# High-dimensional, massive sample-size Cox PH regression for survival analysis

1. Introduction

* Previously: survival analyses limited to applications with only a handful of predictors and a few hundreds or thousands of observations
* Recent advances in data acquisition techniques & access to high computation power -> larger data for analysis
* Large-scale applications: medical adverse event monitoring, longitudinal clinical trials, business data mining tasks
* Require methods for analyzing high dimensional, massive sample-size (HDMSS) data in a survival analysis framework
* Cox model & regularized Cox models
* Works well for small-scale problems >< do not scale well to HDMSS data due to use of costly Newton-Raphson iterations that require inverting large matrices
* Possible workaround and approx. -> large estimated coef variances, numerical ill-conditioning and poor predictive accuracy / calibration
* To solve the optimization problem -> regularized Cox survival modeling that scales for HDMSS data
* Exploit a variation of the cyclic coordinate descent optimization technique
* Avoids overfitting, provide improved predictive performance, efficient during fitting and prediction time

1. Related work (see paper for more info)
2. Regularized Cox survival analysis

* Assuming typical survival analysis setting
* n = **number of individuals** in the training data
* **Survival times** where
* : **indicator variable** such that
* : **p-vector of covariates** for individual i
* Assume that are conditionally independent given and censoring is non-informative
* **Observe data**:
* : p-vector of **unknown, underlying model parameters**
* Assume that **survival times** arise in an **independent and identically distributed fashion from density and survival functions** and parameterized by
* **Likelihood**  of the parametric model:

(3.1)

* **Cox proportional hazard model – semi-parametric hazard function** of the form:

(3.2)

* Similarly, **survival function** unfolds as:

(3.3)

* **Through (3.2) & (3.3), (3.1) falls out as:**

(3.4)

* In the absence of explicit specification of the baseline hazard -> hard to work with directly.
* Alternatively, Cox proposes to **maximize the partial likelihood function**

(3.5)

Where represents the risk set of the ith observation:

Note: the above expression **assumes that there are no tied survival times**. For HDMSS data, it sufficies to break the ties by adding a small random quantity (uniform between ) to the event times.

* One can then estimate through **the joint penalized partial likelihood** , **by assuming a penalty** for that shrinks the components of towards 0.

1. **The L2 penalty and ridge regression**

* For the L2 penalty, we have

(3.6)

* The regularization or tuning parameters are positive constants that control the degree of regularization and we choose them through cross-validation.
* Smaller values of -> stronger shrinkage of toward 0
* Absent further knowledge, we typically assume
* Ridge regression -> typically does not result in a sparse solution

1. **The L1 penalty and lasso regression**

* For L1 penalty, we have:

(3.7)

* represent a vector of regularization / tuning params
* Absent prior knowledge, we assume and select a value using cross-validation
* Lasso regression -> typically result in sparse solution

1. **Finding the param estimates**

* The **penalized partial likelihood of in the L2 case** can be written as:

(4.1)

* **Maximizing**  is equivalent to maximizing

(4.2)

Where the last negated sum is the penalty term.

* **For the L1 case, partial likelihood :**
* For both L1 and L2 regularization, their respective negated log-penalized partial likelihoods are log-convex and a wide range of optimization algorithms can be utilized.

>< Due to high dimensionality, usual methods like Newton-Raphson are not feasible owing to their high memory requirements and numerical instability

* Alternate optimization approach: **column relaxation with logistic loss (CLG) algorithm** (Zhang and Oles, 2000)
* A type of cyclic coordinate descent algorithm – provides the favorable property of scaling to HD data w/ ease of implementation
* **Cyclic coordinate descent algorithm:**
* Setting all variables to some initial value
* Holding all other variables constant, it then **solves a 1D optimization problem** to set the first variable to a value that minimizes or drives downhill the objective function
* Then find the minimizing value of a second variable, while holding all other constant
* Then so on for other variables
* When all variables have been traversed, the algorithm returns to the first variable and starts again
* Multiple passes are made over the variables until some convergence criterion is met
* CLG method relies on 1D updates -> **does not need to compute, store, or invert an HD Hessian matrix**
* Details of the algorithm for **minimizing the negated log-penalized partial likelihood for L1 and L2 penalties**
* Using the CLG algorithm, the 1D optimization problem involves **finding that minimizes ,** assuming that the other are held at their current values
* Using (4.2) in L2 case (and ignoring constants log ), finding is equivalent to finding the z that minimizes:

(4.3)

* **The classic Newton method approximates the** **objective function g(.)** by the first 3 terms of its Taylor series at the current

(4.4)

Where:

(4.5)

(4.6)

* **Value of** for both types of penalties can then be computed as:

(4.7)

1. Efficient computation and storage

* Marked efficiency in computing and storing the inner products via the low-rank update for all i