Memetic evolution of training sets with adaptive radial basis kernels for support vector machines

Jakub Nalepa, Wojciech Dudzik and Michal Kawulok

Silesian University of Technology, Gliwice, Poland

{jakub.nalepa, wojciech.dudzik, michal.kawulok}@polsl.pl

Silesian University of Technology

Context

Main problems of SVMs

- High computational complexity of training $O(t^3)$
- High memory complexity of training $O(t^2)$
- Need to fine-tune the models with hyperparameters
- Classification time linearly depends on number of SV ($O(S)$)

Contribution

- Novel memetic algorithm for evolving reduced training set
- Proposed adaptive radial basis function with γ hyperparameter specific to a training vector

Memetic Evolution of training set for SVM

Algorithm 1 Memetic evolution of SVM training sets.

$P'_{\text{best}} \leftarrow \emptyset$, $P_{\text{best}} \leftarrow \emptyset$, $\mathcal{S}_{\text{pool}} \leftarrow \emptyset$, $\mathcal{S}_{\text{best}} \leftarrow \emptyset$
$(C, \gamma) \leftarrow$ Grid search over a random $\boldsymbol{T'}$ of size $c \cdot K$
$\vec{\gamma} \leftarrow \{\gamma/10, \gamma, 10 \cdot \gamma, 100 \cdot \gamma, 1000 \cdot \gamma\}$
for all γ_i in $\vec{\gamma}$ **do** γ's are sorted (ascendingly)
  $P \leftarrow$ Generate population$(N, C, \gamma_i, \boldsymbol{T})$
  $P'_{\text{best}} \leftarrow$ Find best individual(P)
  if $\eta(P'_{\text{best}}) > \eta(P_{\text{best}})$ **then**
    while termination condition **not** met **do**
      $P' \leftarrow$ Crossover(P)
      $P' \leftarrow$ Educate(P')
      $P' \leftarrow$ Mutate(P')
      $P' \leftarrow$ Calculate fitness$(P', \mathcal{S}_{\text{best}})$
      $\mathcal{S}_{\text{pool}} \leftarrow$ Update SV pool(P')
      $P^{\text{SI}} \leftarrow$ Create super individuals$(\mathcal{S}_{\text{pool}})$
      $P \leftarrow$ Post select(P, P', P^{SI})
      $P'_{\text{best}} \leftarrow$ Find best individual(P)
      Adapt(P)
    if $\eta(P'_{\text{best}}) > \eta(P_{\text{best}})$ **then**
      Add SV(P'_{best}) to $\mathcal{S}_{\text{best}}$
      $\boldsymbol{T} \leftarrow$ Shrink$(\boldsymbol{T}, P)$
      $P_{\text{best}} \leftarrow P'_{\text{best}}$
   Reset LGA, $\mathcal{S}_{\text{pool}} \leftarrow \emptyset$
return P_{best}

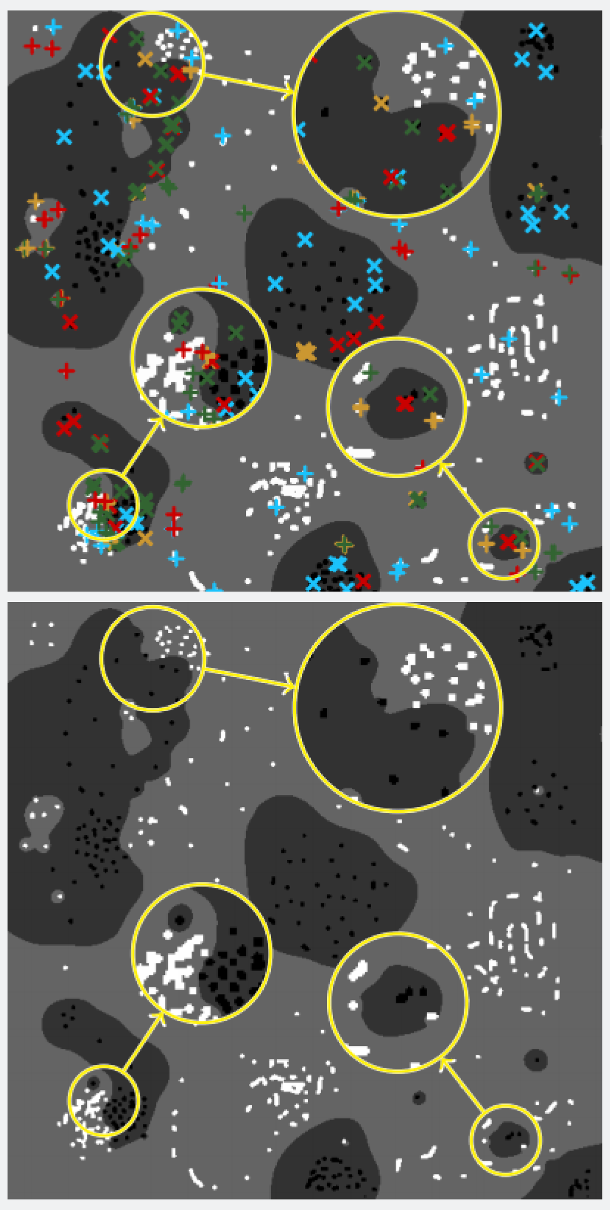

Fig. 1. Assigning different γ's in the RBF kernel to different $\boldsymbol{T}$ vectors can help better "model" the SVM hyperplane

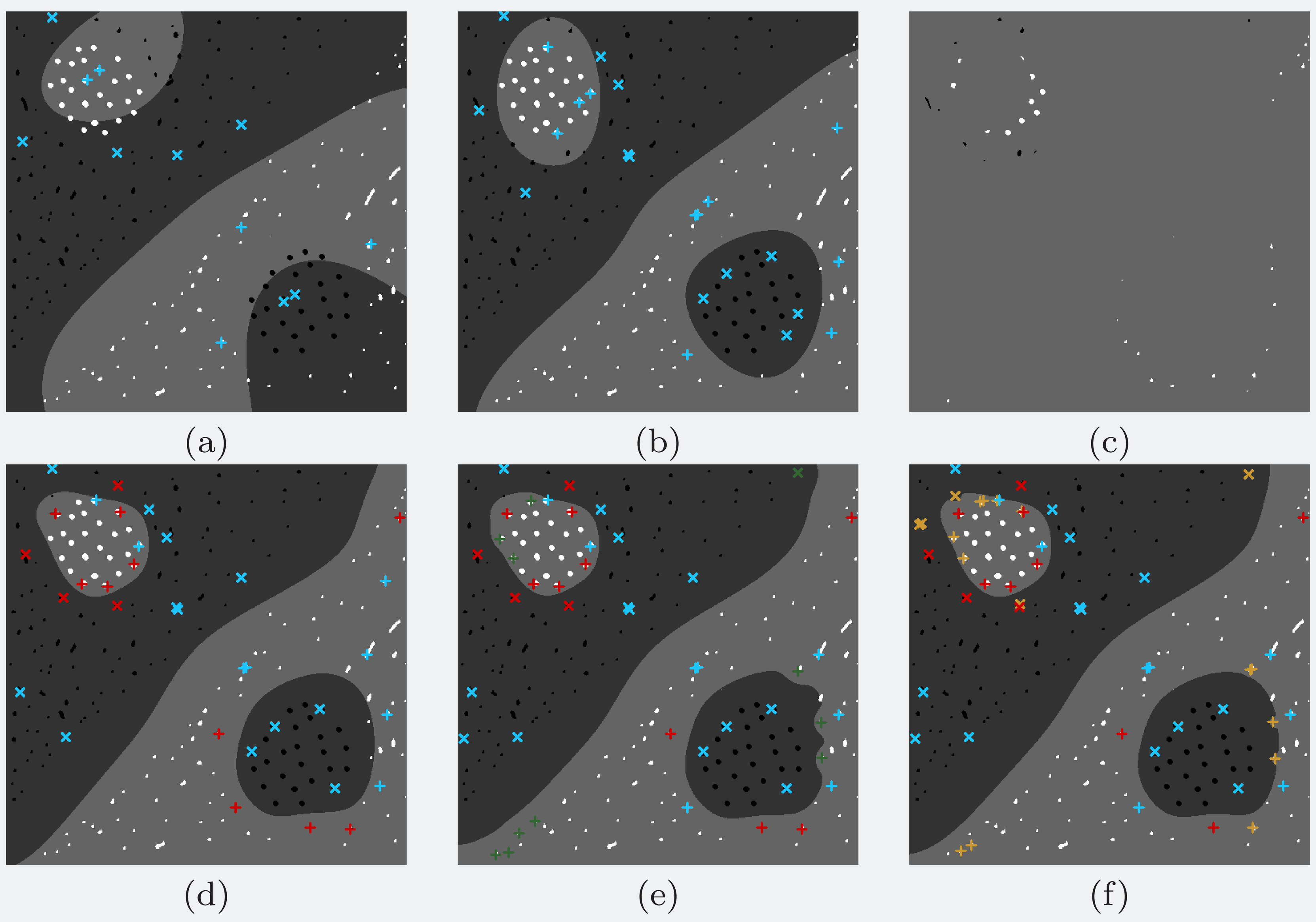

Fig. 2. Visualization of memetic algorithm run. (a) presents the best solution from the initial population, (b) The best solution after finishing evolution with first γ from $\vec{\gamma}$, (c) Shrank training set that will be used in next iteration with subsequent γ. The shrinking procedure is based on the whole population, (d) Solution after second evolution has ended. Added new support vectors marked with red color crosses, (e) Adding next γ value marked with green vectors provided worse classification performance, these support vectors will be removed, (f) final solution for given dataset containing three different γ values.

Experimental Validation

Datasets

We used 96 datasets that were divided into 5 folds containing training, validation ($\boldsymbol{V}$) and test set (Ψ) in the 3:1:1 proportion, respectively. Each evolutionary algorithm was run 10 times per fold.

Experimental Results

Tab. 1 The ranking test over MCC (together with the statistical importance of the differences between our MA and the corresponding approach), for various imbalance ratio ranges. The meanings of ns, *, **, ***, and ****: $p > 0.05$, $p \le 0.05$, $p \le 0.01$, $p \le 0.001$, and $p \le 0.0001$. The best results are boldfaced.

Algorithm	Q1	Q2	Q3	Q4	All
GNB	7.63**	9.04***	8.58****	6.50****	7.94****
LR	6.08ns	5.79ns	5.75ns	5.29ns	5.73****
k-NN(3)	5.33ns	4.88ns	4.67ns	5.50ns	5.09*
k-NN(5)	5.50ns	5.21ns	5.67ns	5.67ns	5.51***
k-NN(7)	5.21ns	5.63ns	6.58**	6.25**	5.92****
SVM(Linear)	6.67ns	5.63ns	6.42**	5.71**	6.10****
SVM(Poly)	5.92ns	4.88ns	5.38ns	4.75ns	5.23**
SVM(RBF)	7.92**	10.71****	9.83****	8.13****	9.15****
MASVM	5.17ns	4.92ns	4.79ns	5.04ns	4.98ns
MASVM(MG)	4.75ns	4.46ns	3.83ns	4.00ns	4.26ns
Ours	**3.75**	**4.42**	**2.92**	**3.21**	**3.57**

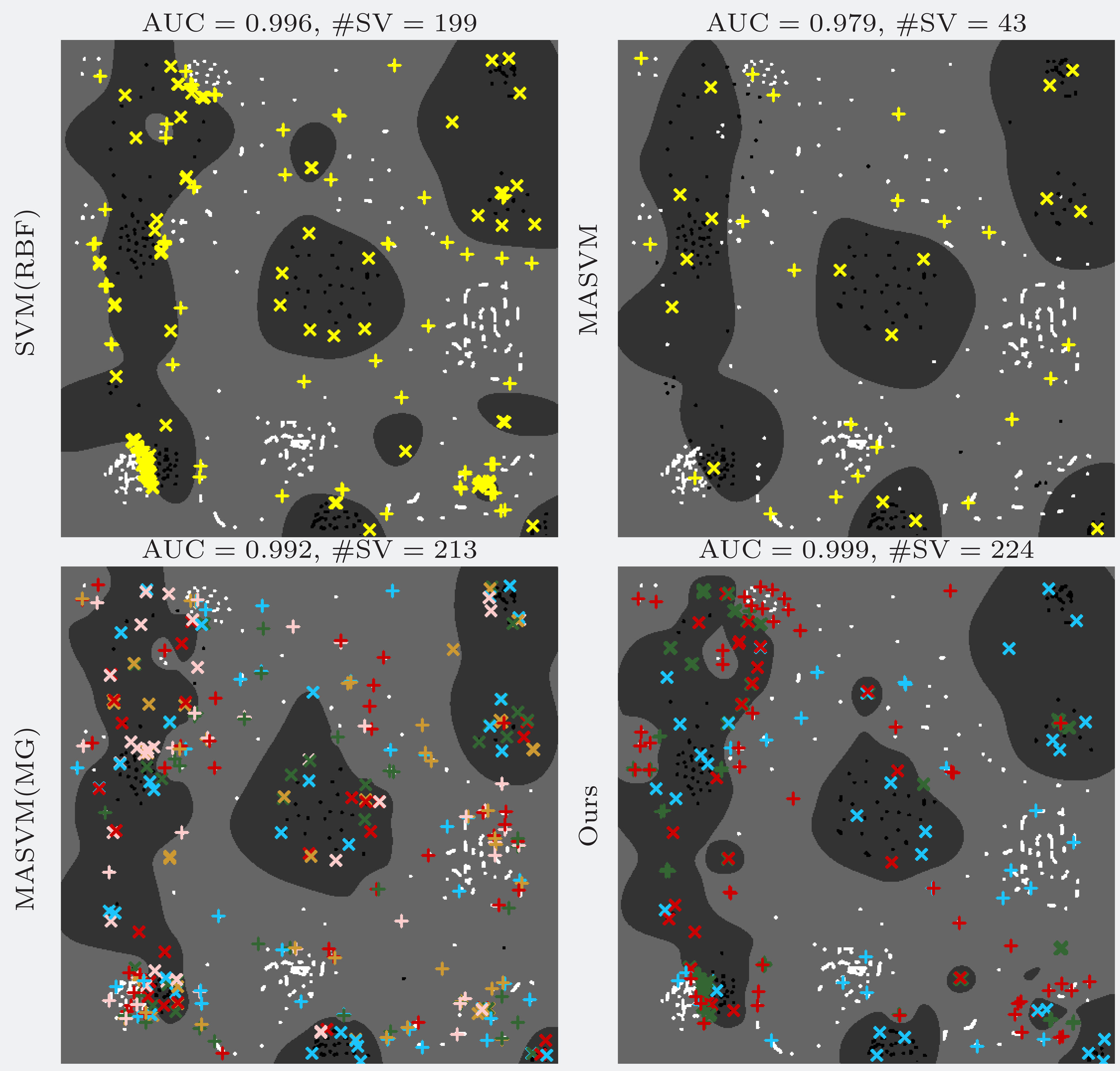

Fig. 3 Example visualizations of the SVs and decision hyperplanes obtained for different methods, together with the AUC (over Ψ) and the number of SVs (#SV). For MASVM(MG) and the proposed MA, we have: $\gamma = 10^2$ (blue), $\gamma = 10^3$ (red), $\gamma = 10^4$ (green), and $\gamma = 10^5$ (orange).

Conclusions

- Our technique outperforms SVMs optimized using other evolutionary methods and other supervised learners.
- It delivers consistent results across sets of various characteristics.
- Our technique can be easily applied in imbalanced classification, where it outperformed all other methods.
- Assigning various γ's to different training vectors is useful in heterogeneous parts of the input space, as visually shown for our 2D datasets.

Acknowledgements

This work was supported by the National Science Centre (DEC-2017/25/B/ST6/00474). WD was co-financed by the European Union through the European Social Fund (POWR.03.02.00-00-I029) and by the Silesian University of Technology (02/020/BKM18/0155)

European Funds

Republic of Poland

European Union
European Social Fund

AIDA