hyper-parameters controlling the strength of the prior over its associated model parameter. The prior is Gaussian, but conditioned on $\boldsymbol{\alpha}$.

RVM (4)

For a certain α, the posterior weight distribution conditioned on the data can be obtained using Bayes' rule:

$$p(\mathbf{w}|\mathbf{t}, \alpha) = \frac{p(\mathbf{t}|\mathbf{w})p(\mathbf{w}|\alpha)}{p(\mathbf{t}|\alpha)} \tag{4}$$

where $p(\mathbf{t}|\mathbf{w})$ is the likelihood, $p(\mathbf{w}|\alpha)$ is the prior, and $p(\mathbf{t}|\alpha)$ is referred to as evidence.

RVM (5)

The parameters $\mathbf{w}$ cannot be obtained analytically. Therefore, a Laplacian approximation procedure is used.

- Since $p(\mathbf{w}|\mathbf{t}, \boldsymbol{\alpha})$ is linearly proportional to $p(\mathbf{t}|\mathbf{w}) \times p(\mathbf{w}|\boldsymbol{\alpha})$, the most probable weights $\mathbf{w}_{MP}$ are obtained by iterative maximization of

$$\log\{p(\mathbf{t}|\mathbf{w})p(\mathbf{w}|\boldsymbol{\alpha})\} = \sum_{n=1}^{N}[t_n \log y_n + (1-t_n)\log(1-y_n)] - \frac{1}{2}\mathbf{w}^T\mathbf{A}\mathbf{w}$$

where $y_n = \sigma\{y(\mathbf{x}_n; \mathbf{w})\}$ and $\mathbf{A} = \texttt{diag}(\alpha_1, ..., \alpha_N)$ are obtained using the current values of $\boldsymbol{\alpha}$. This is a penalized logistic log-likelihood function.

RVM (6)

- Laplace's method is a Gaussian approximation to the log-posterior. At convergence, the negative Hessian $\mathbf{H}$ represents the inverse covariance $\boldsymbol{\Sigma}$ of the Gaussian approximation to the posterior centered at $\mathbf{w}_{MP}$:

$$\mathbf{H} = \nabla_{\mathbf{w}} \nabla_{\mathbf{w}} \log p(\mathbf{w}|\mathbf{t}, \boldsymbol{\alpha})|_{\mathbf{w}_{MP}} = -(\boldsymbol{\Phi}^T \mathbf{B} \boldsymbol{\Phi} + \mathbf{A}) \qquad (5)$$

  where $\mathbf{B} = \mathtt{diag}(\beta_1, \beta_2, ..., \beta_N)$ with $\beta_n = \sigma\{y(\mathbf{x}_n; \mathbf{w}_{MP})\}[1 - \sigma\{y(\mathbf{x}_n; \mathbf{w}_{MP})\}]$, and $\boldsymbol{\Phi}$ is the "design" matrix with $\boldsymbol{\Phi}_{nm} = K(\mathbf{x}_n, \mathbf{x}_{m-1})$ and $\boldsymbol{\Phi}_{n1} = 1$.
- At the mode of $p(\mathbf{w}|\mathbf{t}, \boldsymbol{\alpha})$, using (5) and the fact that $\nabla_{\mathbf{w}} \log p(\mathbf{w}|\mathbf{t}, \boldsymbol{\alpha})|_{\mathbf{w}_{MP}} = \boldsymbol{\Phi}^T(\mathbf{t} - \mathbf{y}) - \mathbf{A}\mathbf{w} = 0$, we can write:

$$\boldsymbol{\Sigma} = (\boldsymbol{\Phi}^T \mathbf{B} \boldsymbol{\Phi} + \mathbf{A})^{-1} \qquad (6)$$

$$\mathbf{w}_{MP} = \mathbf{A}^{-1} \boldsymbol{\Phi}^T (\mathbf{t} - \mathbf{y}) \qquad (7)$$

RVM (7)

- Using the statistics $\boldsymbol{\Sigma}$ and $\mathbf{w}_{MP}$ of the Gaussian approximation, the hyper-parameters $\boldsymbol{\alpha}$ are updated using

$$\alpha_i^{\texttt{new}} = \gamma_i / w_i^2 \tag{8}$$

  where w_i is the ith posterior parameter mean and

$$\gamma_i \equiv 1 - \alpha_i \Sigma_{ii} \tag{9}$$

  where Σ_{ii} is the ith diagonal element of the posterior covariance, computed with the current $\boldsymbol{\alpha}$ values.
- Each $\gamma_i \in [0, 1]$ can be interpreted as a measure of how "well-determined" its corresponding parameter w_i is by the data.

RVM (8)

- For α_i large, where w_i is highly constrained by the prior, $\Sigma_{ii} \approx \alpha_i^{-1}$ and it follows that $\gamma_i \approx 0$. Conversely, when α_i is small and w_i fits the data, $\gamma_i \approx 1$.
- During the optimization process, many α_i will have large values, and thus, the corresponding model parameters are pruned out, implementing sparsity. The optimization process typically continues until the maximum change in α_i values is below a certain threshold or the maximum number of iterations is reached.
- The Gaussian approximation is sometimes considered as a weakness of the method.

Black boxes

- MLP, SVM, and RVM are often considered as "black boxes".
- The classifiers are not transparent enough, in the sense that it is difficult to explain reasons behind different decisions.
- Various non-linear transformations applied to extract features make interpretation very difficult.

Black boxes

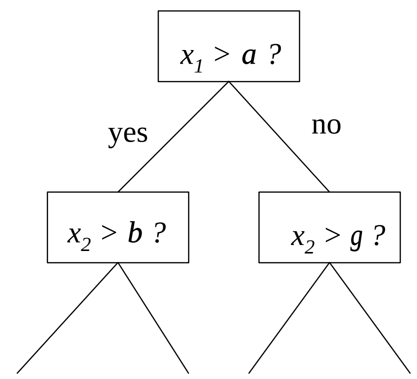
- MLP, SVM, and RVM are often considered as “black boxes”.
- The classifiers are not transparent enough, in the sense that it is difficult to explain reasons behind different decisions.
- Various non-linear transformations applied to extract features make interpretation very difficult.

Black boxes

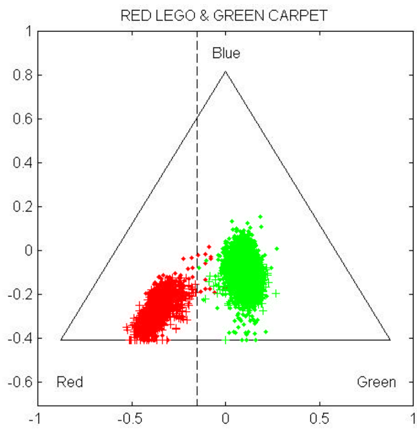
- MLP, SVM, and RVM are often considered as “black boxes”.
- The classifiers are not transparent enough, in the sense that it is difficult to explain reasons behind different decisions.
- Various non-linear transformations applied to extract features make interpretation very difficult.

Decision tree

- Splits data into smaller and smaller subsets.
- Each split increases node purity (e.g. Gini index)
- Splits are usually made along variable axes ⇒ a subdivision into "hypercubes".
- Backwards pruning is important

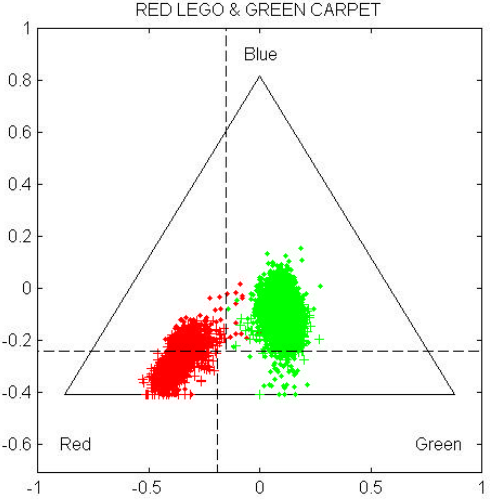


Example: Decision tree, first cut along x_1



- Training error $= 0.06\%$; Test error $= 0.07\%$.
- Rule: IF $x_1 < -0.1515$ THEN red otherwise green.
- No suitable cut along x_2 axis after the first cut along x_1.

Example: Decision tree, first cut along x_2



Random forests

Random forests combine many unpruned trees. Can handle both continuous and categorical variables. A general tool for data mining.

- Data classification
- Prediction
- Analysis of variable importance
- Analysis of data similarities
- Outlier detection
- Replacing missing values

Random forests

Random forests combine many unpruned trees. Can handle both continuous and categorical variables. A general tool for data mining.

- Data classification
- Prediction
- Analysis of variable importance
- Analysis of data similarities
- Outlier detection
- Replacing missing values

Random forests

Random forests combine many unpruned trees. Can handle both continuous and categorical variables. A general tool for data mining.

- Data classification
- Prediction
- Analysis of variable importance
- Analysis of data similarities
- Outlier detection
- Replacing missing values

Random forests

Random forests combine many unpruned trees. Can handle both continuous and categorical variables. A general tool for data mining.

- Data classification
- Prediction
- Analysis of variable importance
- Analysis of data similarities
- Outlier detection
- Replacing missing values

Random forests

Random forests combine many unpruned trees. Can handle both continuous and categorical variables. A general tool for data mining.

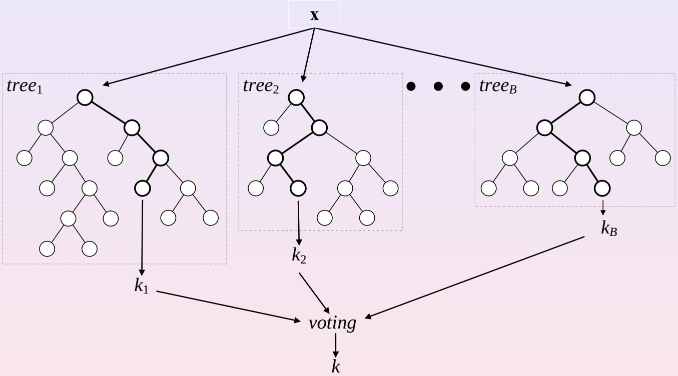
- Data classification
- Prediction
- Analysis of variable importance
- Analysis of data similarities
- Outlier detection
- Replacing missing values

Random forests

Random forests combine many unpruned trees. Can handle both continuous and categorical variables. A general tool for data mining.

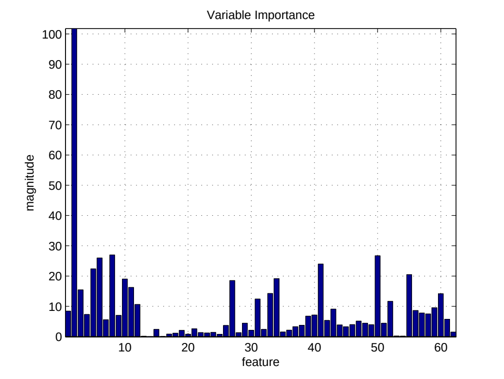
- Data classification
- Prediction
- Analysis of variable importance
- Analysis of data similarities
- Outlier detection
- Replacing missing values

A committee of decision trees



1. Each tree is grown on a bootstrap sample of the training set.
2. At each node, n variables are randomly selected out of the N available.

Variable importance

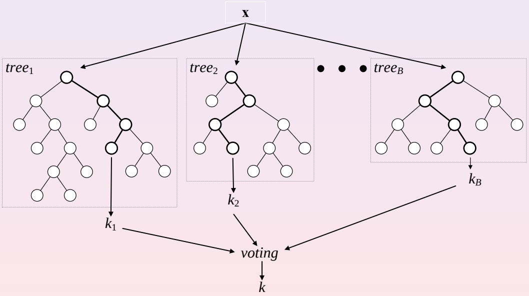


The importance measure $\overline{D}_j$ for variable x_j is given by

$$\overline{D}_j = \frac{1}{B} \sum_{b=1}^{B} (R_b^{oob} - R_{b,j}^{oob}) \quad (10)$$

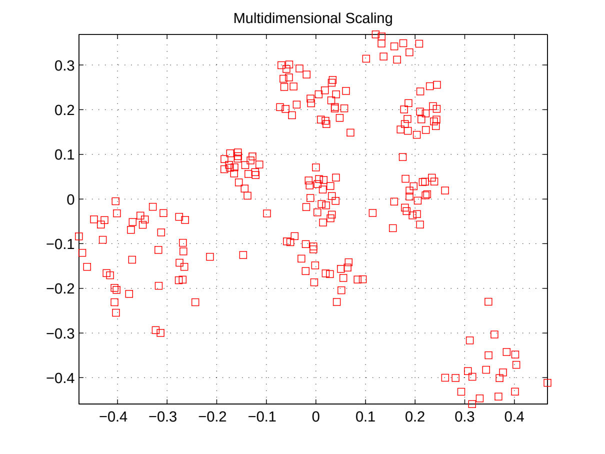
where R_b^{oob} and $R_{b,j}^{oob}$ are the number of correct classifications by the tree b before and after values of x_j are randomly permuted.

Data proximity matrix



1. For each tree grown, the data are run down the tree.
2. If two observations $\mathbf{x}_i$ and $\mathbf{x}_j$ occupy the same terminal node of the tree, $prox(i, j)$ is increased by one.
3. When RF is grown, the proximities are divided by the number of trees in RF.

Analysis of data similarity



Fuzzy rules

- Fuzzy rules are attributed to the class of transparent models.
- Concerning classification, the model is a collection of fuzzy rules R_j of the following form:

  $R_j : \texttt{IF } x_1 \texttt{ is } A_{j1} \texttt{ AND } \ldots \texttt{ AND } x_n \texttt{ is } A_{jn} \texttt{ THEN class } C_q \texttt{ with } z_j^q$

  where $A_{ji}(i = 1, \ldots, n)$ are fuzzy sets defined over the input variables x_i, C_q is a class label and z_j^q is a rule weight.

Membership function

Each fuzzy set is represented by a membership function, a Gaussian function, for example:

$$\mu_{ji} = \exp\left(- \frac{(x_i - c_{ji})^2}{\sigma_{ji}^2} \right) \tag{11}$$

where c_{ji} and σ_{ji} is the center and the width of the Gaussian function, respectively.

Rule weights

There are various ways to determine the rule weights z_j^q:

$$z_j^q = \frac{\sum_{\mathbf{x}_p \in C_q} \mu_{\mathbf{A}_j}(\mathbf{x}_p) - \sum_{\mathbf{x}_p \notin C_q} \mu_{\mathbf{A}_j}(\mathbf{x}_p)}{\sum_{p=1}^{N} \mu_{\mathbf{A}_j}(\mathbf{x}_p)} \tag{12}$$

where N is the number of training patterns and the matching degree of the input pattern $\mathbf{x}_p$ with the antecedent part $\mathbf{A}_j = (A_{j1}, ..., A_{jn})$ is calculated using a T-norm

$$\mu_{\mathbf{A}_j}(\mathbf{x}_p) = T(\mu_{A_{j1}}(x_{p1}), ..., \mu_{A_{jn}}(x_{pn})) \tag{13}$$

The min operator can be used as a T-norm operator, for example.

Classification rule

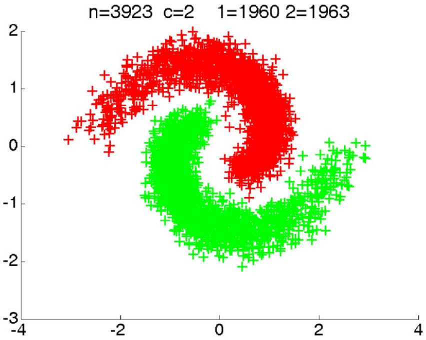
A winning rule is usually used to make a decision. Thus, given a rule base S consisting of L rules, an input pattern $\mathbf{x}_p$ is assigned to the class q if

$$q = \arg\max_{k} \{ T[\mu_{\mathbf{A}_j}(\mathbf{x}_p), z_j^k],\ j = 1, ..., L \} \tag{14}$$

where T is the product T-norm operator.

Initial structure and optimization

Initial structure from clustering, for example. Optimization by genetic search.


n=3923 c=2 1=1960 2=1963
2
1
0
-1
-2
-3
-4
-2
0
2
4