Fast optimization of Multithreshold Entropy Linear Classifier

Rafal Jozefowicz, Wojciech Czarnecki

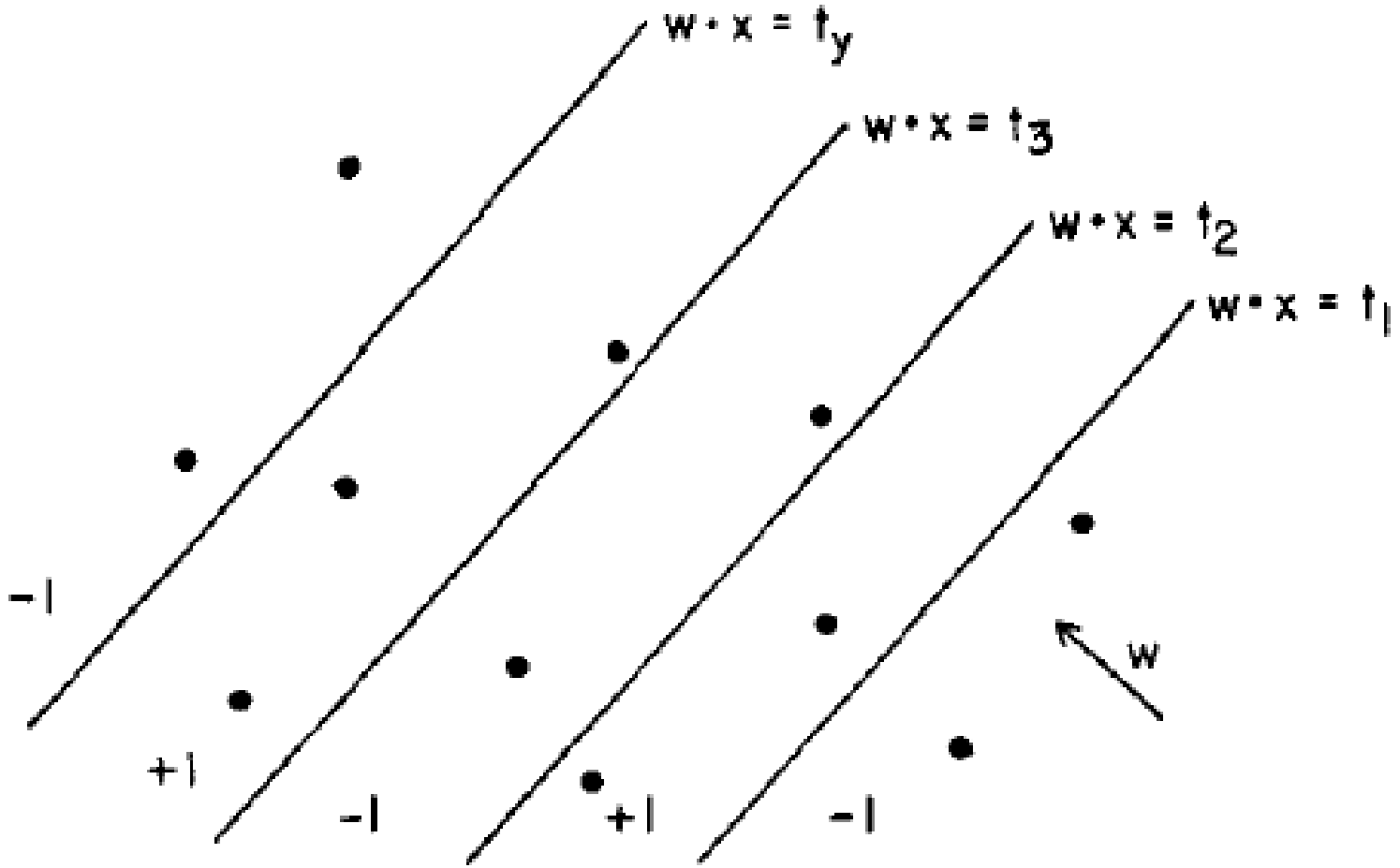
19 February 2015

Idea

Classification procedure:

- Find the best projection w for the training set
- Perform kernel density estimation for each class on the projected data
- Classify samples based on the KDE values

I will focus on the first part of the problem as the other two are not as computationally expensive.


w • x = t_y
w • x = t_3
w • x = t_2
w • x = t_1
-1
+1
-1
+1
-1
w

Find the best projection w for the training set

$$\begin{aligned} &\text{maximize}_{w \in \mathbb{R}^d} \;\; \mathrm{D}_{CS}([\![w^T X_-]\!], [\![w^T X_+]\!]) \\ &\text{subject to } \|w\| = 1 \end{aligned}$$

where $[\![X]\!]$ is the kernel density estimation of X.

Find the best projection w for the training set

$$\begin{aligned} &\text{maximize}_{w \in \mathbb{R}^d} \ \ \mathrm{D}_{CS}([\![w^T X_-]\!], [\![w^T X_+]\!]) \\ &\text{subject to } \|w\| = 1 \end{aligned}$$

where $[\![X]\!]$ is the kernel density estimation of X.

$$\begin{aligned} \mathrm{D}_{\mathrm{CS}}(f,g) = & -2\log\int fg(x)dx + \log\int f^2(x)dx + \log\int g^2(x)dx \\ = & -2\log \mathrm{ip}^{\times}(f,g) + \log \mathrm{ip}^{\times}(f,f) + \log \mathrm{ip}^{\times}(g,g) \\ \mathrm{ip}^{\times}(w) = & \frac{1}{\sqrt{2\pi V(w)|X||Y|}} \sum_{x,y} \exp\left(-\frac{\langle w, x-y\rangle^2}{2V(w)}\right) \\ V(w) = & \ \gamma^2 (\frac{4}{3})^{\frac{1}{25}} (|X|^{-\frac{7}{5}} \sum_x \langle x - \hat{X}, w\rangle^2 + |Y|^{-\frac{7}{5}} \sum_y \langle y - \hat{Y}, w\rangle^2) \end{aligned}$$

where $\mathrm{ip}^{\times}(w) = \mathrm{ip}^{\times}([\![w^T X_-]\!], [\![w^T X_+]\!])$.

Computational complexity

Computing $\text{ip}^{\times}(w)$ is computationally expensive as it requires $O(|X||Y|)$ steps.

Approximations!

We investigated two approximations of $\mathrm{ip}^{\times}(\cdot)$:

- Sorting and Discarding - ignores pairs of points that are too far away from each other as they don't contribute much to the final value. The maximum distance is determined dynamically for each w by the approximation factor.
- Binning - performs binning of the projected points, so that those located near each other are approximated by their empirical mean. The number of bins is determined dynamically for each w by the approximation factor.

Both of them are parameterized by ϵ, which bounds the absolute error made by the algorithms vs $\mathrm{ip}^{\times}(\cdot)$.

Sorting and Discarding

$$\mathrm{ip}_{\mathrm{sort}}{}^{\times}(w,\epsilon) = \frac{1}{\sqrt{2\pi V(w)|X||Y|}} \sum_{x,y\,:\,|\mathbf{x}-\mathbf{y}|<\mathrm{width}(\epsilon)} \exp\left(-\frac{\langle w, x-y\rangle^2}{2V(w)}\right)$$

Theorem

Using adaptive sorting and discarding with distance threshold in each iteration of at least

$$\sqrt{\max\left\{0, -V(w)\ln\left(2\epsilon^2\pi V(w)\right)\right\}}$$

leads to the computation of the $ip^{\times}$ function with at most ϵ error.

Sorting and Discarding

$$\mathrm{ip}_{\mathrm{sort}}{}^{\times}(w,\epsilon) = \frac{1}{\sqrt{2\pi V(w)|X||Y|}} \sum_{x,y\,:\,|\mathbf{x}-\mathbf{y}|<\mathrm{width}(\epsilon)} \exp\left(-\frac{\langle w, x-y\rangle^2}{2V(w)}\right)$$

The algorithm works best for sparse projections. It can be easily implemented in $O(n \log n + |\{x, y : |x - y| < \mathrm{width}(\epsilon)\}|)$ time and linear memory.

Binning

$$\mathrm{ip}_{\mathrm{bin}}{}^{\times}(w,\epsilon) = \frac{1}{\sqrt{2\pi V(w)|X||Y|}} \sum_{x,y} \exp\left(-\frac{(\langle w,x\rangle_b - \langle w,y\rangle_b)^2}{2V(w)}\right)$$

Theorem

Using adaptive binning technique with bin width in each iteration at most

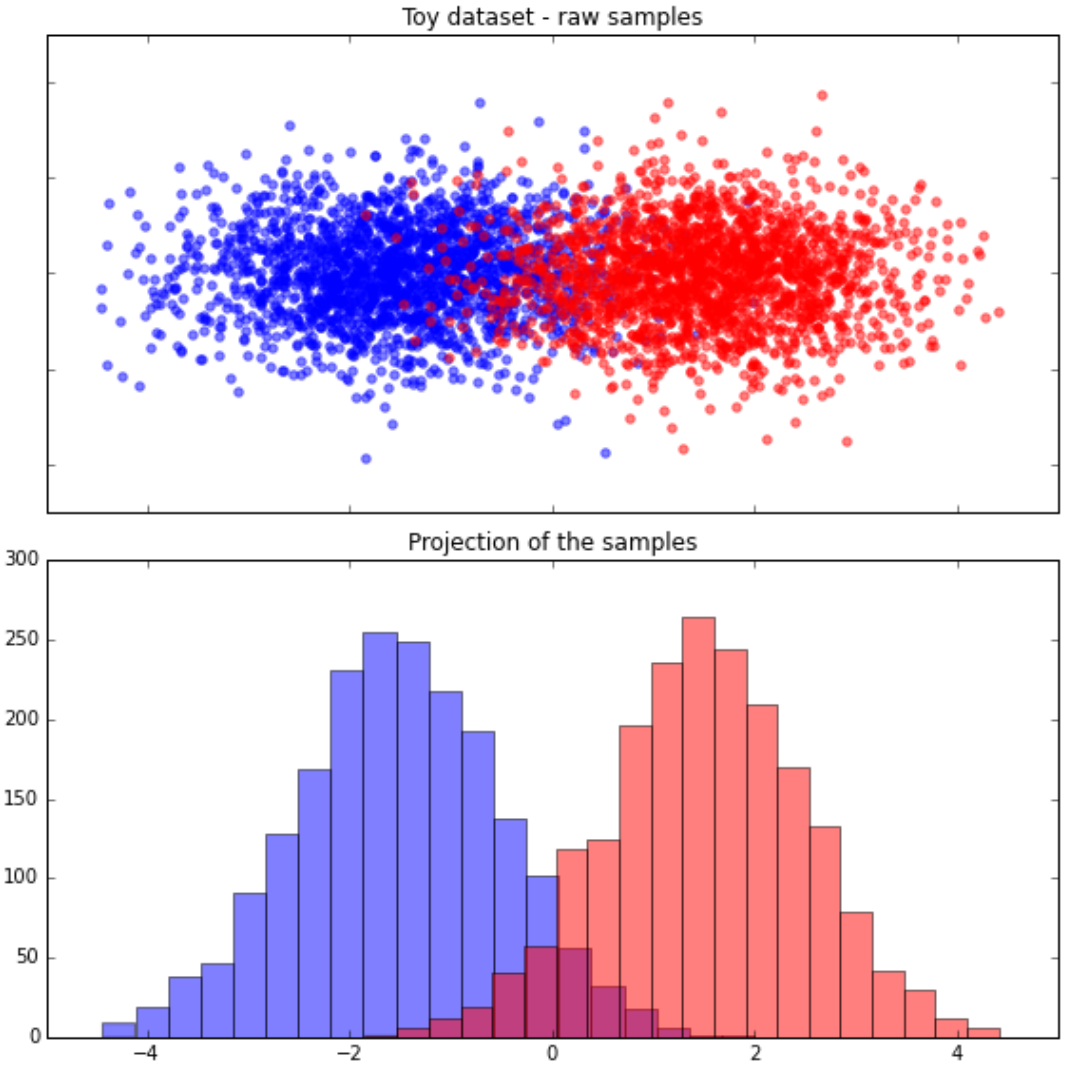
$$\sqrt{-2V(w)\ln\left(\max\left\{0, 1-\epsilon\sqrt{2\pi V(w)}\right\}\right)}$$

leads to the computation of the $ip^{\times}$ function with at most ϵ error.

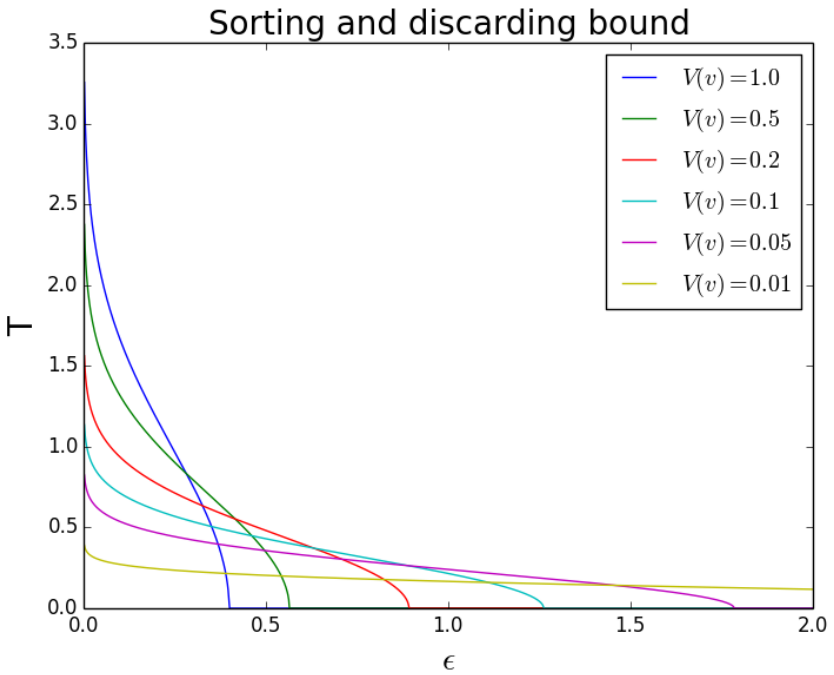
Binning

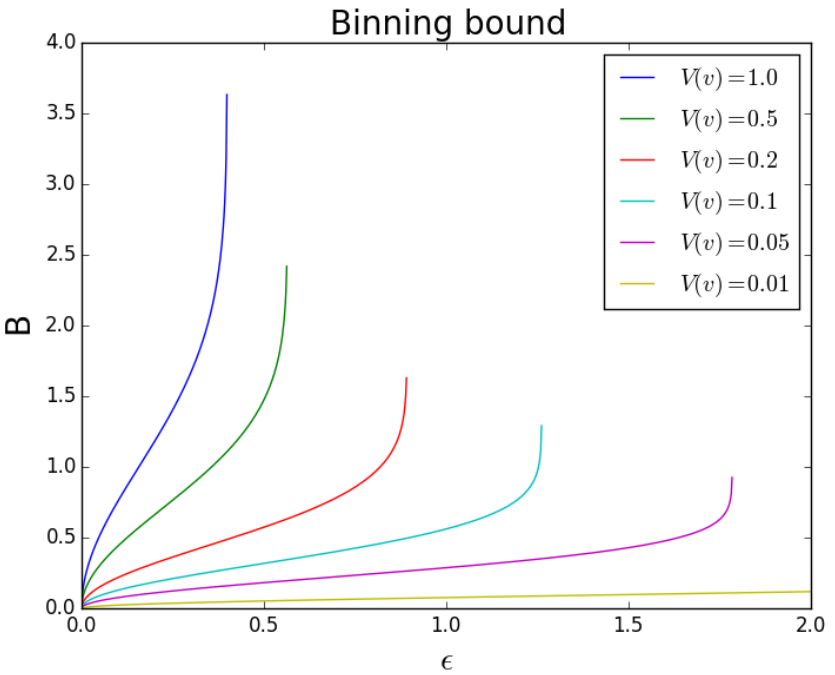
$$\mathrm{ip}_{\mathrm{bin}}{}^{\times}(w, \epsilon) = \frac{1}{\sqrt{2\pi V(w)|X||Y|}} \sum_{x,y} \exp\left(-\frac{(\langle w, x\rangle_b - \langle w, y\rangle_b)^2}{2V(w)}\right)$$

The algorithm works best for dense projections. The time complexity is quadratic in the number of bins.


Toy dataset - raw samples
Projection of the samples
300
250
200
150
100
50
0
−4
−2
0
2
4

Below are plots of the discarding threshold and bin width as the functions of the acceptable error ϵ.





Optimization on the sphere

For better numerical stability, the optimization should be performed on the sphere. By adding a custom regularization term we "guide" the optimization method to stay close to it.

Theorem

Given arbitrary sets $X_-, X_+ \subset \mathbb{R}^d$ *and corresponding* $D_{CS}(w) = D_{CS}([\![w^T X_-]\!], [\![w^T X_+]\!])$ *function we have:*

$$d := \max_{\|w\|=1} D_{CS}(w) = \max_{w} D_{CS}(w) - (\|w\|^2 - 1)^2$$

and

$$\{w : \|w\| = 1 \wedge D_{CS}(w) = d\} = \{w : D_{CS}(w) - (\|w\|^2 - 1)^2 = d\}.$$

Optimization on the sphere

The new objective function allows us to use optimization techniques that are not designed to work on the sphere. In particular, we used L-BFGS and Conjugate Gradients. We don't need any additional hyperparameters to be fitted.

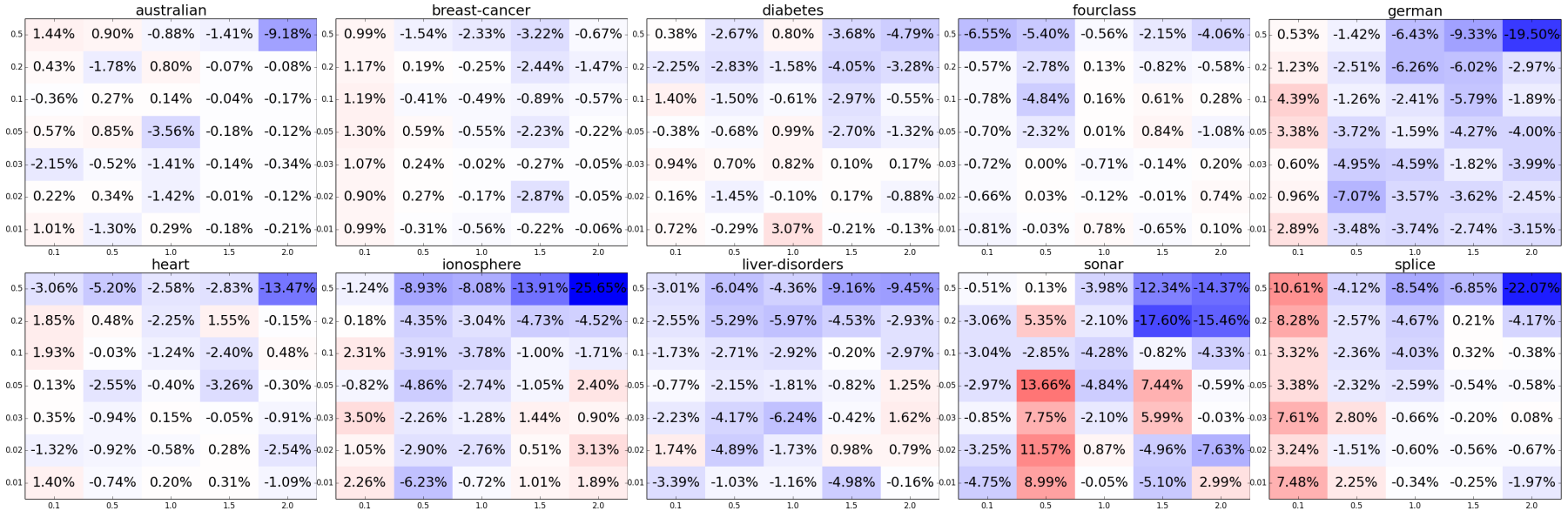
We evaluated proposed approximations on 10 datasets from UCI and libSVM's repositories. Both D_{CS} and its approximations are coded in Python using numpy and scipy.

Mean ratio of exp calls between approximated technique and original method during optimizations.

method	CG		L-BFGS-B	
name	bin	dist	bin	dist
australian	0.11	0.44	0.11	0.45
breast-cancer	0.10	0.46	0.10	0.46
diabetes	0.21	0.56	0.22	0.54
fourclass	0.19	0.51	0.19	0.49
german.numer	0.15	0.47	0.19	0.46
heart	0.29	0.47	0.26	0.47
ionosphere	0.25	0.55	0.24	0.54
liver-disorders	0.29	0.65	0.31	0.67
sonar	0.32	0.53	0.29	0.50
splice	0.19	0.44	0.16	0.43

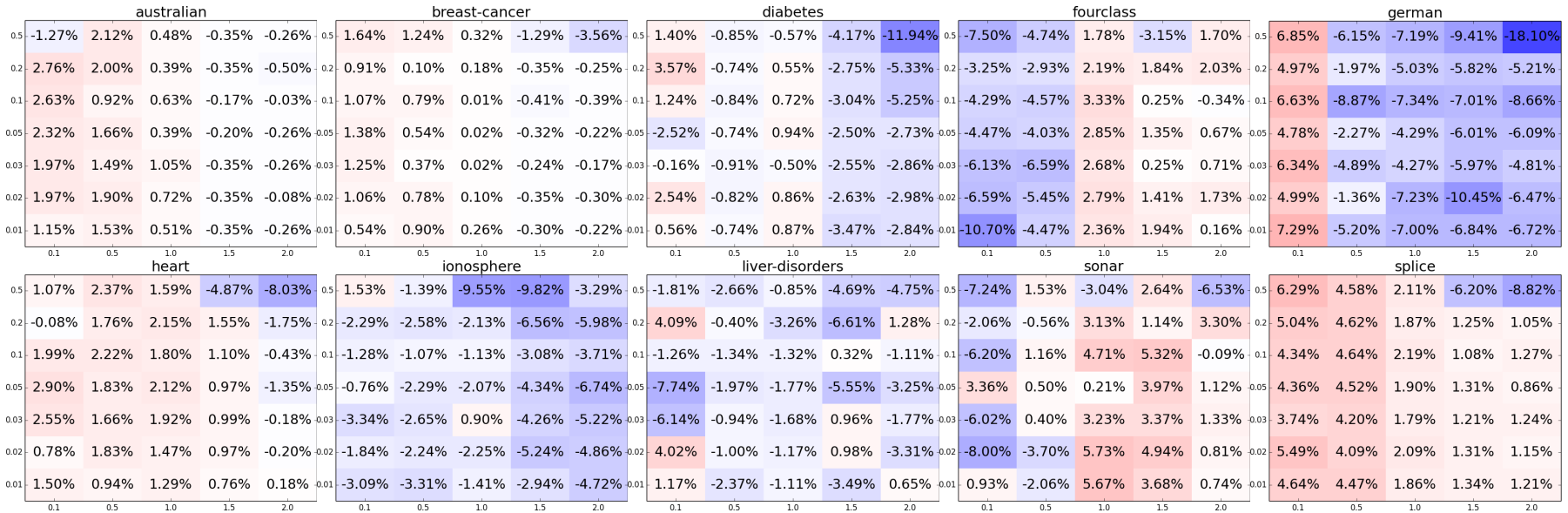
Sorting and Discarding - Results

Comparison of the cross validation BAC scores between given γ hyperparameter of D_{CS} (x-axis), accepted error ϵ (y-axis). Positive values (and corresponding red colors) represent decrease in BAC score while negative values and corresponding blue colors – increase after using approximated method.



Binning - Results

Comparison of the cross validation BAC scores between given γ hyperparameter of D_{CS} (x-axis), accepted error ϵ (y-axis). Positive values (and corresponding red colors) represent decrease in BAC score while negative values and corresponding blue colors – increase after using approximated method.



Number of optimization methods' iterations.

method name	CG bin	CG D_{CS}	CG dist	L-BFGS-B bin	L-BFGS-B D_{CS}	L-BFGS-B dist
australian	4	36	22	11	39	37
breast-cancer	4	35	8	6	39	14
diabetes	3	30	20	18	36	29
fourclass	4	12	10	6	15	14
german.numer	7	60	32	7	58	38
heart	3	40	19	12	34	20
ionosphere	5	600	216	18	384	152
liver-disorders	4	30	22	22	43	30
sonar	4	262	115	15	139	100
splice	4	92	26	14	65	41

Number of optimization methods' iterations.

It's important to note that the number of iterations is not equal to the total number of evaluations of the function. It suggests, though, that when using approximations, the problem becomes simpler than the baseline.

Conclusions

1. We proposed two simple approximation schemes for faster computation of MELC objective function and its gradient.
2. We showed how to efficiently use existing off-the-shelf optimizers by a simple change of the objective function while at the same time still work near the unit sphere. Without the regularization term, the norm of w tends to explode on some datasets and making it numerically unstable.
3. Both algorithms significantly reduce the mean number of exp calls while not sacrificing the resulting accuracy.
4. Our experiments suggest that the approximations act like some kind of regularization of the classifier.

Combining methods

The algorithms complement each other and it is easy to determine, which will be faster for a given projection. This can further improve experimental results and make it more applicable for larger datasets.