Optimization for classification and regression SVM

Lecture 7

(Introduce ε-SVR, ν-SVR),

SVM

- The SVM binary classification algorithm searches for an optimal hyperplane that separates the data into two classes. For separable classes, the optimal hyperplane maximizes a margin (space that does not contain any observations) surrounding itself, which creates boundaries for the positive and negative classes. For inseparable classes, the objective is the same, but the algorithm imposes a penalty on the length of the margin for every observation that is on the wrong side of its class boundary.
- The linear SVM score function is
- $f(x)=x'\beta+b$,
- where:
- x is an observation (corresponding to a row of X).
- The vector β contains the coefficients that define an orthogonal vector to the hyperplane. For separable data, the optimal margin length is $2/\|\beta\|$.
- b is the bias term (corresponding to Mdl.Bias).
- The root of $f(x)$ for particular coefficients defines a hyperplane. For a particular hyperplane, $f(z)$ is the distance from point z to the hyperplane.

SVM

- The algorithm searches for the maximum margin length, while keeping observations in the positive (y = 1) and negative (y = –1) classes separate.
- For separable classes, the objective is to minimize ||β|| with respect to the β and b subject to $y_j f(x_j) \geq 1$, for all j = 1,..,n. This is the primal formalization for separable classes. For inseparable classes, the algorithm uses slack variables (ξ_j) to penalize the objective function for observations that cross the margin boundary for their class. $\xi_j = 0$ for observations that do not cross the margin boundary for their class, otherwise $\xi_j \geq 0$.
- The objective is to minimize $0.5\|\beta\|^2 + C\sum \xi_j$

  with respect to the β, b, and ξ_j subject to

and for all j = 1,..,n, and for a positive scalar box constraint C. This is the primal formalization for inseparable classes.

SVM

- The algorithm uses the Lagrange multipliers method to optimize the objective, which introduces n coefficients α1,...,αn . The dual formalizations for linear SVM are as follows:
- For separable classes, minimize

with respect to α1,. $0.5\sum_{j=1}^{n}\sum_{k=1}^{n}\alpha_j\alpha_k y_j y_k x_j' x_k - \sum_{j=1}^{n}\alpha_j$

, αj ≥ 0 for all j = 1,...,n, and Karush-Kuhn_Tucker (KKT) complimentarity conditions.

SVM

For inseparable classes, the objective is the same as for separable classes, except for the additional condition $0 \leq \alpha_j \leq C$ for all $j = 1,..,n$.

The resulting score function is

$$\hat{f}(x) = \sum_{j=1}^{n} \hat{\alpha}_j y_j x' x_j + \hat{b}.$$

- $\hat{b}$ is the e $\hat{\alpha}_j$ is the jth estimate of the vector $\hat{\alpha}$, $j = 1,...,n$. Written this way, the score function is free of the estimate of β as a result of the primal formalization.
- The SVM algorithm classifies a new observation z using $\text{sign}(\hat{f}(z))$.
- In some cases, a nonlinear boundary separates the classes. Nonlinear SVM works in a transformed predictor space to find an optimal, separating hyperplane.

SVM

The dual formalization for nonlinear SVM is

$$0.5\sum_{j=1}^{n}\sum_{k=1}^{n}\alpha_j\alpha_k y_j y_k G(x_j,x_k) - \sum_{j=1}^{n}\alpha_j$$

with respect to α αj≤C for all j = 1,..,n, and the KKT complementarity conditions. G(xk,xj) are elements of the Gram matrix. The resulting score function is

$$\hat{f}(x) = \sum_{j=1}^{n}\hat{\alpha}_j y_j G(x,x_j) + \hat{b}.$$

SVM Regression

- SVM regression is considered a nonparametric regression technique because it relies on kernel functions.
- Statistics and Machine Learning Toolbox implements linear epsilon-insensitive SVM (ε-SVM) regression, which is also known as L1 loss. In ε-SVM regression, the set of training data includes predictor variables and observed response values.
- The goal is to find a function f(x) that deviates from yn by a value no greater than ε for each training point x, and at the same time is as flat as possible.

SVM Regression

Suppose we have a set of training data where xn is a multivariate set of N observations with observed response values yn.

To find the linear function

$f(x)=x'\beta+b$,

and ensure that it is as flat as possible, find $f(x)$ with the minimal norm value ($\beta'\beta$). This is formulated as a convex optimization problem to minimize

subject to all residuals having a value less than ε:

SVM Regression

It is possible that no such function $f(x)$ exists to satisfy these constraints for all points. To deal with otherwise infeasible constraints, introduce slack variables ξ_n and ξ^*_n for each point. This approach is similar to the “soft margin” concept in SVM classification, because the slack variables allow regression errors to exist up to the value of ξ_n and ξ^*_n, yet still satisfy the required conditions.

Including slack variables leads to the objective function, also known as the primal formula:

,

subject to:

SVM Regression

The constant C is the box constraint, a positive numeric value that controls the penalty imposed on observations that lie outside the epsilon margin (ε) and helps to prevent overfitting (regularization). This value determines the trade-off between the flatness of f(x) and the amount up to which deviations larger than ε are tolerated.

The linear ε-insensitive loss function ignores errors that are within ε distance of the observed value by treating them as equal to zero. The loss is measured based on the distance between observed value y and the ε boundary. This is formally described by

$$L_\varepsilon = \begin{cases} 0 & \text{if } |y - f(x)| \le \varepsilon \\ |y - f(x)| - \varepsilon & \text{otherwise} \end{cases}$$

SVM Regression

- Linear SVM Regression: Dual Formula
- The optimization problem previously described is computationally simpler to solve in its Lagrange dual formulation. The solution to the dual problem provides a lower bound to the solution of the primal (minimization) problem.
- The optimal values of the primal and dual problems need not be equal, and the difference is called the “duality gap.” But when the problem is convex and satisfies a constraint qualification condition, the value of the optimal solution to the primal problem is given by the solution of the dual problem.

SVM Regression

To obtain the dual formula, construct a Lagrangian function from the primal function by introducing nonnegative multipliers αn and α*n for each observation xn. This leads to the dual formula, where we minimize

$$L(\alpha) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)x_i' x_j + \varepsilon\sum_{i=1}^{N}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{N} y_i(\alpha_i^* - \alpha_i)$$

subject to th[illegible]

The β parame[illegible]pletely described as a linear combination of the training [illegible]sing the equation

$$\sum_{n=1}^{N}(\alpha_n - \alpha_n^*) = 0$$
$$\forall n : 0 \le \alpha_n \le C$$
$$\forall n : 0 \le \alpha_n^* \le C .$$

$$\beta = \sum_{n=1}^{N}(\alpha_n - \alpha_n^*)x_n .$$

SVM Regression

The function used to predict new values depends only on the support vectors:

$$f(x) = \sum_{n=1}^{N} \left(\alpha_n - \alpha_n^*\right)\left(x_n' x\right) + b\,.$$

The Karush-Kuhn-Tucker (KKT) complementarity conditions are optimization constraints required to obtain optimal solutions. For linear SVM regression, these conditions are

$$\forall n : \alpha_n(\varepsilon + \xi_n - y_n + x_n'\beta + b) = 0$$
$$\forall n : \alpha_n^*(\varepsilon + \xi_n^* + y_n - x_n'\beta - b) = 0$$
$$\forall n : \xi_n(C - \alpha_n) = 0$$
$$\forall n : \xi_n^*(C - \alpha_n^*) = 0\,.$$

SVM Regression

These conditions indicate that all observations strictly inside the epsilon tube have Lagrange multipliers $\alpha n = 0$ and $\alpha n^* = 0$. If either αn or αn^* is not zero, then the corresponding observation is called a support vector.

The property Alpha of a trained SVM model stores the difference between two Lagrange multipliers of support vectors, $\alpha n - \alpha n^*$. The properties SupportVectors and Bias store xn and b, respectively.

SVM Regression

- Some regression problems cannot adequately be described using a linear model. In such a case, the Lagrange dual formulation allows the previously-described technique to be extended to nonlinear functions .
- Obtain a nonlinear SVM regression model by replacing the dot product x1′x2 with a nonlinear kernel function G(x1,x2) = <ϕ(x1),ϕ(x2)>, where ϕ(x) is a transformation that maps x to a high-dimensional space. Statistics and Machine Learning Toolbox provides the following built-in semidefinite kernel functions.

Kernel Function
$G(x_j, x_k) = x_j' x_k$
$G(x_j, x_k) = \exp(-\|x_j - x_k\|^2)$
$G(x_j, x_k) = (1 + x_j' x_k)^q$, where q is in the set {2,3,...}.

SVM Regression

The Gram matrix is an n-by-n matrix that contains elements $g_{i,j} = G(x_i, x_j)$. Each element $g_{i,j}$ is equal to the inner product of the predictors as transformed by ϕ. However, we do not need to know ϕ, because we can use the kernel function to generate Gram matrix directly. Using this method, nonlinear SVM finds the optimal function $f(x)$ in the transformed predictor space.

SVM Regression

The dual formula for nonlinear SVM regression replaces the inner product of the predictors (xi′xj) with the corresponding element of the Gram matrix (gi,j).

Nonlinear SVM regression finds the coefficients that minimize

$$L(\alpha) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)G(x_i, x_j) + \varepsilon\sum_{i=1}^{N}(\alpha_i + \alpha_i^*) - \sum_{i=1}^{N} y_i(\alpha_i - \alpha_i^*)$$

subject to

$$\sum_{n=1}^{N}(\alpha_n - \alpha_n^*) = 0$$

$$\forall n : 0 \le \alpha_n \le C$$

$$\forall n : 0 \le \alpha_n^* \le C\,.$$

SVM Regression

The function used to predict new values is equal to

$$f(x) = \sum_{n=1}^{N} (\alpha_n - \alpha_n^*) G(x_n, x) + b\,.$$

The KKT compleme

$$\forall n : \alpha_n(\varepsilon + \xi_n - y_n + f(x_n)) = 0$$
$$\forall n : \alpha_n^*(\varepsilon + \xi_n^* + y_n - f(x_n)) = 0$$
$$\forall n : \xi_n(C - \alpha_n) = 0$$
$$\forall n : \xi_n^*(C - \alpha_n^*) = 0\,.$$

SVM Regression

- Algorithms to Solve SVM Regression
- The minimization problem can be expressed in standard quadratic programming form and solved using common quadratic programming techniques. However, it can be computationally expensive to use quadratic programming algorithms, especially since the Gram matrix may be too large to be stored in memory. Using a decomposition method instead can speed up the computation and avoid running out of memory.
- Decomposition methods (also called chunking and working set methods) separate all observations into two disjoint sets: the working set and the remaining set. A decomposition method modifies only the elements in the working set in each iteration. Therefore, only some columns of the Gram matrix are needed in each iteration, which reduces the amount of storage needed for each iteration.

SVM Regression

- Algorithms to Solve SVM Regression
- Sequential minimal optimization (SMO) is the most popular approach for solving SVM problems. SMO performs a series of two-point optimizations. In each iteration, a working set of two points are chosen based on a selection rule that uses second-order information. Then the Lagrange multipliers for this working set are solved analytically.
- In SVM regression, the gradient vector ∇L for the active set is updated after each iteration.

SVM Regression

Algorithms to Solve SVM Regression

Iterative single data algorithm (ISDA) updates one Lagrange multiplier with each iteration. ISDA is often conducted without the bias term b by adding a small positive constant a to the kernel function. Dropping b drops the sum constraint

$$\sum_{n=1}^{N} \left(\alpha_i - \alpha^*\right) = 0$$

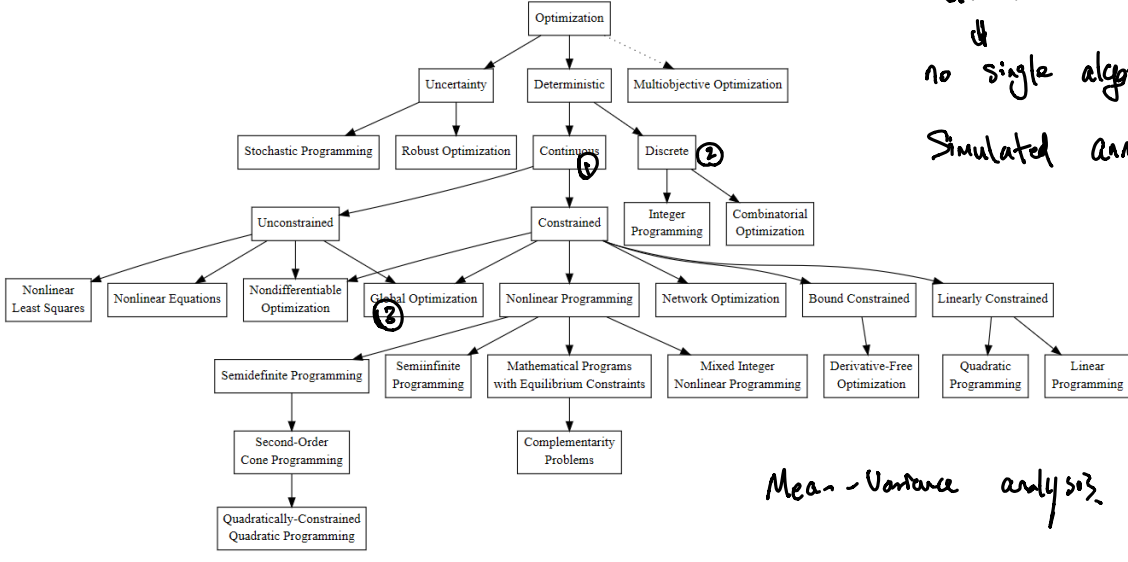
in the dual equation. Th one Lagrange multiplier in each iteration, which makes it easier than SMO to remove outliers. ISDA selects the worst KKT violator among all the αn and αn* values as the working set to be updated.

Regression Trees

- Trees can also be used for regression
- The main difference with classification tree is that the response variable is numerical (real-valued).
- The predicted value is an average over all the values in the partition belonging to a leaf (terminal node)
- The splitting criterion for regression split is minimization of RMSE
- We can apply this criterion iteratively as long as the drop in RMSE due to a split is greater than a chosen threshold
- Another possibility is to build the tree for as long as RMSE drops, and then do a pruning to get rid of the branches that contribute little to the precision

RMSE: root of MSE

Optimization



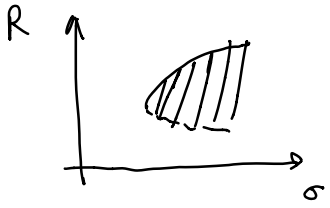
① eg. newtons method

② Traveling salesman problem.

③ Global and local optimization
⇓
no single algorithm for that

Simulated annealing

Mean-Variance analysis



- this doesn't really work in real-life because it is unstable. In order to overcome this problem, we can do things like: risk-parity.