A New Generalized Error Path Algorithm for Model Selection

Bin Gu[1,2] JSGUBIN@NUIST.EDU.CN
Charles X. Ling[1] CLING@CSD.UWO.CA
[1]Department of Computer Science, University of Western Ontario, Canada
[2]School of Computer & Software, Nanjing University of Information Science & Technology, China

Abstract

Model selection with cross validation (CV) is very popular in machine learning. However, CV with grid and other common search strategies cannot guarantee to find the model with minimum CV error, which is often the ultimate goal of model selection. Recently, various solution path algorithms have been proposed for several important learning algorithms including support vector classification, Lasso, and so on. However, they still do not guarantee to find the model with minimum CV error. In this paper, we first show that the solution paths produced by various algorithms have the property of piecewise linearity. Then, we prove that a large class of error (or loss) functions are piecewise constant, linear, or quadratic w.r.t. the regularization parameter, based on the solution path. Finally, we propose a new generalized error path algorithm (GEP), and prove that it will find the model with minimum CV error for the entire range of the regularization parameter. The experimental results on a variety of datasets not only confirm our theoretical findings, but also show that the best model with our GEP has better generalization error on the test data, compared to the grid search, manual search, and random search.

1. Introduction

In machine learning, most of learning algorithms are parameterized (normally continuously). For example, support vector machines (SVMs) (Vapnik, 1998) have a regularization parameter controlling the trade-off between a large margin and a small error penalty. Lasso (Tibshirani, 1996) has a regularization parameter on model's L_1 penalty to lead to sparse solutions. It is obvious that one fundamental task of the parameterized learning algorithms is model selection: tuning the parameters of models to achieve optimal generalization performance.

Model selection with cross validation (CV) (Arlot et al., 2010) is very popular in machine learning. The main idea of CV is to divide data into two parts (once or several times): one (the training set) used to train a model and the other (the validation set) used to estimate the error of the model. CV selects the parameter among a group of candidates with the smallest CV error, where the CV error is the average of the multiple validation errors. Normally, K-fold, leave-one-out, or repeated random sub-sampling procedures were used for CV. For example, Jahrer & Töscher (2012) used 16-fold CV for collaborative filtering. Foster et al. (2014) used leave-one-out CV on SVM for medical diagnosis. Usai et al. (2009) used repeated random sub-sampling validation on Lasso for genomic selection. Izbicki (2013); Pahikkala et al. (2012) proposed fast algorithms for computing CV error for each candidate. Because of the simplicity and the universality, CV is a widespread strategy for model selection (Arlot et al., 2010).

As mentioned above, CV works with a group of candidate values of parameter. Normally, the parameter are searched by some strategies. The most popular strategy is grid searching. For example, the regularization parameter C of SVM is searched on a 18 grid linearly spaced in the region $\{(\log_2 C)| -9 \leq \log_2 C \leq 8\}$, as used in Yang & Ong (2011). Grid search is reliable in low dimensional parameter spaces. For high dimensional parameter spaces (such as parameters in Deep Belief Networks (Hinton et al., 2006)), manual search (Hinton, 2010), and random search (Bergstra & Bengio, 2012) were often used for CV. However, as we know, CV with grid, manual, and random search strategies only considers finite candidates due to the limited computing resources. It cannot guarantee to find the model with the minimum CV error in the whole parameter space, which is often the ultimate goal of model selection.

In this decade, a novel learning methodology called solution path (Hastie et al., 2004) has been developed for con-

Proceedings of the 32^{nd} International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37.

tinuously tracing the solutions with respect to a parameter. In a solution path, one solution can act on an interval of the regularization parameter in which the solutions share a same linearity property. Thus, solution path algorithm can effectively represent the entire solutions based on a finite number of solutions at the knee points. Solution path is much more general as it represents an entire continuous space, than the grid, and other common search strategies as they only represents a finite number of discrete points.

Solution path algorithms have been proposed for several important learning algorithms. For example, Hastie et al. (2004) proposed a solution path algorithm for C-support vector classification (C-SVC). Bach et al. (2006) proposed solution path algorithm for $2C$-support vector classification ($2C$-SVC). Gunter & Zhu (2007), and Wang et al. (2008) proposed solution path algorithms for ε-support vector regression (ε-SVR) to trace the solutions w.r.t. ε and the regularization parameter, respectively. Rosset & Zhu (2007) proposed a solution path algorithm for Lasso. Takeuchi et al. (2009) proposed a solution path for kernel quantile regression (KQR). Ong et al. (2010) proposed an improved solution path algorithm to handle the singularities encountered in the method of Hastie et al. (2004). Karasuyama & Takeuchi (2011) proposed an approximate solution path for C-SVC. Gu et al. (2012) proposed a solution path algorithm for ν-support vector classification (ν-SVC). Giesen et al. (2012) proposed an approximate solution path algorithm for a general class of regularized optimization problems.

Solution path algorithms can fit the entire solutions with respect to one parameter. However, they still do not guarantee to find the model with minimum CV error, as they do not correspond to "error path" in CV. That is, previous solution path algorithms do not directly lead to global minimal CV error. To the best of our knowledge, the only work on CV with global search is (Yang & Ong, 2011)[1]. Based on the solution paths of (Hastie et al., 2004; Ong et al., 2010), Yang & Ong (2011) proposed an error path algorithm mainly for C-SVC and standard error function of binary classification. A remark in (Yang & Ong, 2011) said that the error path algorithm is also applicable, with minor modifications, when the error function[2] is the weighted error rate, the precision, the recall, and the F-measure. It can guarantee to find the model with minimum CV error. However, their method is limited to the two solution path algorithms and error functions of binary classification.

[1]Recently, Shibagaki et al. (2015) proposed a method for approximately computing error path, which solves the optimization problem for multiple times. In this paper, we mainly discuss the fast algorithm of error path, based on solution path.

[2]Although several functions (e.g., the precision, the recall, and the F-measure) are related to accuracy, we totally call them error functions in this paper for simplifying terms.

In this paper, we design a new generalized error path algorithm (GEP). We believe this is very important for model selection, for three reasons. First, GEP incorporates more general error (or loss) functions. Second, GEP works with more general solution path algorithms. Third and most importantly, we can obtain minimum CV errors directly for a large variety of error (or loss) functions and solution path algorithms. More specifically, we first show that the solution paths produced by various algorithms have the property of piecewise linearity. Then, we point out model function builds the bridge between solution path and error path, and show that the piecewise linearity of solution path leads to the piecewise linearity of model function. Based on the piecewise linearity of model function, we prove that a large class of error (or loss) functions are piecewise constant, linear, or quadratic w.r.t. the regularization parameter. Finally, we propose our GEP for the generalized error (or loss) functions and solution path algorithms, which guarantees to find the models with the minimum CV error. The experimental results on a variety of datasets not only confirm our theoretical findings, but also show that the best model with our GEP has better generalization error on the test data, compared to the grid search, manual search, and random search.

We organize the rest of the paper as follows. In Section 2 we propose our GEP algorithm. In Section 3 we show how to do CV based on the GEPs. In Section 4, we present the experimental results on a variety of datasets. Finally, in Section 5, we give some concluding remarks.

2. Generalized Error Path

As mentioned above, solution path algorithms do not directly lead to global minimal CV error; however, error path algorithm can. Thus, more attention should be paid to error path algorithms for any work related to CV. In the following, we first give a formal description to error path algorithm.

Given a validation set $\mathcal{V} = \{(\widetilde{x}_i, \widetilde{y}_i)\}_{i=1}^{\ell}$. The error path algorithm is trying to compute the error on the validation set $E(\hat{\alpha}(\lambda), \mathcal{V}, L)$ for all $\lambda \in [a, b]$, where $\hat{\alpha}(\lambda)$ is the solution of a learning problem, λ is the parameter of the learning problem, L is an error (loss) function.

In order to propose the generalized error path algorithm, we first give a brief review on the solution path algorithms (Section 2.1), then show that the model functions $f(x)$ are piecewise linear w.r.t. λ (see the left part of Fig. 1, and Section 2.2), when solution path is piecewise linear. Based on the piecewise linearity on model functions, we then show that the error path on common error (loss) functions could be piecewise constant, linear, or quadratic w.r.t. λ (see the right part of Fig. 1, and Section 2.3). Finally, we propose a

Table 1. Representative solution path algorithms. The piecewise linearities on these solution paths also produce the piecewise linearities on their model functions. (BC and R are the abbreviations of binary classification and regression, respectively.)

Problem	Task	Reference	Parameter	Exact	Piecewise
C-SVC	BC	Hastie et al. (2004)	Regularization parameter C	Yes	Linear
$2C$-SVC	BC	Bach et al. (2006)	Regularization parameters C_+, C_-	Yes	Linear
ε-SVR	R	Gunter & Zhu (2007)	Regularization parameter	Yes	Linear
ε-SVR	R	Wang et al. (2008)	Regularization parameter and ε	Yes	Linear
Lasso	R	Rosset & Zhu (2007)	Regularization parameter	Yes	Linear
KQR	R	Takeuchi et al. (2009)	Quantile order $\tau \in (0, 1)$	Yes	Linear
C-SVC	BC	Ong et al. (2010)	Regularization parameter C	Yes	Linear
C-SVC	BC	Karasuyama & Takeuchi (2011)	Regularization parameter C	No	Linear
ν-SVC	BC	Gu et al. (2012)	Regularization parameter ν	Yes	Linear
General	BC+R	Giesen et al. (2012)	Regularization parameter	No	Linear

framework to compute generalized error path (Section 2.4), which may be piecewise constant, linear, or quadratic w.r.t. λ.

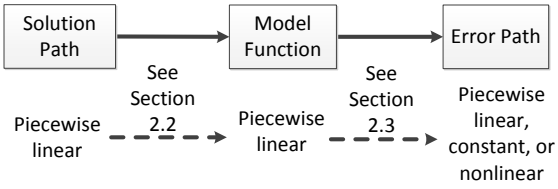

Figure 1. Model function builds a bridge from solution path to error path. **Left**: The piecewise linearity of solution path leads to the piecewise linearity of model function. **Right**: The error path could be piecewise constant, linear, or quadratic, based on the piecewise linearity on model functions.

2.1. Brief Review on Solution Path

As we have mentioned, solution path can efficiently trace the solutions with respect to a parameter. In this section, we will first give a more formal description about solution path algorithm, and then look through various solution path algorithms. After extensive literature search, it seems that most of solution paths have the property of piecewise linearity.

Given a generic learning problem $J(\alpha, S, \lambda)$, where $S = \{(x_i, y_i)\}_{i=1}^l$ is a training set, $x_i \in \mathbb{R}^d$, $y_i \in \{+1, -1\}$ is for binary classification, $y_i \in \mathbb{R}$ is for regression, α is the coefficients of model, λ is a parameter of the optimization problem. The solution path algorithm is trying to compute $\hat{\alpha}(\lambda) = \arg\min_\alpha J(\alpha, S, \lambda)$ for all $\lambda \in [a, b]$.

As previously mentioned, there have been various solution path algorithms (Hastie et al., 2004; Bach et al., 2006; Gunter & Zhu, 2007; Wang et al., 2008; Rosset & Zhu, 2007; Takeuchi et al., 2009; Ong et al., 2010; Karasuyama & Takeuchi, 2011; Gu et al., 2012; Giesen et al., 2012) proposed for important learning algorithms (e.g., C-SVC, $2C$-SVC, ε-SVR, Lasso, KQR, ν-SVC, and so on). We reorganize them into Table 1. From Table 1, it shows that all these solution path algorithms are returning the exact solutions to the corresponding problems, except Karasuyama & Takeuchi (2011) and Giesen et al. (2012). A more important observation in Table 1 is that all these solution paths are piecewise linear w.r.t. the parameter listed in Table 1. It means that the piecewise linearity is important for designing solution path algorithms.

Besides being used to design solution path algorithms, the property of piecewise linearity can also produce the piecewise linearity on model function (we will discuss it in detail in Section 2.2), which can be used to efficiently compute error path in CV. Thus, the piecewise linearity of solution path is is a core conception in this paper.

A formal definition about the piecewise linearity of the solution path was gave in Rosset & Zhu (2007). However, their definition is for the exact solution path. In order to incorporate the approximate solution paths (e.g. Karasuyama & Takeuchi (2011)), we modify the definition, and give a more general one as following:

Definition 1. *Suppose $\tilde{\alpha}(\lambda)$ is returned by a solution path. The solution $\tilde{\alpha}(\lambda)$ is called piecewise linear as a function of λ, if existing $a = a_0 < a_1 < a_2 < \cdots < a_m = b$, and the corresponding vectors $\beta^{[1]}, \beta^{[2]}, \cdots, \beta^{[m]}$, such that the solution $\hat{\alpha}(\lambda)$ is given exactly or approximately, by $\tilde{\alpha}(a_{k-1}) + \beta^{[k]}(\lambda - a_{k-1})$, $\forall \lambda \in [a_{k-1}, a_k]$.*

2.2. From Solution Path to Model Function

As we all know, learning a (linear or nonlinear) model $f(x)$ with a linear representation $f(x) = \langle G(x), \alpha \rangle$ is dominant in machine learning[3]. For example, to obtain a nonlinear model, a popular way is learning a linear model in a high dimensional kernel space, based on the representer theorem (Scholkopf & Smola, 2002). Importantly, all the problems in Table 1 are also considering models with a linear representation, which are verified as following:

[3] $G(x)$ denotes a mapping function from x to a vector with the same size of α.

1. For C-SVC, $2C$-SVC, ν-SVC, the models are $f(x) = \sum_i^l \alpha_i y_i K(x, x_i) + \alpha_0$.
2. For ε-SVR, we define an extended training sample set $\{(x_i, y_i, z_i = -1)\}_{i=1}^l \cup \{(x_i, y_i, z_i = +1)\}_{i=1}^l$, where z_i is the label of the training sample (x_i, y_i). The model of ε-SVR is $f(x) = \sum_i^{2l} \alpha_i z_i K(x, x_i) + \alpha_0$.
3. For Lasso, the model is $f(x) = x^T \alpha$.
4. For KQR, the model is $f(x) = \sum_i^l \alpha_i K(x, x_i) + \alpha_0$.

where $K(\cdot, \cdot)$ is a kernel function, $\hat{\alpha}_0$ is the offset of the model function.

Because of the linear representation of models, in Lemma 1, we show that the piecewise linearity of solution path leads to the piecewise linearity of model function. The detailed proof of Lemma 1 is presented in Appendix A.

Lemma 1. *Given an interval $[a_{k-1}, a_k]$ in a solution path, and the corresponding vector $\beta^{[k]}$. If the model is with linear representation, there exists a scale $\gamma^{[k]}$ such that the model function $f_\lambda(x)$ can be represented as $f_\lambda(x) = f_{a_{k-1}}(x) + \gamma^{[k]}(\lambda - a_{k-1})$, $\forall \lambda \in [a_{k-1}, a_k]$.*

Specifically, γ in the problems of Table 1 can be computed as following:

1. For C-SVC, $2C$-SVC, and ν-SVC, $\gamma = \sum_i^l \beta_i y_i K(x, x_i) + \beta_0$.
2. For ε-SVR, $\gamma = \sum_i^{2l} \beta_i z_i K(x, x_i) + \beta_0$.
3. For Lasso, $\gamma = x^T \beta$.
4. For KQR, $\gamma = \sum_i^l \beta_i K(x, x_i) + \beta_0$.

Thus, we know that the model functions are also piecewise linear w.r.t. λ, when solution path is piecewise linear. The piecewise linearity of model functions will help us to reveal the relationship between the error path on common error (loss) functions and the parameter λ.

2.3. From Model Function to Error Path

The model functions are piecewise linear w.r.t. λ as mentioned above. Thus, given a validation set $\mathcal{V} = \{(\widetilde{x}_i, \widetilde{y}_i)\}_{i=1}^\ell$, and an interval $[a_{k-1}, a_k]$, we can fit all errors on the validation set for common error (or loss) functions of regression and binary classification problems.

In the following, we will show that the error path on the entire interval $[a, b]$ could be piecewise quadratic, linear, or constant w.r.t. λ, on common error (loss) functions for regression and binary classification problems.

Piecewise Quadratic (Type-1): For regression, if we consider square error (loss) function $L(\widetilde{y}, \widehat{y}) = (\widetilde{y} - \widehat{y})^2$, where $\widehat{y} = f_\lambda(\widetilde{x})$ is the predicted value of input $\widetilde{x}$, the mean square error (MSE) on the validation set is

$$
\begin{aligned}
E(\hat{\alpha}(\lambda), \mathcal{V}, L) &= \frac{1}{\ell} \sum_{i=1}^{\ell} L(\widetilde{y}_i, \widehat{y}_i) \qquad (1)\\
&= \frac{1}{\ell} \sum_{i=1}^{\ell} (\widetilde{y}_i - f_\lambda(\widetilde{x}_i))^2 \\
&= \frac{1}{\ell} \sum_{i=1}^{\ell} \left(\widetilde{y}_i - f_{a_{k-1}}(\widetilde{x}_i) - \gamma_i^{[k]}(\lambda - a_{k-1})\right)^2 \\
&= \frac{1}{\ell} \sum_{i=1}^{\ell} (\gamma_i^{[k]})^2 \lambda^2 + \\
&\quad \frac{1}{\ell} \sum_{i=1}^{\ell} \left(\widetilde{y}_i - f_{a_{k-1}}(\widetilde{x}_i) + \gamma_i^{[k]} a_{k-1}\right)^2 \\
&\quad - \frac{1}{\ell} \sum_{i=1}^{\ell} 2 \times \gamma_i^{[k]} (\widetilde{y}_i - f_{a_{k-1}}(\widetilde{x}_i) + \gamma_i^{[k]} a_{k-1}) \lambda
\end{aligned}
$$

Based on (1), it is easy to see that $E(\hat{\alpha}(\lambda), \mathcal{V}, L)$ is a quadratic function of λ in the interval $[a_{k-1}, a_k]$. Thus, MSE is *piecewise quadratic* w.r.t. λ (see Fig. 2), in the whole interval $[a, b] = \bigcup_{k=1}^m [a_{k-1}, a_k]$.

Piecewise Linear (Type-2): If we consider the absolute error (loss) function $L(\widetilde{y}, \widehat{y}) = |\widetilde{y} - \widehat{y}| = |\widetilde{y} - f_\lambda(\widetilde{x})|$ for regression, according to the sign of $\widetilde{y} - f_\lambda(\widetilde{x})$, the validation set $\mathcal{V}$ can have the partition $\pi(\lambda)$ as:

$$
\begin{aligned}
\pi(\lambda) = \{\{i \in \mathcal{V} : \widetilde{y}_i - f_\lambda(\widetilde{x}_i) \geq 0\}, \qquad (2)\\
\{i \in \mathcal{V} : \widetilde{y}_i - f_\lambda(\widetilde{x}_i) < 0\}\} \stackrel{\text{def}}{=} \{\mathcal{I}_+(\lambda), \mathcal{I}_-(\lambda)\}
\end{aligned}
$$

Thus, we can define an invariant region of λ, which has a same partition $\pi(\lambda_0)$, as $\mathcal{IR}(\lambda_0) = \{\lambda \in [a_{k-1}, a_k] : \pi(\lambda) = \pi(\lambda_0)\}$. Theorem 1 shows that $\mathcal{IR}(\lambda_0)$ is a nontrivial interval region (not a single point). The detailed proof of Theorem 1 is presented in Appendix B.

Theorem 1. *The $\mathcal{IR}(\lambda_0)$ is a convex set and its closure is a nontrivial interval region.*

If an interval $\mathcal{IR}(\lambda_0)$ is given, the mean absolute error (MAE) on the validation set can be computed as:

$$
\begin{aligned}
E(\hat{\alpha}(\lambda), \mathcal{V}, L) &= \frac{1}{\ell} \sum_{i=1}^{\ell} |\widetilde{y}_i - f_\lambda(\widetilde{x}_i)| \qquad (3)\\
&= \frac{1}{\ell} \Big(\sum_{i \in \mathcal{I}_+(\lambda_0)} (\widetilde{y}_i - f_\lambda(\widetilde{x}_i)) - \sum_{i \in \mathcal{I}_-(\lambda_0)} (\widetilde{y}_i - f_\lambda(\widetilde{x}_i)) \Big) \\
&= \frac{1}{\ell} \left(\sum_{i \in \mathcal{I}_+(\lambda_0)} \left(\widetilde{y}_i - f_{\lambda_0}(\widetilde{x}_i) - \gamma_i^{[k]}(\lambda - \lambda_0)\right) \right.
\end{aligned}
$$

$$
\begin{aligned}
&- \sum_{i\in\mathcal{I}_-(\lambda_0)} \left(\widetilde{y}_i - f_{\lambda_0}(\widetilde{x}_i) - \gamma_i^{[k]}(\lambda-\lambda_0)\right)\Bigg) \\
= \;& \frac{1}{\ell}\left(\sum_{i\in\mathcal{I}_-(\lambda_0)} \gamma_i^{[k]} - \sum_{i\in\mathcal{I}_+(\lambda_0)} \gamma_i^{[k]}\right)\cdot\lambda \\
&+\frac{1}{\ell}\Bigg(\sum_{i\in\mathcal{I}_-(\lambda_0)} (\widetilde{y}_i - f_{\lambda_0}(\widetilde{x}_i) + \gamma_i^{[k]}\lambda_0) \\
&- \sum_{i\in\mathcal{I}_+(\lambda_0)} (\widetilde{y}_i - f_{\lambda_0}(\widetilde{x}_i) + \gamma_i^{[k]}\lambda_0)\Bigg)
\end{aligned}
$$

Based on (3), it is easy to see that $E(\hat{\alpha}(\lambda), \mathcal{V}, L)$ is a linear function of λ, in the interval $\mathcal{IR}(\lambda_0)$. Corollary 1 shows that, for each interval $[a_{k-1}, a_k]$, there exists an interval sequence $\{[b_0, b_1], [b_1, b_2], \cdots, [b_{n_k-1}, b_{n_k}]\}$ with $b_{i-1} < b_i$ such that $\bigcup_{i=1}^{n}[b_{i-1}, b_i] = [a_{k-1}, a_k]$, and each interval $[b_{i-1}, b_i]$ corresponds a $\mathcal{IR}(\lambda_i)$. According to the proof of theorem 1, Corollary 1 can be proved easily.

Corollary 1. *For each $[a_{k-1}, a_k]$, there exists a finite number n_k of λ_i, such that $\bigcup_{i=1}^{n_k}\mathcal{IR}(\lambda_i) = [a_{k-1}, a_k]$, and $\forall i, j$, if $i \neq j$, $\mathcal{IR}(\lambda_i) \cap \mathcal{IR}(\lambda_j) = \varnothing$ or $|\mathcal{IR}(\lambda_i) \cap \mathcal{IR}(\lambda_j)| = 1$.*

Thus, according to Theorem 1, Corollary 1, and the equation (3), MAE is *piecewise linear* w.r.t. λ (see Fig. 2), in the whole interval $[a, b]$.

Piecewise Constant (Type-3): For binary classification, if we consider standard loss function $L(\widetilde{y}, \widehat{y}) = \frac{1}{2}|\widetilde{y} - \widehat{y}|$, where $\widehat{y} = \text{sign}(f_\lambda(\widetilde{x}))$ is the predicted label of input $\widetilde{x}$. According to the sign of $f_\lambda(\widetilde{x})$, the validation set $\mathcal{V}$ can have the partition $\widetilde{\pi}(\lambda)$ as:

$$
\begin{aligned}
\widetilde{\pi}(\lambda) &= \{\{i \in \mathcal{V} : f_\lambda(\widetilde{x}_i) \geq 0\}, \{i \in \mathcal{V} : f_\lambda(\widetilde{x}_i) < 0\}\} \\
&\overset{\text{def}}{=} \left\{\widetilde{\mathcal{I}}_+(\lambda), \widetilde{\mathcal{I}}_-(\lambda)\right\} \qquad (4)
\end{aligned}
$$

Thus, we can also define a invariant region of λ, which has a same partition $\widetilde{\pi}(\lambda_0)$, as $\widetilde{\mathcal{IR}}(\lambda_0) = \{\lambda \in [a_{k-1}, a_k] : \widetilde{\pi}(\lambda) = \widetilde{\pi}(\lambda_0)\}$. Theorem 2 also shows that $\widetilde{\mathcal{IR}}(\lambda_0)$ is a nontrivial interval region. It can be proved, similar to Theorem 1.

Theorem 2. *The $\widetilde{\mathcal{IR}}(\lambda_0)$ is a convex set and its closure is a nontrivial interval region.*

If an interval $\widetilde{\mathcal{IR}}(\lambda_0)$ is given, the error rate (ER) on the validation set can be computed as:

$$
\begin{aligned}
E(\hat{\alpha}(\lambda), \mathcal{V}, L) &= \frac{1}{2\ell}\sum_{i=1}^{\ell} |\widetilde{y}_i - \text{sign}\left(f_\lambda(\widetilde{x}_i)\right)| \\
&= \frac{1}{2\ell}\left(\sum_{i\in\mathcal{I}_+(\lambda_0)} |\widetilde{y}_i - 1| + \sum_{i\in\mathcal{I}_-(\lambda_0)} |\widetilde{y}_i + 1|\right) \qquad (5)
\end{aligned}
$$

Based on (5), it is easy to see that $E(\hat{\alpha}(\lambda), \mathcal{V}, L)$ is a constant function of λ, in an interval $\widetilde{\mathcal{IR}}(\lambda_0)$. Corollary 2 shows that, for each interval $[a_{k-1}, a_k]$, there exists an interval sequence $\{[b_0, b_1], [b_1, b_2], \cdots, [b_{n_k-1}, b_{n_k}]\}$ with $b_{i-1} < b_i$ such that $\bigcup_{i=1}^{n}[b_{i-1}, b_i] = [a_{k-1}, a_k]$, and each interval $[b_{i-1}, b_i]$ corresponds a $\widetilde{\mathcal{IR}}(\lambda_i)$. It can be proved, similar to Corollary 1.

Corollary 2. *For each $[a_{k-1}, a_k]$, there exists a finite number n_k of λ_i, such that $\bigcup_{i=1}^{n_k}\widetilde{\mathcal{IR}}(\lambda_i) = [a_{k-1}, a_k]$, and $\forall i, j$, if $i \neq j$, $\widetilde{\mathcal{IR}}(\lambda_i) \cap \widetilde{\mathcal{IR}}(\lambda_j) = \varnothing$ or $|\widetilde{\mathcal{IR}}(\lambda_i) \cap \widetilde{\mathcal{IR}}(\lambda_j)| = 1$.*

Thus, according to Theorem 2, Corollary 2, and the equation (5), ER is *piecewise constant* w.r.t. λ (see Fig. 2), in the whole interval $[a, b]$.

Similarly, we can also show that, the weighted error rate, the precision, the recall, and the F-measure (Yang & Ong, 2011) are *piecewise constant* w.r.t. λ.

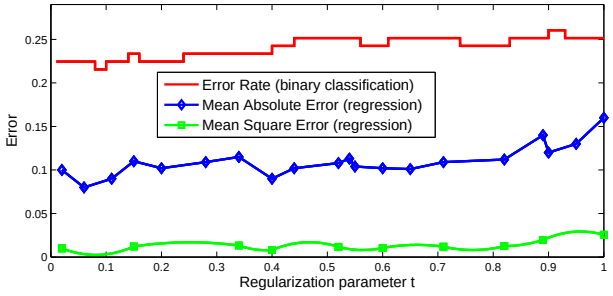

Figure 2. The error path w.r.t. the regularization parameter.

2.4. Computing Generalized Error Path

The last step in Fig. 1 is to compute generalized error path, which may be piecewise constant, linear, or quadratic w.r.t. λ. To fit various error paths, we define a 2-tuple (I, P), where I is an interval, P is a set of parameters describing the constant, linear, and quadratic functions in the interval I. According to (1), (3), and (5), (I, P) for the three types of error path is defined, respectively, as following:

1. For piecewise quadratic (**Type-1**) error path:
   $I = [a_{k-1}, a_k]$, $P.c_2 = \frac{1}{\ell}\sum_{i=1}^{\ell}(\gamma_i^{[k]})^2$,
   $P.c_1 = -\frac{1}{\ell}\sum_{i=1}^{\ell} 2\times\gamma_i^{[k]}(\widetilde{y}_i - f_{a_{k-1}}(\widetilde{x}_i) + \gamma_i^{[k]}a_{k-1})$,
   $P.c_0 = \frac{1}{\ell}\sum_{i=1}^{\ell}\left(\widetilde{y}_i - f_{a_{k-1}}(\widetilde{x}_i) + \gamma_i^{[k]}a_{k-1}\right)^2$.

2. For piecewise linear (**Type-2**) error path:
   $I = \mathcal{IR}(\lambda_0)$,
   $P.c_1 = \frac{1}{\ell}\left(\sum_{i\in\mathcal{I}_-(\lambda_0)}\gamma_i^{[k]} - \sum_{i\in\mathcal{I}_+(\lambda_0)}\gamma_i^{[k]}\right)$,
   $P.c_0 = \frac{1}{\ell}(\sum_{i\in\mathcal{I}_-(\lambda_0)}(\widetilde{y}_i - f_{\lambda_0}(\widetilde{x}_i) + \gamma_k\lambda_0) - \sum_{i\in\mathcal{I}_+(\lambda_0)}(\widetilde{y}_i - f_{\lambda_0}(\widetilde{x}_i) + \gamma_k\lambda_0))$.

Algorithm 1 GEP (Generalized error path algorithm)

Input: A solution path w.r.t. λ in an interval $[a, b] = \bigcup_{k=1}^{m}[a_{k-1}, a_k]$, a validation set $\mathcal{V}$.
Output: An error path $\{(I_1, P_1), (I_2, P_2), (I_3, P_3), \cdots\}$.

```
Initialize λ = a, k = 0, j = 1.
while λ < b do
    Update k = k + 1, read the k-th sub-interval [a_{k-1}, a_k] from the solution path.
    while λ < a_k do
        Compute (I_j, P_j) for the leftmost in [λ, a_k].
        Update λ = R(I_j), j = j + 1.
    end while
end while
```

3. For piecewise constant (**Type-3**) error path: $I = \widetilde{\mathcal{IR}}(\lambda_0)$, $P.c_0 = E(\hat{\alpha}(\lambda_0), \mathcal{V}, L)$.

Given a solution path w.r.t. λ in an interval $[a, b] = \bigcup_{k=1}^{m}[a_{k-1}, a_k]$, to compute error path for the interval $[a_{k-1}, a_k]$, we explore all (I, P) in the $[a_{k-1}, a_k]$. Repeating this procedure on all $[a_{k-1}, a_k]$ derives a generalized error path algorithm (i.e., GEP, Algorithm 1). Note that in Algorithm 1, the function $R(I)$ is to return the rightmost point in the interval I.

3. Cross Validation with GEP

Based on GEP, we expect to compute the error path for CV, which can definitely find the model with the global minimum CV error. Without losing generality, we consider K-fold CV here, however, extending to other types of CV (e.g. repeated random sub-sampling validation, leave-one-out CV, etc.) is straightforward.

In K-fold CV, the samples are randomly partitioned into K equal size subsets (i.e., $\mathcal{V}_1, \cdots, \mathcal{V}_K$). For each $k = 1, \cdots, K$, we train a model from the other $K-1$ parts, and compute its validation error $E(\hat{\alpha}^k(\lambda), \mathcal{V}_k, L)$ in predicting the k-th part. The K-fold CV error is given as $CV(\lambda) = \frac{1}{K}\sum_{k=1}^{K} E(\hat{\alpha}^k(\lambda), \mathcal{V}_k, L)$.

As mentioned above, K-fold CV error is the mean of K validation errors on the K folds. Thus, the error path of K-fold CV intuitively can be obtained by averaging K error paths. A simple merge procedure (i.e., CV-GEP, Algorithm 2), like the one used in merge sort (Cormen et al., 2009), can be designed to compute the average of K validation error paths. Specifically, we give an index i_k for each error path to point the corresponding 2-tuple $(I_{i_k}^k, P_{i_k}^k)$. Initially, all indices are set to 1. In each iteration, we compute the intersection of the K intervals $\{I_{i_k}^k\}_{k=1}^K$, and the corresponding parameters with averaging K 2-tuples. It gives a 2-tuple (I_j, P_j) for the error path of K-fold CV. In the following, we give the details of computing I_j and P_j in each iteration.

Algorithm 2 CV-GEP (Cross validation with GEP)

Input: K error paths $\{(I_1^k, P_1^k), (I_2^k, P_2^k), (I_3^k, P_3^k), \cdots\}_{k=1}^K$ in the interval $[a, b]$.
Output: An error path $\{(I_1, P_1), (I_2, P_2), (I_3, P_3), \cdots\}$.

```
Initialize λ = a, i^1 = 1, ··· , i^K = 1, j = 1.
while λ < b do
    Compute (I_j, P_j) from (I_{i^1}^1, P_{i^1}^1), ··· , (I_{i^K}^K, P_{i^K}^K).
    Update i^1, ··· , i^K. (If R(I_j) = R(I_{i^k}^k), update i^k = i^k + 1.)
    Update λ = R(I_j), and j = j + 1.
end while
```

1. The interval I_j is the intersection of the K intervals $\{I_{i_k}^k\}_{k=1}^K$, and can be computed as $I_j = I_{i_1}^1 \cap I_{i_2}^2 \cap \cdots \cap I_{i_K}^K$.
2. According to (1), (3), and (5), the set of parameters in P_j is computed as $P_j.c_2 = \frac{1}{K}\sum_{k=1}^{K} P_{i_k}^k.c_2$ (if existing), $P_j.c_1 = \frac{1}{K}\sum_{k=1}^{K} P_{i_k}^k.c_1$ (if existing), and $P_j.c_0 = \frac{1}{K}\sum_{k=1}^{K} P_{i_k}^k.c_0$.

If $R(I_j) = R(I_{i^k}^k)$, we update $i^k = i^k + 1$ in each iteration. Repeating this procedure until all intervals $I_{i_k}^k$ in the K error paths are scanned, produces an error path for K-fold CV. Based on the error path of K-fold CV, the values of λ with the minimum K-fold CV error can be found easily.

4. Experiments

In this section, we first give the experimental setup, then present our experimental results and discussion.

4.1. Experimental Setup

Design of Experiments: As mentioned above, GEP can theoretically find the global minimum CV error with a finite number of points. In this section, we do experiments not only to verify our theoretical findings, but also to show that the best model with our GEP has better generalization error on the test data, compared to the grid search, manual search, and random search.

Because this paper focuses on binary classification and regression, we do experiments mainly on C-SVC, and Lasso. The corresponding solution path algorithms used in our experiments are Hastie et al. (2004); Rosset & Zhu (2007), respectively. In order to compare different methods fairly, we set the numbers of candidate points in grid search, manual search, and random search are same with the number of knee points of solution path.

Implementation: We implemented our GEP in MATLAB. For the solution path of C-SVC, we used the

implementation in http://web.eecs.umich.edu/~cscott/code.html#svmpath, where the Gaussian kernel $K(x_1, x_2) = \exp(-\kappa\|x_1 - x_2\|^2)$ with $\kappa = 0.5$ was used. We implemented the solution path algorithm of (Rosset & Zhu, 2007) for Lasso. In addition, the details of the implementations of the grid search, manual search, and random search are described as following:

1. grid search (GS): GS is done on a τ grid linearly spaced in the region $\{\log_2 \lambda | -20 \leq \log_2 \lambda \leq 20\}$.
2. manual search (MS): MS is done on the set $\{1, 10, 100\}$, then followed by a fine GS on a $\tau-3$ uniform grid linearly spaced by 0.1 in the $\log_2 \lambda$ space. If $\tau < 4$, the fine GS will be ignored.
3. random search (RS): RS is done with τ random points generated from the uniform distribution in the region $\{\log_2 \lambda | -20 \leq \log_2 \lambda \leq 20\}$.

where τ is the average number of knee points in the solution paths.

Datasets: The Ionosphere, Diabetes, Hill-Valley, Breast Cancer, Housing, Forest Fires, Auto MPG, and Triazines datasets are from the UCI benchmark repository (Bache & Lichman, 2013). Friedman is an artificial data set as produced in (Friedman, 1991).

The Spine dataset collected by us is to diagnose degenerative disc disease depending on five image texture features quantified from magnetic resonance imaging, where 157 records were marked normal and 193 records were marked abnormal by an experienced radiologist.

Table 2. Summary of datasets. (BC=binary classification, R=regression)

Dataset	Number of samples	Dimensionality	Task
Ionosphere	354	34	BC
Diabetes	768	8	BC
Hill-Valley	606	100	BC
Breast Cancer	683	10	BC
Spine	350	5	BC
Friedman	1,500	10	R
Housing	506	13	R
Forest Fires	517	12	R
Auto MPG	392	7	R
Triazines	186	60	R

We randomly partition each dataset into 65% training and 35% test sets. For each dataset, the training set is used with a 5-fold CV procedure to determine the optimal parameter.

4.2. Experimental Results and Discussion

As mentioned above, in order to compare different methods fairly, we set the numbers of candidate points in GS, MS, and RS are same with these numbers of knee points. Table 3 presents the average numbers of knee points in the solution paths of Hastie et al. (2004); Rosset & Zhu (2007) on different datasets.

Based on the average numbers of knee points in Table 3, Table 4 presents the average of CV errors obtained from GS, MS, RS, and our GEP, on a variety of datasets. The results confirm that, GEP guarantees to find the model with minimum CV error for the entire range of the regularization parameter, with a finite number of points.

Table 5 presents the average errors on the test data, over 10 trails, obtained from GS, MS, RS, and our GEP. The results show that the best model with our GEP has better generalization error on the test data, compared to the grid search, manual search, and random search.

Table 3. The average numbers of knee points in the solution paths on different datasets.

Dataset	Knee points	Dataset	Knee points
Ionosphere	168	Friedman	10
Diabetes	2	Housing	10
Hill-Valley	38	Forest Fires	12
Breast Cancer	1	Auto MPG	7
Spine	237	Triazines	12

5. Conclusion

In this paper, we proposed a new generalized error path algorithm (GEP). We believe this is very important for model selection, for the following three reasons. First, GEP incorporates more general error (or loss) functions. Second, GEP works with more general solution path algorithms. Third and most importantly, we can obtain minimum CV errors directly for a large variety of error (or loss) functions and solution path algorithms. The experimental results on a variety of datasets not only confirm our theoretical findings, but also show that the best model with our GEP has better generalization error on the test data, compared to the grid search, manual search, and random search.

As mentioned previously, we mainly discuss the error paths on binary classification and regression in this paper. If the solution paths of other learning problems (e.g. ordinal regression, ranking, multi classification, etc.) are available, we believe that the corresponding error paths can be obtained similarly. For several advantaged error (or accuracy) criterions (e.g. AUC (Ridgway et al., 2014)), we believe our GEP also works on them.

Table 4. The results of CV error obtained from GS, MS, RS, and our GEP.

Dataset	ER				Dataset	MAE				MSE			
	GS	MS	RS	GEP		GS	MS	RS	GEP	GS	MS	RS	GEP
Ionosphere	0.062	0.068	0.068	**0.060**	Friedman	2.08	2.1	2.07	**1.98**	6.81	6.82	6.82	**6.60**
Diabetes	0.655	0.655	0.565	**0.345**	Housing	2.12	2.13	2.13	**2.09**	8.11	8.14	8.14	**8.05**
Hill-Valley	0.465	0.470	0.468	**0.460**	Forest Fires	17.9	18.3	18.2	**17.2**	3251	3258	3263	**3238**
Breast Cancer	0.528	0.528	0.474	**0.342**	Auto MPG	2.39	2.41	2.39	**2.36**	9.62	9.63	9.62	**9.58**
Spine	0.060	0.058	0.059	**0.055**	Triazines	2.39	2.40	2.40	**2.37**	9.66	9.67	9.66	**9.63**

Table 5. The average errors on the test data, over 10 trails, obtained from GS, MS, RS, and our GEP.

Dataset	ER				Dataset	MAE				MSE			
	GS	MS	RS	GEP		GS	MS	RS	GEP	GS	MS	RS	GEP
Ionosphere	0.066	0.066	0.065	**0.063**	Friedman	2.12	2.16	2.14	**2.08**	6.84	6.83	6.84	**6.75**
Diabetes	0.356	0.356	0.358	**0.347**	Housing	2.05	2.06	2.06	**2.05**	8.56	8.63	8.66	**8.44**
Hill-Valley	0.514	0.519	0.514	**0.512**	Forest Fires	18.4	19.1	18.9	**17.9**	5468	5475	5471	**5446**
Breast Cancer	0.528	0.528	0.474	**0.342**	Auto MPG	2.74	2.75	2.76	**2.62**	14.6	14.6	14.7	**14.0**
Spine	0.060	0.061	0.061	**0.058**	Triazines	2.76	2.77	2.77	**2.54**	14.7	14.8	14.6	**14.2**

Appendix A: Proof of Lemma 1

Given an interval $[a_{k-1}, a_k]$ in the solution path, and the corresponding vector $\beta^{[k]}$. If the model is with linear representation, the model function $f_\lambda(x)$, $\forall\lambda \in [a_{k-1}, a_k]$, can be computed as:

$$\begin{aligned} & f_\lambda(x) \\ = \; & \langle G(x), \tilde{\alpha}(\lambda)\rangle = \langle G(x), \tilde{\alpha}(a_{k-1}) + \beta^{[k]}(\lambda - a_{k-1})\rangle \\ = \; & \langle G(x), \tilde{\alpha}(a_{k-1})\rangle + \langle G(x), \beta^{[k]}\rangle(\lambda - a_{k-1}) \qquad (6) \\ \stackrel{\text{def}}{=} \; & f_{a_{k-1}}(x) + \gamma^{[k]}(\lambda - a_{k-1}) \end{aligned}$$

where $\gamma^{[k]}$ denotes the linear relationship between $f_\lambda(x)$ and λ in the interval $[a_{k-1}, a_k]$, and can be computed by $\beta^{[k]}$. This completes the proof.

Appendix B: Proof of Theorem 1

Let λ_1 and λ_2 be two arbitrary values in $\mathcal{IR}(\lambda_0) \subseteq [a_{i-1}, a_i]$. $\forall\theta \in [0,1]$, we define $\lambda(\theta) = \theta\lambda_1 + (1-\theta)\lambda_2$, then we can prove that:

$$\begin{aligned} & \widetilde{y} - f_{\lambda(\theta)}(\widetilde{x}) \qquad (7) \\ = \; & \widetilde{y} - (f_{\lambda_0}(\widetilde{x}_i) - \gamma_i(\theta\lambda_1 + (1-\theta)\lambda_2 - \lambda_0)) \\ = \; & \theta\cdot(\widetilde{y} - f_{\lambda_0}(\widetilde{x}_i) - \gamma_i(\lambda_1 - \lambda_0)) \\ & +(1-\theta)\cdot(\widetilde{y} - f_{\lambda_0}(\widetilde{x}_i) - \gamma_i(\lambda_2 - \lambda_0)) \\ = \; & \theta\cdot(\widetilde{y} - f_{\lambda_1}(\widetilde{x})) + (1-\theta)\cdot(\widetilde{y} - f_{\lambda_2}(\widetilde{x})) \end{aligned}$$

Based on (7), we can conclude that, $\forall\theta \in [0,1]$, $\mathcal{I}_+(\lambda(\theta)) = \mathcal{I}_+(\lambda_0)$ and $\mathcal{I}_-(\lambda(\theta)) = \mathcal{I}_-(\lambda_0)$. Thus, we have that $\mathcal{IR}(\lambda_0)$ is a convex set.

Further, $\forall\lambda \in \mathcal{IR}(\lambda_0)$, according to the definition of $\pi(\lambda)$, we have

$$\forall i \in \mathcal{I}_+(\lambda_0): \quad \widetilde{y}_i - f_{\lambda_0}(\widetilde{x}_i) - \gamma_i(\lambda - \lambda_0) \geq 0 \qquad (8)$$

$$\forall i \in \mathcal{I}_-(\lambda_0): \quad \widetilde{y}_i - f_{\lambda_0}(\widetilde{x}_i) - \gamma_i(\lambda - \lambda_0) < 0 \qquad (9)$$

The closure of inequalities (8)-(9) can be rewritten as:

$$\begin{aligned} & \max_{\substack{i\in\mathcal{I}_+(\lambda_0)\wedge\gamma_i<0 \\ i\in\mathcal{I}_-(\lambda_0)\wedge\gamma_i>0}} \frac{\widetilde{y}_i - f_{\lambda_0}(\widetilde{x}_i)}{\gamma_i} + \lambda_0 \leq \lambda \qquad (10) \\ \leq \; & \min_{\substack{i\in\mathcal{I}_+(\lambda_0)\wedge\gamma_i>0 \\ i\in\mathcal{I}_-(\lambda_0)\wedge\gamma_i<0}} \frac{\widetilde{y}_i - f_{\lambda_0}(\widetilde{x}_i)}{\gamma_i} + \lambda_0 \end{aligned}$$

Assume $\mathcal{IR}(\lambda_0) = \{\lambda_0\}$, we have

$$\max_{\substack{i\in\mathcal{I}_+(\lambda_0)\wedge\gamma_i<0 \\ i\in\mathcal{I}_-(\lambda_0)\wedge\gamma_i>0}} \frac{\widetilde{y}_i - f_{\lambda_0}(\widetilde{x}_i)}{\gamma_i} = 0 \qquad (11)$$

$$\min_{\substack{i\in\mathcal{I}_+(\lambda_0)\wedge\gamma_i>0 \\ i\in\mathcal{I}_-(\lambda_0)\wedge\gamma_i<0}} \frac{\widetilde{y}_i - f_{\lambda_0}(\widetilde{x}_i)}{\gamma_i} = 0 \qquad (12)$$

If existing a sample i reaching the equality of (12), we can change $\pi(\lambda_0)$ as following:

1. if $i \in \mathcal{I}_+(\lambda_0)$, we update $\mathcal{I}_+(\lambda_0) \leftarrow \mathcal{I}_+(\lambda_0) - \{i\}$, and $\mathcal{I}_-(\lambda_0) \leftarrow \mathcal{I}_-(\lambda_0) \cup \{i\}$.

2. if $i \in \mathcal{I}_-(\lambda_0)$, we update $\mathcal{I}_-(\lambda_0) \leftarrow \mathcal{I}_-(\lambda_0) - \{i\}$, and $\mathcal{I}_+(\lambda_0) \leftarrow \mathcal{I}_+(\lambda_0) \cup \{i\}$.

Thus, the interval region of (10) corresponding to the new $\pi(\lambda_0)$ is not a single point. This completes the proof.

Acknowledgments

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the NSF of China (61202137).

References

Arlot, Sylvain, Celisse, Alain, et al. A survey of cross-validation procedures for model selection. *Statistics surveys*, 4:40–79, 2010.

Bach, Francis R, Heckerman, David, and Horvitz, Eric. Considering cost asymmetry in learning classifiers. *The Journal of Machine Learning Research*, 7:1713–1741, 2006.

Bache, K. and Lichman, M. UCI machine learning repository, 2013. URL http://archive.ics.uci.edu/ml.

Bergstra, James and Bengio, Yoshua. Random search for hyper-parameter optimization. *The Journal of Machine Learning Research*, 13(1):281–305, 2012.

Cormen, Thomas H, Leiserson, Charles E, Rivest, Ronald L, and Stein, Clifford. Introduction to algorithms. third, 2009.

Foster, Kenneth R, Koprowski, Robert, and Skufca, Joseph D. Machine learning, medical diagnosis, and biomedical engineering research-commentary. *Biomed Eng Online*, 13(94):10–1186, 2014.

Friedman, Jerome H. Multivariate adaptive regression splines. *The annals of statistics*, pp. 1–67, 1991.

Giesen, Joachim, Müller, Jens, Laue, Soeren, and Swiercy, Sascha. Approximating concavely parameterized optimization problems. In *Advances in Neural Information Processing Systems*, pp. 2105–2113, 2012.

Gu, Bin, Wang, Jian-Dong, Zheng, Guan-Sheng, and Yu, Yue-Cheng. Regularization path for ν-support vector classification. *Neural Networks and Learning Systems, IEEE Transactions on*, 23(5):800–811, 2012.

Gunter, Lacey and Zhu, Ji. Efficient computation and model selection for the support vector regression. *Neural Computation*, 19(6):1633–1655, 2007.

Hastie, Trevor, Rosset, Saharon, Tibshirani, Robert, and Zhu, Ji. The entire regularization path for the support vector machine. In *Journal of Machine Learning Research*, pp. 1391–1415, 2004.

Hinton, Geoffrey. A practical guide to training restricted boltzmann machines. *Momentum*, 9(1):926, 2010.

Hinton, Geoffrey, Osindero, Simon, and Teh, Yee-Whye. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

Izbicki, Michael. Algebraic classifiers: a generic approach to fast cross-validation, online training, and parallel training. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pp. 648–656, 2013.

Jahrer, Michael and Töscher, Andreas. Collaborative filtering ensemble. In *KDD Cup*, pp. 61–74, 2012.

Karasuyama, Masayuki and Takeuchi, Ichiro. Suboptimal solution path algorithm for support vector machine. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pp. 473–480, 2011.

Ong, Chong-Jin, Shao, Shiyun, and Yang, Jianbo. An improved algorithm for the solution of the regularization path of support vector machine. *Neural Networks, IEEE Transactions on*, 21(3):451–462, 2010.

Pahikkala, Tapio, Suominen, Hanna, and Boberg, Jorma. Efficient cross-validation for kernelized least-squares regression with sparse basis expansions. *Machine learning*, 87(3):381–407, 2012.

Ridgway, James, Alquier, Pierre, Chopin, Nicolas, and Liang, Feng. Pac-bayesian auc classification and scoring. In *Advances in Neural Information Processing Systems*, pp. 658–666, 2014.

Rosset, Saharon and Zhu, Ji. Piecewise linear regularized solution paths. *The Annals of Statistics*, pp. 1012–1030, 2007.

Scholkopf, Bernhard and Smola, Alex. Learning with kernels. *MIT Press*, 11:110–146, 2002.

Shibagaki, Atsushi, Suzuki, Yoshiki, and Takeuchi, Ichiro. Approximately optimal selection of regularization parameters in cross-validation for regularized classifiers. *arXiv preprint arXiv:1502.02344*, 2015.

Takeuchi, Ichiro, Nomura, Kaname, and Kanamori, Takafumi. Nonparametric conditional density estimation using piecewise-linear solution path of kernel quantile regression. *Neural Computation*, 21(2):533–559, 2009.

Tibshirani, Robert. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

Usai, M Graziano, Goddard, Mike E, and Hayes, Ben J. Lasso with cross-validation for genomic selection. *Genetics research*, 91(06):427–436, 2009.

Vapnik, V. *Statistical Learning Theory*. John Wiley and Sons, Inc., New York, NY, 1998.

Wang, Gang, Yeung, Dit-Yan, and Lochovsky, Frederick H. A new solution path algorithm in support vector regression. *Neural Networks, IEEE Transactions on*, 19(10): 1753–1767, 2008.

Yang, Jian-Bo and Ong, Chong-Jin. Determination of global minima of some common validation functions in support vector machine. *Neural Networks, IEEE Transactions on*, 22(4):654–659, 2011.