DOI:10.1587/transinf.2024EDL8040

Publicized:2024/08/08

LETTER

Mixup SVM Learning for Compound Toxicity Prediction Using Human Pluripotent Stem Cells

Rikuto MOCHIDA[†], **Miya NAKAJIMA**[†], **Haruki ONO**[†], **Takahiro ANDO**[†], ***Nonmembers,***
and **Tsuyoshi KATO**[†], ***Member***

SUMMARY Drug discovery, characterized by its time-consuming and costly nature, demands approximately 9 to 17 years and around two billion dollars for development. Despite the extensive investment, about 90% of drugs entering clinical trials face withdrawal, with compound toxicity accounting for 30% of these instances. Ethical concerns and the discrepancy in mechanisms between humans and animals have prompted regulatory restrictions on traditional animal-based toxicity prediction methods. In response, human pluripotent stem cell-based approaches have emerged as an alternative. This paper investigates the scalability challenges inherent in utilizing pluripotent stem cells due to the costly nature of RNAseq and the lack of standardized protocols. To address these challenges, we propose applying Mixup data augmentation, a successful technique in deep learning, to kernel SVM, facilitated by Stochastic Dual Coordinate Ascent (SDCA). Our novel approach, Exact SDCA, leverages intermediate class labels generated through Mixup, offering advancements in both efficiency and effectiveness over conventional methods. Numerical experiments reveal that Exact SDCA outperforms Approximate SDCA and SGD in attaining optimal solutions with significantly fewer epochs. Real data experiments further demonstrate the efficacy of multiplexing gene networks and applying Mixup in toxicity prediction using pluripotent stem cells.

key words: *compound toxicity prediction, pluripotent stem cell, kernel SVM, optimization.*

1. Introduction

Drug discovery is one of the most time-consuming and costly fields among all research and development areas. Developing a drug requires approximately 9 to 17 years and costs around two billion dollars [1]–[3]. Moreover, about 90% of drugs that reach clinical trials are forced to withdraw [4]. Thirty percent of these withdrawals are due to compound toxicity. To prevent withdrawal, it is crucial to predict toxicity and screen it before clinical trials.

Traditionally, compound toxicity prediction relied mainly on animal experiments. However, in recent years, many countries have planned or implemented regulations or bans due to ethical concerns [5]. Additionally, differences in the mechanisms of action between humans and animals mean that compounds approved through animal experiments may still pose toxicity risks to humans [6].

As an alternative to animal experiments, toxicity prediction using human pluripotent stem cells has garnered attention. Specifically, exposing compounds to pluripotent stem cells such as iPS cells or ES cells and predicting toxicity through changes in gene expression levels via machine learning models. Yamane et al. demonstrated an approach to predict toxicity by constructing gene networks from changes in gene expression, thus highlighting the usefulness of human pluripotent stem cells [7].

However, methods using pluripotent stem cells suffer from the drawback of difficulty in scaling up data. Currently, RNAseq, which is used to obtain expression data, costs expensively. Measuring gene expression levels for each compound and over time, and conducting repeat experiments, would incur significant costs and time. Furthermore, as the protocols for obtaining gene expression data are not fully standardized, integrating expression data measured by different research organizations into training data is not straightforward. The barriers to scaling up data prevent large capacity deep-learning models from good generalization performance. Even with linear models or kernel methods that do not require deep learning, it is still challenging to reach a sufficient data size.

This paper demonstrates that the predictive performance can be improved by applying Mixup data augmentation [8], which has been successful in deep learning, to kernel SVM. Mixup is a method of augmenting data by taking convex combinations of training examples, including class labels. In this study, Stochastic Dual Coordinate Ascent (*SDCA*) [9] was applied for the learning of the predictor. Through this research, the following achievements were obtained:

- We developed **Exact SDCA**, which learns on the reproducing kernel Hilbert space using intermediate class labels generated by applying Mixup. To our knowledge, the learning algorithm for kernel methods on intermediate class labels has not been thoroughly analyzed. Applying SDCA eliminates the need for hyperparameters for optimization, which is more convenient than **SGD** (Stochastic Gradient Descent) requiring interactive optimization.
- We discovered that each iteration of Exact SDCA for learning the kernel SVM can be executed with the same computational cost as when intermediate class labels are not used.
- Through numerical experiments, we confirmed that Exact SDCA developed in this study reaches the optimal solution with significantly fewer epochs compared to **Approximate SDCA** and SGD.

[†]The authors are with the Faculty of Informatics, Gunma University, 4–2 Aramakicho, Maebashi City, Gunma Prefecture, 371–8510, Japan

- We demonstrated through experiments using real data that multiplexing gene networks and applying Mixup are effective in toxicity prediction using pluripotent stem cells.

2. Learning Problem

The Mixup method generates new examples from a training dataset by randomly selecting two examples $(\boldsymbol{x}_i^{\mathrm{o}}, y_i^{\mathrm{o}}), (\boldsymbol{x}_j^{\mathrm{o}}, y_j^{\mathrm{o}}) \in \mathbb{R}^d \times \{\pm 1\}$ and using a value η generated from a beta distribution in the interval $[0, 1]$. The new examples are generated as follows:

$$\begin{aligned} \boldsymbol{x}_{\mathrm{new}} &= (1-\eta)\boldsymbol{x}_i^{\mathrm{o}} + \eta \boldsymbol{x}_j^{\mathrm{o}}, \\ y_{\mathrm{new}} &= (1-\eta) y_i^{\mathrm{o}} + \eta y_j^{\mathrm{o}}. \end{aligned} \tag{1}$$

By repeating this process, data augmentation is achieved. It should be noted that the class labels y_{new} of newly generated compounds are not binary but intermediate values between $+1$ and -1.

The data augmentation targeted in this study is not limited to Mixup alone. Its variant, GenLabel [10], also generates intermediate class labels from discrete binary class labels, similar to Mixup. The optimization algorithm developed in this study can be applied to a wide range of data augmentation that generates continuous class labels $y_{\mathrm{new}} \in [-1, +1]$ from discrete class labels $y_i^o \in \{\pm 1\}$.

We wish to train a predictor $f : \mathbb{R}^d \to \mathbb{R}$ from the dataset with continuous class labels

$$(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n) \in \mathbb{R}^d \times [-1, +1] \tag{2}$$

generated with some data augmentation method. To train the predictor f from such a dataset, we use the *mixup loss* function $\phi_{\mathrm{mup}}(\cdot\,; y) : \mathbb{R} \to \mathbb{R}$ defined by the following equation:

$$\phi_{\mathrm{mup}}(s\,; y) := \frac{1+y}{2}\phi(s) + \frac{1-y}{2}\phi(-s). \tag{3}$$

Here, $\phi : \mathbb{R} \to \mathbb{R}$ is a loss function that does not use intermediate class labels. SVM uses the smooth hinge loss function as the loss function:

$$\phi_{\mathrm{smh}}(s) := \begin{cases} -s + 1 - \frac{\gamma_{\mathrm{sm}}}{2} & \text{if } s < 1 - \gamma_{\mathrm{sm}}, \\ \frac{1}{2\gamma_{\mathrm{sm}}}(s-1)^2 & \text{if } 1 - \gamma_{\mathrm{sm}} \le s < 1, \\ 0 & \text{if } 1 \le s. \end{cases} \tag{4}$$

where $\gamma_{\mathrm{sm}} \in (0, 1)$ is a constant. The main contribution of this study is a theoretical finding when employing the mixup loss function defined with $\phi = \phi_{\mathrm{smh}}$ in (3). Kernel SVM learning problem is to find the optimal predictor f that minimizes the regularized empirical risk $R[f]$ from the reproducing kernel Hilbert space $\mathcal{H}_\kappa$ constructed from a positive definite kernel function $\kappa : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ (e.g., RBF kernel, polynomial kernel). This involves solving the following minimization problem:

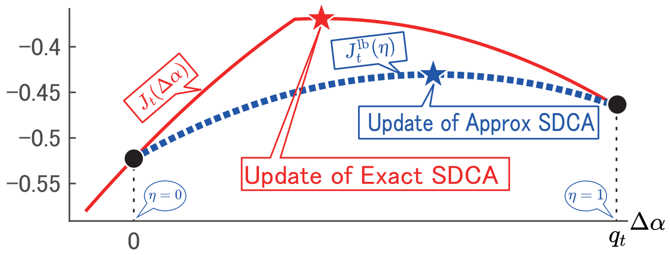

Fig. 1 Exact SDCA and approximate SDCA. Exact SDCA updates the dual variables to maximize $J_t(\Delta\alpha_{i_t})$, whereas Approximate SDCA abbreviated to Approx SDCA maximizes the parabola $J_t^{\mathrm{lb}}(\eta)$.

Algorithm 1: Exact SDCA for maximizing $D(\alpha)$.

```
1 begin
2     Choose $\alpha^{(0)}$ s.t. $\alpha^{(0)} \in \mathrm{dom}(-D)$;
3     for $t := 1$ to $T$ do
4         Pick $i$ randomly from $\{1, \dots, n\}$;
5         $\Delta\alpha_t := \operatorname{argmax}_{\Delta\alpha \in \mathbb{R}} J_t(\Delta\alpha)$;
6         $\alpha^{(t)} := \alpha^{(t-1)} + \Delta\alpha_t \boldsymbol{e}_{i_t}$;
7     end
8 end
```

$$\begin{aligned} &\min \quad R[f] \quad \text{wrt} \quad f \in \mathcal{H}_\kappa, \\ &\text{where} \quad R[f] := \frac{\lambda}{2}\|f\|_{\mathcal{H}_\kappa}^2 + \frac{1}{n}\sum_{i=1}^n \phi_{\mathrm{mup}}(f(\boldsymbol{x}_i)\,; y_i), \end{aligned} \tag{5}$$

where $\lambda > 0$ is the regularization constant.

3. Optimization Algorithms

In this study, we apply SDCA to minimize the regularized empirical risk $R[f]$. SDCA maximizes the dual function $D : \mathbb{R}^n \to \mathbb{R}$ instead of directly minimizing $R[f]$:

$$D(\boldsymbol{\alpha}) := -\frac{\lambda}{2}\|f_\alpha\|_{\mathcal{H}_\kappa}^2 - \frac{1}{n}\sum_{i=1}^n \phi_{\mathrm{mup}}^*(-\alpha_i\,; y_i), \tag{6}$$

where $\phi_{\mathrm{mup}}^*(\cdot\,; y_i) : \mathbb{R} \to \mathbb{R} \cup \{+\infty\}$ represents the convex conjugate of $\phi_{\mathrm{mup}}(\cdot\,; y_i)$; we have defined

$$f_\alpha := \frac{1}{\lambda n}\sum_{i=1}^n \alpha_i \kappa(\mathbf{x}_i, \cdot) \in \mathcal{H}_\kappa. \tag{7}$$

If $\boldsymbol{\alpha}_\star \in \mathbb{R}^n$ is the solution that maximizes $D(\boldsymbol{\alpha})$, then the solution that minimizes $R[f]$ is expressed as $f_\star := f_{\alpha_\star}$.

Algorithm 1 describes the framework of SDCA. An example $i_t \in [n]$ is selected at random at each iteration t and updates the corresponding dual variable by $\alpha_{i_t}^{(t+1)} := \alpha_{i_t}^{(t)} + \Delta\alpha$, where $\boldsymbol{\alpha}^{(t)} := \left[\alpha_1^{(t)}, \dots, \alpha_n^{(t)}\right]^\top$ is the value of the dual variable vector at iteration $t-1$, and the rest of $(n-1)$ entries are fixed. Equivallently, the update rule can be rewritten as

$$\boldsymbol{\alpha}^{(t)} := \boldsymbol{\alpha}^{(t-1)} + \Delta\alpha \boldsymbol{e}_{i_t}. \tag{8}$$

Ideally, the value of $\Delta\alpha$ should be determined to maximize

$$J_t(\Delta\alpha) := D(\boldsymbol{\alpha}^{(t-1)} + \Delta\alpha \boldsymbol{e}_{i_t}) - D(\boldsymbol{\alpha}^{(t-1)}) \tag{9}$$

where $\boldsymbol{e}_i$ is the unit vector with ith entry one. *Exact SDCA* has referred to the SDCA achieving this ideal update rule in each iteration. However, it is not simple to implement the Exact SDCA for general loss functions, because there exists no closed form solution for the ideal exact update rule, which can be seen from the rearrangement of the function J_t:

$$J_t(\Delta\alpha) = \frac{\lambda}{2}\left\|\boldsymbol{w}^{(t-1)}\right\|^2 - \frac{\lambda}{2}\left\|\boldsymbol{w}^{(t-1)} + \frac{\boldsymbol{x}_{i_t}\Delta\alpha}{\lambda n}\right\|^2 + \frac{1}{n}\phi^*_{\text{mup}}(-\alpha^{(t-1)}_{i_t}; y_{i_t}) - \frac{1}{n}\phi^*_{\text{mup}}(-\alpha^{(t-1)}_{i_t} - \Delta\alpha; y_{i_t}). \tag{10}$$

This implies that the intractability of maximization of $J_t(\Delta\alpha)$ because of the fact that the term $\Delta\alpha$ is hidden in the argument of the loss function $\phi^*_{\text{mup}}(\cdot\,; y_{i_t})$. In this study, we have discovered the following theorem:

Theorem 3.1: If employing the mixup loss function defined with $\phi = \phi_{\text{smh}}$ in (3), the function $J_t(\Delta\alpha)$ can be represented by a *concave piecewise quadratic function* with at most two intervals. Furthermore, it is possible to achieve each update of Exact SDCA with the computational cost $O(n)$ in the setting that the pre-computed kernel values $K_{i,j} := \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$ are available.

The proof is given in Appendix A. The next section shall report numerical examples illustrating Exact SDCA converges faster than the conventional SDCA referred to as *Approximate SDCA*. In what follows, how Approximate SDCA updates a dual variable at each iteration is described. Conventionally, when the function $J_t(\Delta\alpha)$ is not a single parabola, the function $J_t(\Delta\alpha)$ is approximated by a single parabola function $J^{\text{lb}}_t(\eta)$, which serves as the lower bound of $J_t(\eta q_t)$ [11]: $\forall\eta \in [0, 1]$,

$$J_t(\eta q_t) \geq -a_t\eta^2 + b_t\eta =: J^{\text{lb}}_t(\eta), \tag{11}$$

for some $a_t > 0$ and $b_t \in \mathbb{R}$, where $q_t := -\nabla\phi_{\text{mup}}(f_{\boldsymbol{\alpha}^{(t-1)}}(\boldsymbol{x}_{i_t}); y_{i_t}) - \alpha^{(t-1)}_i$. The values of the two coefficients a_t and b_t are found based on the smoothness of the loss function, say $1/\gamma_{\text{sm}}$. Instead of maximizing $J^{\text{lb}}_t(\Delta\alpha)$ directly, Approximate SDCA uses the maximizer of the lower-bound $J^{\text{lb}}_t(\eta)$ to update the dual variable by $\Delta\alpha := \eta_\star q_t$ at each iteration, where $\eta_\star \in \operatorname*{argmax}_{\eta\in[0,1]} J^{\text{lb}}_t(\eta)$.

As shown in Figure 1, in Approximate SDCA, updates are made to suboptimal positions. Inequality (11) ensures that the improvement of the exact update is always larger than (or equal to, in the worst case) that of the approximate update, leading to faster convergence to the optimum. Theorem 3.1 demonstrates that even when $J_t(\Delta\alpha)$ is not a single parabola, exact updates can be performed with the same computational cost as in the case of a parabola.

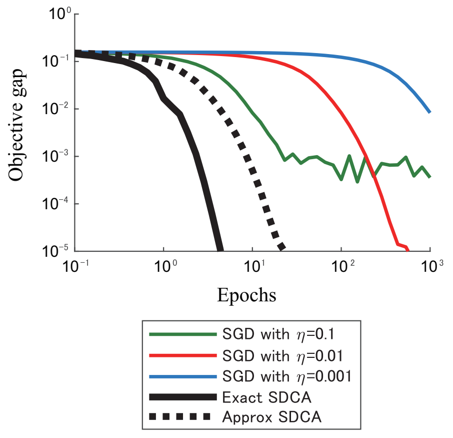

Fig. 2 Comparison of convergence speeds of Exact SDCA, Approximate SDCA, and SGD with different step sizes η.

Table 1 Comparison of toxicity prediction performance by F1 Score. In the table, NT and CT, respectively, denote tasks predicting the neurotoxicity and the cardiotoxicity.

Method	NT	CT
StemPanTox [7]	0.588	0.703
Multi-Net w/o Mixup	0.800	0.897
Multi-Net w Mixup	**0.838**	**0.906**

4. Experiments

4.1 Convergence Perforamances

We compared Exact SDCA, approximate SDCA, and SGD through numerical experiments. We set the number of examples to $n = 1{,}000$ and used $\lambda = 1/n$ as the regularization constant. Figure 2 plots the progress of learning with the horizontal axis representing the number of epochs and the vertical axis representing the objective gap $R[f] - R[f_\star]$. Exact SDCA attained $R[f] - R[f_\star] < 10^{-5}$ in only 5.34 epochs, while approximate SDCA required 43.29 epochs. SGD reached $R[f] - R[f_\star] < 10^{-5}$ within 1,000 epochs only when $\eta = 0.01$.

4.2 Toxicity Prediction Performances

We compared the conventional StemPanTox method [7], which uses a single gene network, with a method that multiplexes gene networks and further augments data with 100 additional examples using Mixup. We predicted the neurotoxicity and cardiotoxicity of 24 compounds used in [7] using a leave-one-out approach and evaluated them with F1 scores. Table 1 shows the results, where the method with multiple gene networks is referred to as *Multi-Net*. It was confirmed that by multiplexing gene networks and augmenting data with Mixup, the prediction performance was improved.

5. Conclusions

In this paper, we proposed a learning algorithm to predict toxicity using human pluripotent stem cells. Considering the difficulty of scaling gene expression data from human pluripotent stem cells, we explored Mixup data augmentation in SVM training. This resulted in training data containing class labels with intermediate values. When applying Exact SDCA to training data containing intermediate class labels, the objective function of the subproblem to be solved at each iteration does not generally become a parabola. Nevertheless, in this study, we discovered that Exact SDCA can be applied with the same computational cost as when not using intermediate class labels. Additionally, numerical experiments demonstrated that Exact SDCA developed in this study could reach the optimal solution in significantly fewer epochs compared to Approximate SDCA and SGD. Through computational experiments using real gene-expression data, we empirically verified that multiplexing gene networks and applying Mixup data augmentation are effective in predicting toxicity using human pluripotent stem cells.

Appendix A: Proof of Theorem 3.1

The function $J_t : \mathbb{R} \to \mathbb{R}$ can be rearranged as

$$J_t(\Delta\alpha) = -\frac{K_{i,i}}{2\lambda n^2}\Delta\alpha^2 - \frac{f_{\alpha^{(t-1)}}(\boldsymbol{x}_i)}{n}\Delta\alpha - \frac{1}{n}\phi^*_{\text{mup}}(-\alpha_i^{(t-1)} - \Delta\alpha\,;\, y_i) + \frac{1}{n}\phi^*_{\text{mup}}(-\alpha_i^{(t-1)}\,;\, y_i). \tag{A· 1}$$

where the convex conjugates of the loss functions are written as

$$\phi^*_{\text{mup}}(u\,;\,y_i) = \frac{\gamma_{\text{sm}}}{1-y_i}u^2 + \left(-1 + \frac{1+y_i}{1-y_i}\gamma_{\text{sm}}\right)u + \left(\frac{\gamma_{\text{sm}}}{2(1-y_i)} - 1\right)(1+y_i) \tag{A· 2}$$

if $-\frac{1}{2}(1+y_i) \le u \le -y_i$;

$$\phi^*_{\text{mup}}(u\,;\,y_i) = \frac{\gamma_{\text{sm}}}{1+y_i}u^2 + \left(+1 - \frac{1-y_i}{1+y_i}\gamma_{\text{sm}}\right)u + \left(\frac{\gamma_{\text{sm}}}{2(1+y_i)} - 1\right)(1-y_i) \tag{A· 3}$$

if $-y_i < u \le \frac{1}{2}(1-y_i)$; otherwise $\phi^*_{\text{mup}}(u\,;\,y_i) = +\infty$. Thus, the convex conjugate of each loss function is a convex piecewise quadratic function. Combining this fact with (A· 1) implies that J_t is a concave piecewise quadratic function. Maximization of $J_t(\Delta\alpha)$ can be done by finding $\nabla J_t(\Delta\alpha) = 0$. Computing the coefficients of $\Delta\alpha^2$ and $\Delta\alpha$ is necessary to find the optimal update. The most costly term contained in the coefficients is

$$f_{\alpha^{(t-1)}}(\boldsymbol{x}_i) = \frac{1}{\lambda n}\sum_{j=1}^{n} K_{i,j}\alpha_j^{(t-1)} \tag{A· 4}$$

which requires $O(n)$ computational time. *q.e.d.*

Acknowledgments

This research was supported by JSPS KAKENHI Grants 22K04372 and 23H03555.

References

[1] D.G. Brown, H.J. Wobst, A. Kapoor, L.A. Kenna, and N. Southall, “Clinical development times for innovative drugs,” Nature Reviews Drug Discovery, vol.21, no.11, pp.793–794, Nov. 2021. noi:10.1038/d41573-021-00190-9.

[2] M. Omejc, “Drug development: The journey of a medicine from lab to shelf,” J Develop Drugs, vol.9, no.1, p.e155, 2020. doi:10.35248/2329-6631.20.9.e155.

[3] C.L. Munro and R.H. Savel, “Narrowing the 17-year research to practice gap,” American Journal of Critical Care, vol.25, no.3, pp.194–196, May 2016. doi:10.4037/ajcc2016449.

[4] D. Sun, W. Gao, H. Hu, and S. Zhou, “Why 90% of clinical drug development fails and how to improve it?,” Acta Pharm Sin B, vol.12, no.7, pp.3049–3062, Jul 2022.

[5] D. Grimm, “EPA plan to end animal testing splits scientists,” Science, vol.365, no.6459, p.1231, Sept. 2019.

[6] P. Perel, I. Roberts, E. Sena, P. Wheble, C. Briscoe, P. Sandercock, M. Macleod, L.E. Mignini, P. Jayaram, and K.S. Khan, “Comparison of treatment effects between animal experiments and clinical trials: systematic review,” BMJ, vol.334, no.7586, p.197, Dec. 2006. doi:10.1136/bmj.39048.407928.be.

[7] J. Yamane, T. Wada, H. Otsuki, K. Inomata, M. Suzuki, T. Hisaki, S. Sekine, H. Kouzuki, K. Kobayashi, H. Sone, J.K. Yamashita, M. Osawa, M.K. Saito, and W. Fujibuchi, “StemPanTox: A fast and wide-target drug assessment system for tailor-made safety evaluations using personalized iPS cells,” iScience, vol.25, no.7, p.104538, July 2022. doi:10.1016/j.isci.2022.104538.

[8] H. Zhang, M. Cisse, Y.N. Dauphin, and D. Lopez-Paz, “mixup: Beyond empirical risk minimization,” International Conference on Learning Representations, 2018.

[9] S. Shalev-Shwartz and T. Zhang, “Stochastic dual coordinate ascent methods for regularized loss,” J. Mach. Learn. Res., vol.14, no.1, pp.567–599, Feb. 2013. `http://gofile.me/71IvF/EloMGHHTI`.

[10] J.Y. Sohn, L. Shang, H. Chen, J. Moon, D. Papailiopoulos, and K. Lee, “GenLabel: Mixup relabeling using generative models,” Proceedings of the 39th International Conference on Machine Learning, ed. K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Proceedings of Machine Learning Research, vol.162, pp.20278–20313, PMLR, 17–23 Jul 2022.

[11] Y. Takada, R. Mochida, M. Nakajima, S. suke Kadoya, D. Sano, and T. Kato, “Stochastic dual coordinate ascent for learning sign constrained linear predictors,” IEICE Transactions on Information & Systems, vol.-, no.-, pp.–, - -. accepted.