Robust Probability Support Vector Machines

Manuel Gonzalo Carvajal

Instituto de Matemática Interdisciplinar, Universidad Complutense de Madrid

Support Vector Machines, SVMs for short, are Machine Learning algorithms mainly used to solve the problem of binary classification. These algorithms work by constructing a classifier function from the solution of an optimization problem that involves a set of previously labeled data. The following problem corresponds to soft-margin SVM, which is one of the most simple SVM models that can be considered and usually taken as reference when testing new proposals.

$$\begin{aligned} \min_{\mathbf{w},b,\xi} \ & \frac{1}{2}||\mathbf{w}||_2^2 + C\sum_{i=1}^{m}\xi_i & \\ \text{s.t.} \ & y_i(\mathbf{w}^\top\mathbf{x}_i + b) \geq 1 - \xi_i & \forall i = 1,\dots,m \\ & \xi_i \geq 0 & \forall i = 1,\dots,m \end{aligned}$$

Here $\mathbf{w}$ and b are, respectively, the normal vector and the bias of the hyperplane obtained as the classification boundary and ξ is a slack variable related to classification errors. The vectors $\mathbf{x}_i$ are the observations in the training set and y_i their labels.
Our work consists on the proposal and test of novel SVM type models obtained by introducing the pinball loss, which has already been used to improve basic SVM in [3], into the probability SVMs presented in [4] to increase their robustness to noise in the training data.

Proposed Models

Probability support vector machines [4] are designed to provide with values between 0 and 1 instead of just ± 1 when classifying an observation. This can directly be interpreted as a probability of that observation to be in certain class. To accomplish this we simply add an additional restriction to the optimization problem given above:

$$0 \leq \mathbf{w}^\top\mathbf{x}_i + b \leq 1 \quad \forall i = 1,\dots,m$$

We also need to adapt the decision boundary to the value $\mathbf{w}^\top\mathbf{x}_i + b = 0.5$ and that is done by the following modification of the first restriction:

$$y_i(\mathbf{w}^\top\mathbf{x}_i + b) \geq 1 - \xi_i \longrightarrow y_i(\mathbf{w}^\top\mathbf{x}_i + b - 0.5) \geq 0.5\varepsilon - \xi_i \quad \forall i = 1,\dots,m$$

These 2 modifications lead to the model known as probability support vector machine, or PSVM for short. We can further modify the problem by introducing a term in the objective function that rewards classifications with probability close to 0 or 1 according to the class, and the resulting model is called conditional probability support vector machine, or CPSVM for short.

The standard loss function used in the model presented at the beginning is related to the minimal distance, which implies that it is sensible to noise, particularly to noise close of the decision boundary. To solve this problem [3] proposes the use of the pinball loss, that is related to quantile distance instead. This is introduced in the model by changing the second restriction:

$$\xi_i \geq 0 \longrightarrow y_i(\mathbf{w}^\top\mathbf{x}_i + b - 0.5) \leq 0.5\varepsilon + \frac{1}{\tau}\xi_i \quad \forall i = 1,\dots,m$$

Combining all the previous modification we obtain the following novel problem that we will denote as pin-PSVM, or pin-CPSVM if the additional term for the objective function is included (colored in red in the equation). We also highlighted in blue the modified restriction corresponding to pinball loss.

$$\begin{aligned} \min_{\mathbf{w},b,\xi} \ & \frac{1}{2}||\mathbf{w}||_2^2 + \frac{C_1}{\varepsilon}\sum_{i=1}^{m}\xi_i \ \textcolor{red}{-C_2\sum_{i=1}^{m} y_i(\mathbf{w}^\top\mathbf{x}_i + b)} & \\ \text{s.t.} \ & y_i(\mathbf{w}^\top\mathbf{x}_i + b - 0.5) \geq 0.5\varepsilon - \xi_i & \forall i = 1,\dots,m \\ & \textcolor{blue}{y_i(\mathbf{w}^\top\mathbf{x}_i + b - 0.5) \leq 0.5\varepsilon + \frac{1}{\tau}\xi_i} & \forall i = 1,\dots,m \\ & 0 \leq \mathbf{w}^\top\mathbf{x}_i + b \leq 1 & \forall i = 1,\dots,m \end{aligned}$$

By applying the Karush-Kuhn-Tucker conditions the dual formulation of the problem can be obtained. This is useful since the dual formulation allows to directly introduce different kernels to perform non-linear classification if desired and it is also better suited to be solved with computational methods, as it is directly expressed as a quadratic programming problem. The resulting dual formulation is:

$$\begin{aligned} \min_{\alpha,\beta,\gamma} \ & \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}(\alpha_i y_i + \beta_i - \gamma_i)(\alpha_j y_j + \beta_j - \gamma_j)\mathbf{x}_i^\top\mathbf{x}_j + \sum_{i=1}^{m}\left[\gamma_i - \frac{1}{2}\alpha_i(y_i + \varepsilon)\right] \\ \text{s.t.} \ & \sum_{i=1}^{m}(\alpha_i y_i + \beta_i - \gamma_i) = 0 \\ & \textcolor{blue}{-\tau\frac{C_1}{\varepsilon}} + \textcolor{red}{C_2} \leq \alpha_i \leq \frac{C_1}{\varepsilon} + \textcolor{red}{C_2}\ , \ \beta_i, \gamma_i \geq 0 \ \ \forall i = 1,\dots,m \end{aligned}$$

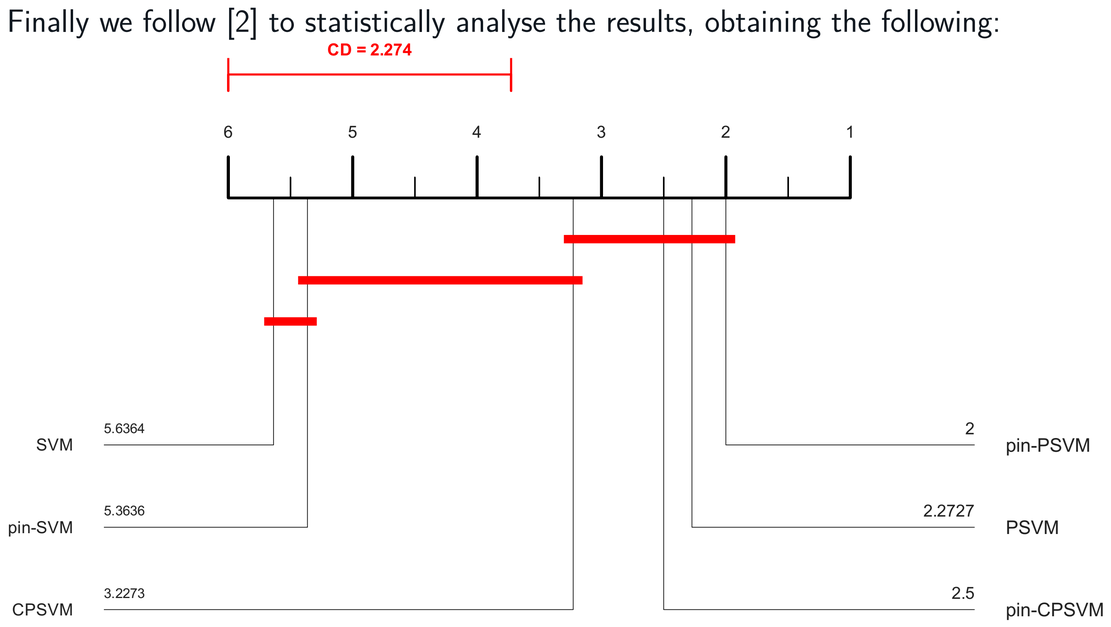
Numerical Experiment

In order to test our proposed models we considered a set of 11 databases. We started using a grid-search strategy to find the best performing, in terms of balance accuracy, set of hyperparameters for every model and database without introducing noise.
Then we fixed the best hyperparameters for each case and solved the problem again, this time introducing different levels of uniform noise on the training data. We performed 100 repetitions of noise for each case and obtained our measurements as the mean of those repetitions.

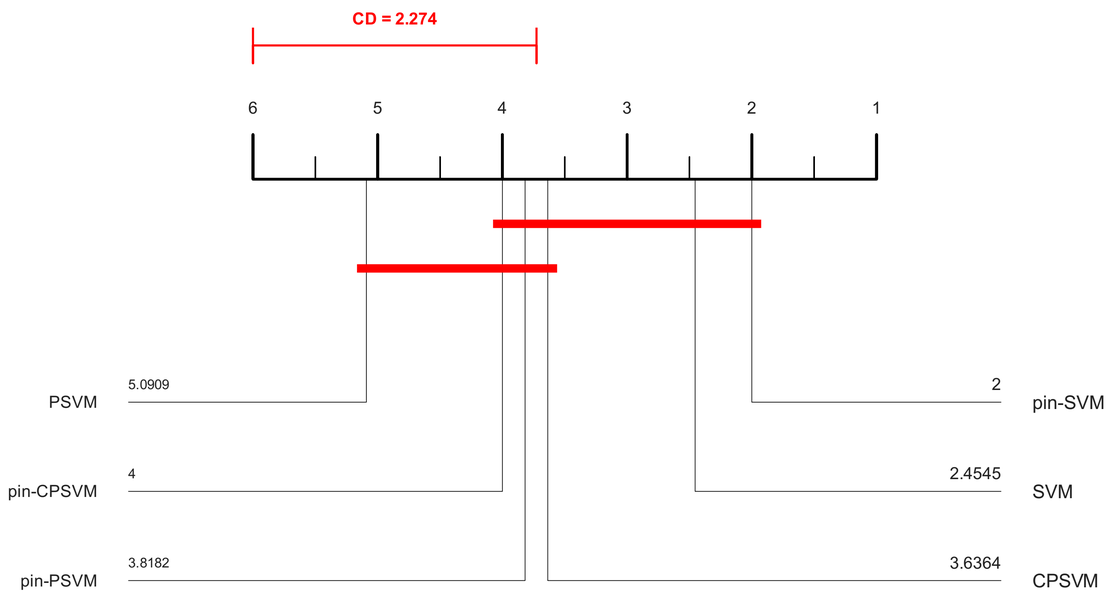
As stated in the beginning, we are interested in the robustness of these models to noise in data, so we calculated the mean relative difference in norm of the values of $\mathbf{w}$ and b obtained with and without noise:

$$R_{\mathbf{w}} = \frac{\overline{||\mathbf{w} - \mathbf{w}_0||}}{||\mathbf{w}_0||} \quad \text{and} \quad R_b = \frac{\overline{|b - b_0|}}{|b_0|}$$

Finally we follow [2] to statistically analyse the results, obtaining the following:



The above graph shows the mean rank (lower is better) over the 11 databases tested in the experiment for each model when considering the robustness measure for the bias of the hyperplane, R_b. Models connected by a red line are not significantly separated with 95% confidence. We observe that the introduction of pinball loss improves the performance of every model. We also note that probability SVM models are more robust than classical SVM in terms of the bias of the hyperplane.



Studying now the results for $R_{\mathbf{w}}$ we find that pinball loss has a positive effect when applied to SVM and PSVM but negative for CPSVM. In this case we observe that SVM is more robust that probability SVMs in terms of $\mathbf{w}$. This contrast in the behaviour of $\mathbf{w}$ and b agrees with the general principle of variance-bias balance. In this context variance is related to changes in $\mathbf{w}$ and bias with changes in b, so it is expected that more complex models show higher variance and lower bias when compared with simpler ones.

References

[1] M. Carrasco et al. "A study of PSVM and CPSVM models: analysis, correction, and application in operations research". In: *Annals of Operation Research* (2025).
[2] J. Demsar. "Statistical comparisons of classifiers over multiple data set". In: *Journal of Machine Learning Research* 7 (2006).
[3] X. Huang, L. Shi, and J.A.K. Suykens. "Support Vector Machine Classifier with Pinball Loss". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 36.5 (2014).
[4] Y. Shao et al. "Twin SVM for conditional probability estimation in binary and multiclass classification". In: *Pattern Recognition* 136 (2023).