BALANCING PREDICTIVE RELEVANCE OF LIGAND BIOCHEMICAL ACTIVITIES

Marek Pecha

Department of Applied Mathematics, FEECS, VŠB-TU Ostrava

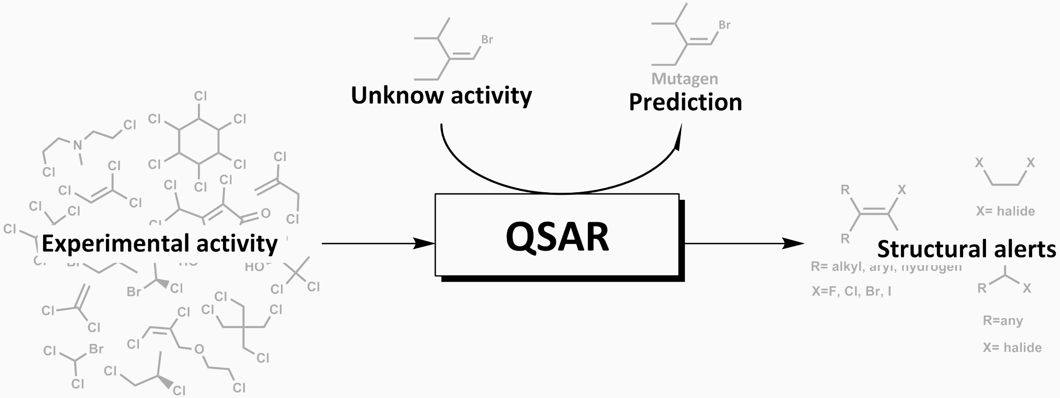
Czech Academy of Sciences, Institute of Geonics

BOS/SOR 2020, Warsaw [online]
December 15, 2020

Outline

- Supervised Biochemical Modelling
- Support Vector Machines
- No-bias data classification
- Model calibration
- PermonSVM
- Benchmarks
- Conclusions

Supervised Biochemical Modelling



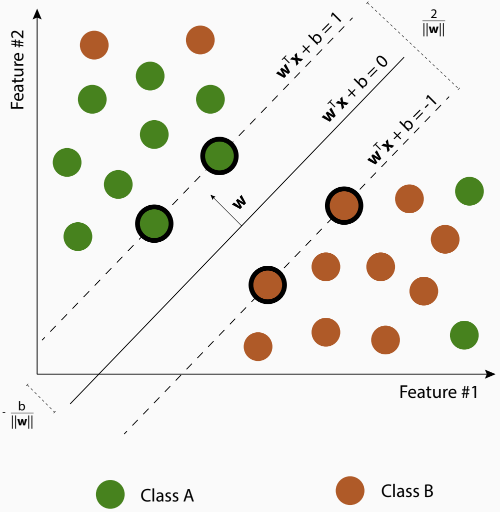
http://bio-hpc.eu/research-lines/qsar/

Support Vector Machines — Soft-margin

The SVM solves a problem of finding a classification model in a form of maximal-margin hyperplane such that

$$H = \langle \boldsymbol{w}, \boldsymbol{x} \rangle + b, \tag{1}$$

where $\boldsymbol{w}$ is a normal vector of hyperplane H and b is its bias alongside origin. The points which lies on geometric margin $\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b = \pm 1$ are called support vectors.



Support Vector Machines — Soft-margin

The problem of finding the hyperplane can be formulated as a constrained optimization problem in the following primal formulation:

$$\arg\min_{\boldsymbol{w},\ b,\ \xi_i} \ \frac{1}{2}\langle \boldsymbol{w}, \boldsymbol{w} \rangle + C \sum_{i=1}^{m} \xi_i \ \text{ s.t. } \begin{cases} y_i \left(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle - b \right) \geq 1 - \xi_i, \\ \xi_i \geq 0, \ \ i \in \{1, 2, \ldots, m\}, \end{cases} \tag{2}$$

where $\xi_i := \max\left(0, 1 - [\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle - b]\right)$ is hinge loss function quantifies error between current and correct classification of sample $\boldsymbol{x}_i$.

The variable $C \in \mathbb{R}^+$ is a penalty that penalizes misclassification error.

The value of C is user-defined or determined using hyperparameter optimization (HyperOpt) techniques, e.g. grid-search combined with cross-validation.

Support Vector Machines — Soft-margin

Exploiting the Lagrange duality and evaluating Karush-Kuhn-Tucker conditions, we transform (2) into the dual formulation so that

$$\arg\min_{\boldsymbol{\alpha}} \ \frac{1}{2}\boldsymbol{\alpha}^T \boldsymbol{Y}^T \boldsymbol{K} \boldsymbol{Y} \boldsymbol{\alpha} - \boldsymbol{\alpha}^T \boldsymbol{e} \ \text{ s.t. } \begin{cases} \boldsymbol{o} \leq \boldsymbol{\alpha} \leq C\boldsymbol{e}, \\ \boldsymbol{B}_e \boldsymbol{\alpha} = 0, \end{cases} \tag{3}$$

where $\boldsymbol{e} = [1, 1, \dots, 1]^T$, $\boldsymbol{o} = [0,\ 0,\ \dots,\ 0]^T$, $\boldsymbol{X} = [\boldsymbol{x}_1,\ \boldsymbol{x}_2,\ \dots, \boldsymbol{x}_m]$, $\boldsymbol{y} = [y_1,\ y_2,\ \dots, y_m]^T$, $\boldsymbol{Y} = diag(\boldsymbol{y})$, $\boldsymbol{B}_e = \left[\boldsymbol{y}^T\right]$; $\boldsymbol{K} \in \mathbb{R}^{m \times m}$ is Symmetric Positive Semi-definite (SPS) matrix such that $\boldsymbol{K} := \boldsymbol{X}^T \boldsymbol{X}$. The formulation (3) is called $\ell 1$-loss SVM.

Support Vector Machines — Soft-margin

Further, we introduce dual to primal reconstruction formulas for the normal vector

$$\boldsymbol{w} = \boldsymbol{X}\boldsymbol{Y}\alpha, \tag{4}$$

and the bias

$$b = \frac{1}{\mathrm{card}(J)} \left(\boldsymbol{X}_{*J}^T \boldsymbol{w} - \boldsymbol{y}_J \right) \boldsymbol{e}_J^T, \tag{5}$$

where $J = \{i \mid 0 < \alpha_i < C, \ i = 1, 2, \ \dots, k\}$ is the support vector index set, $\mathrm{card}(J)$ presents its cardinality, $\boldsymbol{X}_{*J}$ denotes the submatrix of the matrix $\boldsymbol{X}$ with the column indices belonging to J; $\boldsymbol{y}_J$ and $\boldsymbol{e}_J$ are subvectors of the vectors $\boldsymbol{y}$ and $\boldsymbol{e}$, respectively. Using the reconstructed normal vector $\boldsymbol{w}$ and bias b, we set the decision rule up so that

$$\mathrm{sgn}\left(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b\right) = \begin{cases} +1 \dots \boldsymbol{x}_i \text{ belongs to Class A,} \\ -1 \dots \boldsymbol{x}_i \text{ belongs to Class B.} \end{cases} \tag{6}$$

Support Vector Machines — Hessian regularization

Instead of linear sum of the loss functions ξ_i, let us substitute it by sum of squared loss functions in the objective such (3) results into following form

$$\underset{\boldsymbol{w},\ b,\ \xi_i}{\arg\min} \frac{1}{2}\langle \boldsymbol{w}, \boldsymbol{w} \rangle + \frac{C}{2}\sum_{i=1}^{m} \xi_i^2 \ \text{ s.t. } \begin{cases} y_i\left(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b\right) \geq 1 - \xi_i, \\ i \in \{1, 2, \ldots, m\}. \end{cases} \tag{7}$$

The formulation (7) is called a primal $\ell 2$-loss SVM. By exploiting this approach, we can observe the term that quantifies misclassification error

$$\sum_{i=1}^{m} \xi_i^2 \geq 0,$$

therefore we do not consider $\xi_i > 0$ as constraint.

Support Vector Machines — Hessian regularization

As for the $\ell 1$-loss SVM, we derive dual formulation using the Lagrange duality, and, evaluating the KKT conditions, the primal formulation (7) transforms into the dual formulation as follows

$$\arg\min_{\boldsymbol{\alpha}} \ \frac{1}{2}\boldsymbol{\alpha}^T \left(\boldsymbol{H} + C^{-1}\boldsymbol{I}\right)\boldsymbol{\alpha} - \boldsymbol{\alpha}^T\boldsymbol{e} \ \text{ s.t. } \begin{cases} \mathbf{o} \leq \boldsymbol{\alpha}, \\ \boldsymbol{B}_{\boldsymbol{e}}\boldsymbol{\alpha} = \mathbf{o}. \end{cases} \tag{8}$$

Since the Hessian is regularized by matrix $C^{-1}\boldsymbol{I}$, it becomes symmetric positive definite (SPD). Finally, we adapt the support vector index set J such that $J = \{i \mid 0 < \alpha_i, \ i = 1, 2, \ \ldots, k\}$ for the reconstruction formulas (4), (5).

Support Vector Machines – No-bias data classification

In the case of the no-bias classification, we do not consider bias b in a classification model.

We include it into the problem by means of augmenting the vector $\boldsymbol{w}$ and each sample $\boldsymbol{x}_i$ with an additional dimension such that

$$\widehat{\boldsymbol{w}} \leftarrow \begin{bmatrix} \boldsymbol{w} \\ B \end{bmatrix}, \quad \widehat{\boldsymbol{x}}_i \leftarrow \begin{bmatrix} \boldsymbol{x}_i \\ \beta \end{bmatrix},$$

where $B \in \mathbb{R}$, and $\beta \in \mathbb{R}^+$ is a user defined variable that (typically set to 1).

Support Vector Machines – No-bias data classification

Let $p \in \{1, 2\}$, then, using augmented samples $\widehat{\boldsymbol{x}}_i, \ i = 1, 2, \ldots, m$ and vector $\widehat{\boldsymbol{w}}$, we can modify the both primal SVM formulations, i.e. (2) and (7), into the problem of finding hyperplane $\widehat{H} := \langle \widehat{\boldsymbol{w}}, \widehat{\boldsymbol{x}} \rangle$ as follows

$$\underset{\widehat{\boldsymbol{w}},\, \widehat{\xi}_i}{\arg\min} \ \frac{1}{2}\langle \widehat{\boldsymbol{w}}, \widehat{\boldsymbol{w}} \rangle + \frac{C}{p}\sum_{i=1}^{m} \widehat{\xi}_i^p \ \text{ s.t. } \begin{cases} y_i \langle \widehat{\boldsymbol{w}}, \widehat{\boldsymbol{x}}_i \rangle \geq 1 - \widehat{\xi}_i, \\ \widehat{\xi}_i \geq 0 \ \text{ if } p = 1, \ i \in \{1, 2, \ldots, m\}, \end{cases} \tag{9}$$

where $\widehat{\xi}_i = \max\left(0, 1 - y_i \langle \widehat{\boldsymbol{w}}, \widehat{\boldsymbol{x}}_i \rangle\right)$ is the hinge loss function releated to augmented samples $\widehat{\boldsymbol{x}}_i$.

Support Vector Machines — Model calibration

Platt proposed approximating a posterior probability by a parametric form of a sigmoid function such that

$$P(y = 1 \mid \boldsymbol{x}) \approx P_{A,B}(y = 1 \mid \boldsymbol{x}) = \frac{1}{1 + \exp\left(Af(\boldsymbol{x}) + B\right)}, \tag{10}$$

where parameters A, B are fitted using maximum likehood estimation.

The model (10) assumes the raw SVM output $f(\boldsymbol{x}) := H(\boldsymbol{x}) = \langle \boldsymbol{w}, \boldsymbol{x} \rangle + b$.

In order to no-bias classification, we define $\widehat{f}(\widehat{\boldsymbol{x}}) := \widehat{H}(\widehat{\boldsymbol{x}}) = \langle \widehat{\boldsymbol{w}}, \widehat{\boldsymbol{x}} \rangle$.

To avoid model overfitting, Platt suggested to use a new training set, i.e. calibration set, for training a calibrated model.

Let us denote calibration dataset as an ordered set as follows

$$CA := \{(f_1, y_1), (f_2, y_2), \dots (f_l, y_l)\},$$

where l is a number of the calibration samples, f_j is estimate of $f(\boldsymbol{x}_j)$ or $\widehat{f}(\widehat{\boldsymbol{x}}_{\boldsymbol{j}})$, $j \in \{1, 2, \dots, l\}$.

Additionally, Platt proposed transformation of binary labels y_j to target probabilities t_j such that $t_j = \frac{N_p+1}{N_p+2}$ iff $y = +1$, or $t_j = \frac{1}{N_n+2}$ iff $y = -1$, where N_p and N_n are numbers of positive and negative calibration samples, respectively.

Support Vector Machines — Model calibration

The best parameter setting, i.e. A^* and B^*, is determined by minimizing cross-entropy so that

$$\arg\min_{A,B} - \sum_{j=1}^{l} \left[\ t_j \log\left(p_j\right) + \left(1 - t_j\right) \log\left(1 - p_j\right)\ \right], \tag{11}$$

where $p_j = \frac{1}{1+\exp\left(Af_j+B\right)}$. To solve (11), Hsuan-Tien Lin et. all propose the Newton method.

PermonSVM — Features

Features:

- Uses PETSc and PermonQP
- Bias and no-bias formulations
- User defined penalty for unbalanced datasets ($C+$ and $C-$)
- Cross validation:
  - k-fold
  - Stratified k-fold
- Grid search
- Model score (accuracy, sensitivity, specifity, F1, MCC)
- Datasets - training, test, calibration
- Parallel IO (LIBSVM, HDF5, PETSc binary)

PermonSVM — Calling API

```
MPI_comm    comm = PETSC_COMM_WORLD;
SVM         svm;
PetscViewer viewer;

Mat         Xt_test,Y_test,Y_pred;

char    file_training[PETSC_MAX_PATH_LEN] = "examples/heart_scale.tr.h5";
char    file_test[PETSC_MAX_PATH_LEN] = "examples/heart_scale.te.h5";
char    file_calibration[PETSC_MAX_PATH_LEN] = "examples/heart_scale.ca.h5";

TRY( SVMCreate(comm,&svm) );
TRY( SVMSetType(svm,SVMPC) );
TRY( SVMSetFromOptions(svm) );
TRY( PetscViewerHDF5Open(comm,file_training,FILE_MODE_READ,&viewer) );
TRY( PetscViewerHDF5SetAIJNames(viewer,"i","j","a","ncols") );
TRY( SVMLoadTrainingDataset(svm,viewer) );
TRY( PetscViewerDestroy(&viewer) );

TRY( PetscViewerHDF5Open(comm,file_test,FILE_MODE_READ,&viewer) );
TRY( PetscViewerHDF5SetAIJNames(viewer,"i","j","a","ncols") );
TRY( SVMLoadTestDataset(svm,viewer) );
TRY( PetscViewerDestroy(&viewer) );

TRY( PetscViewerHDF5Open(comm,file_calibration,FILE_MODE_READ,&viewer) );
TRY( PetscViewerHDF5SetAIJNames(viewer,"i","j","a","ncols") );
TRY( SVMLoadCalibDataset(svm,viewer) );
TRY( PetscViewerDestroy(&viewer) );

TRY( SVMSetHyperOpt(svm,PETSC_TRUE) );
TRY( SVMTrain(svm) );
TRY( SVMTest(svm) );
```

Benchmarks – Balancing predictive relevance

We demonstrate technique of calibrating models related to Active-vs-Inactive no-bias classification on 3 targets.

Target (dataset)	#ligands (QSARs)	#active+	#inactive-
abl1 (training)	640	312	328
abl1 (calibration)	200	92	108
abl1 (test)	160	81	79
adora2a (training)	640	343	297
adora2a (calibration)	200	105	95
adora2a (test)	160	95	65
cnr1 (training)	640	392	248
cnr1 (calibration)	200	123	77
cnr1 (test)	160	110	50

Benchmarks – Balancing predictive relevance

For training (uncalibrated) classification models, we choose the best penalty C_{BE} from the set $\widehat{C} = \{2^p, p \in \{-7, -6, \ldots, 6, 7\}\}$ algorithmically employing the HyperOpt by means of grid-search combined with stratified 3-fold CV.

The relative norm of projected gradient being smaller than $1e-1$ is used as stopping criterion for the MPRGP (Modified Proportioning and Reduced Gradient Projection) algorithm in all presented experiments. The expansion step-size is fixed and determined such as $\alpha = 2.0/\|\boldsymbol{H}\|_2$, where $\|\boldsymbol{H}\|_2 = \sqrt{\lambda_{max}(\boldsymbol{H}^T\boldsymbol{H})}$.

Using PETSc implementation of the Newton method, the S-shaped calibration function is computed by minimizing cross-entropy of calibration data.

We use a deterministic approach instead of stochastic optimization.

Benchmarks – Balancing predictive relevance

Table 1: abl1, adora2a, cnr1 targets: evaluation of performance scores associated with uncalibrated models with C_{BE} and calibrated models with optimal threshold (thr.) in a sense of labels (binary classification) on test datasets.

Target	Loss	Uncalibrated model				Calibrated model			
		C_{BE}	Pre. [%]	Sen. [%]	AUC	Thr.	Pre. [%]	Sen. [%]	AUC
abl1	$\ell 1$	2^{-6}	71.60	65.17	0.66	0.52	65.43	64.63	0.64
	$\ell 2$	2^{-5}	69.14	60.22	0.64	0.54	60.49	60.49	0.60
adora2a	$\ell 1$	2^{-6}	70.53	82.72	0.74	0.41	78.95	78.95	0.74
	$\ell 2$	2^{-7}	70.53	83.75	0.74	0.44	78.95	78.95	0.74
cnr1	$\ell 1$	2^{-6}	90.00	82.50	0.77	0.63	83.64	83.64	0.74
	$\ell 2$	2^{-6}	87.27	81.36	0.74	0.58	83.64	83.64	0.74
cnr2	$\ell 1$	2^{-6}	83.93	82.46	0.83	0.53	83.04	83.04	0.72
	$\ell 2$	2^{-5}	86.61	82.91	0.85	0.53	83.93	83.93	0.73

Benchmarks – Balancing predictive relevance

Table 2: abl1, adora2a, cnr1, cnr2 calibrated single-target models: comparing quality of models in probabilistic sense (Brier score).

Target	Loss	Brier score
abl1	$\ell 1$	0.1105
	$\ell 2$	0.0947
adora2a	$\ell 1$	0.1280
	$\ell 2$	0.1222
cnr1	$\ell 1$	0.0905
	$\ell 2$	0.0710
cnr2	$\ell 1$	0.0889
	$\ell 2$	0.0611

The models trained using $\ell 2$-loss seem to be better calibrated by comparing Brier scores for all cases than ones related to the $\ell 1$-loss SVM. This could a consequence of underlying model robustness.

Benchmarks – Balancing predictive relevance

Table 3: abl1, adora2a, cnr1, cnr2 biological targets: elapsed time related to training of models including HyperOpt and calibration.

Loss	Elapsed time [s] (HyperOpt + Training + Calibration)			
	abl1	adora2a	cnr1	cnr2
$\ell 1$	2.15	2.61	1.95	2.57
$\ell 2$	1.38	1.86	1.57	1.58

We can observe speedups 1.56 (abl1), 1.40 (adora2a), 1.24 (cnr2), and 1.62 (cnr1) in order to using the $\ell 2$-loss SVM against the $\ell 1$-loss SVM.

Conclusions

- Advantage of SVMs: finding a learning function maximizing geometric margin.
- Disadvantages of SVMs: sensitivity to imbalanced datasets, outliers and multicollinearies among training samples, which could be a cause of preferencing one group over another.
- Additional calibrationing a model is required – Platt's Calibration was tested.
- To obtain better calibrated model (Brier score), it seems it is better to train model using $\ell 2$-loss SVM.

- Testing approach on large-scale dataset.
- Comparing Platt's scaling with isotonic regression.

Thank you for your kind patience and attention. Any questions?