# Machine Learning for Survival Analysis

**Chandan K. Reddy**

Dept. of Computer Science

Virginia Tech

http://www.cs.vt.edu/~reddy

**Yan Li**

Dept. of Computational Medicine and Bioinformatics

Univ. of Michigan, Ann Arbor

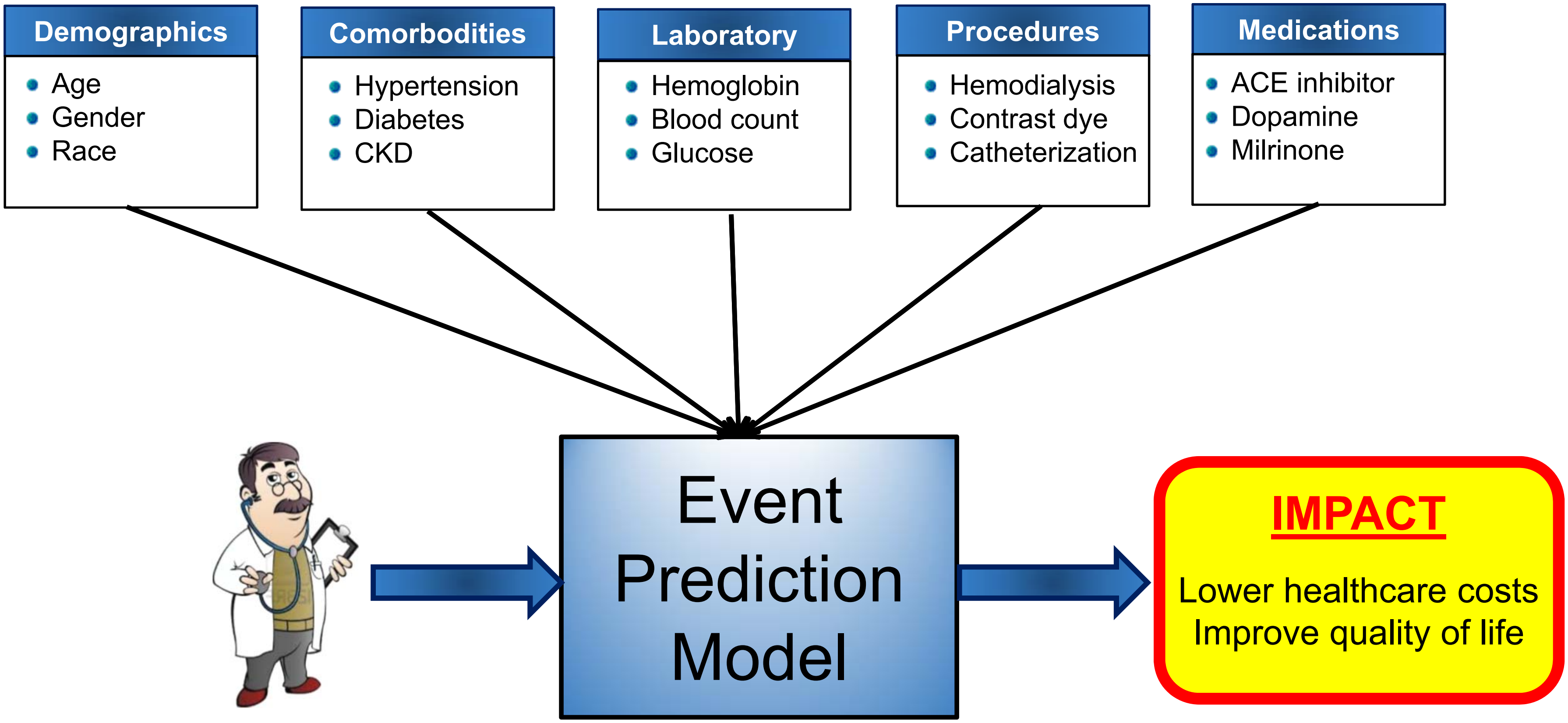
# Tutorial Outline

- Basic Concepts
- Statistical Methods
- Machine Learning Methods
- Related Topics

# Tutorial Outline

- **Basic Concepts**
- Statistical Methods
- Machine Learning Methods
- Related Topics

# Healthcare



**Demographics**
- Age
- Gender
- Race

**Comorbodities**
- Hypertension
- Diabetes
- CKD

**Laboratory**
- Hemoglobin
- Blood count
- Glucose

**Procedures**
- Hemodialysis
- Contrast dye
- Catheterization

**Medications**
- ACE inhibitor
- Dopamine
- Milrinone

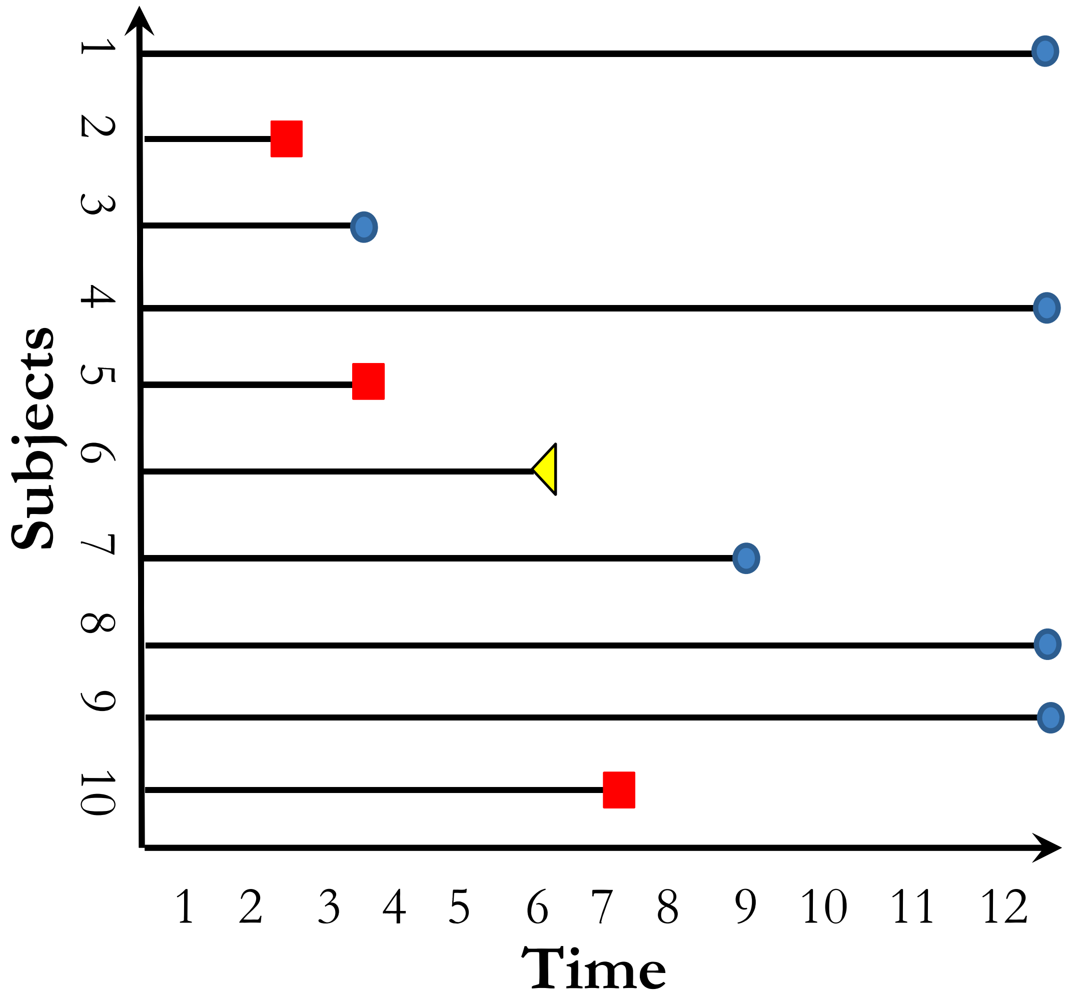
Event Prediction Model

**IMPACT**

Lower healthcare costs
Improve quality of life

- **Event of Interest :** Rehospitalization; Disease recurrence; Cancer survival
- **Outcome:** Likelihood of hospitalization within t days of discharge

# Mining Events in Longitudinal Data



**Classification Problem:**

3 +ve and 7 -ve

Cannot predict the time of event

Need to re-train for each time

**Regression Problem:**

Can predict the time of event

Only 3 samples (not 10)

– loss of data

Ping Wang, Yan Li, Chandan, K. Reddy, "**Machine Learning for Survival Analysis: A Survey**". ACM Computing Surveys (under revision), 2017.

# Problem Statement

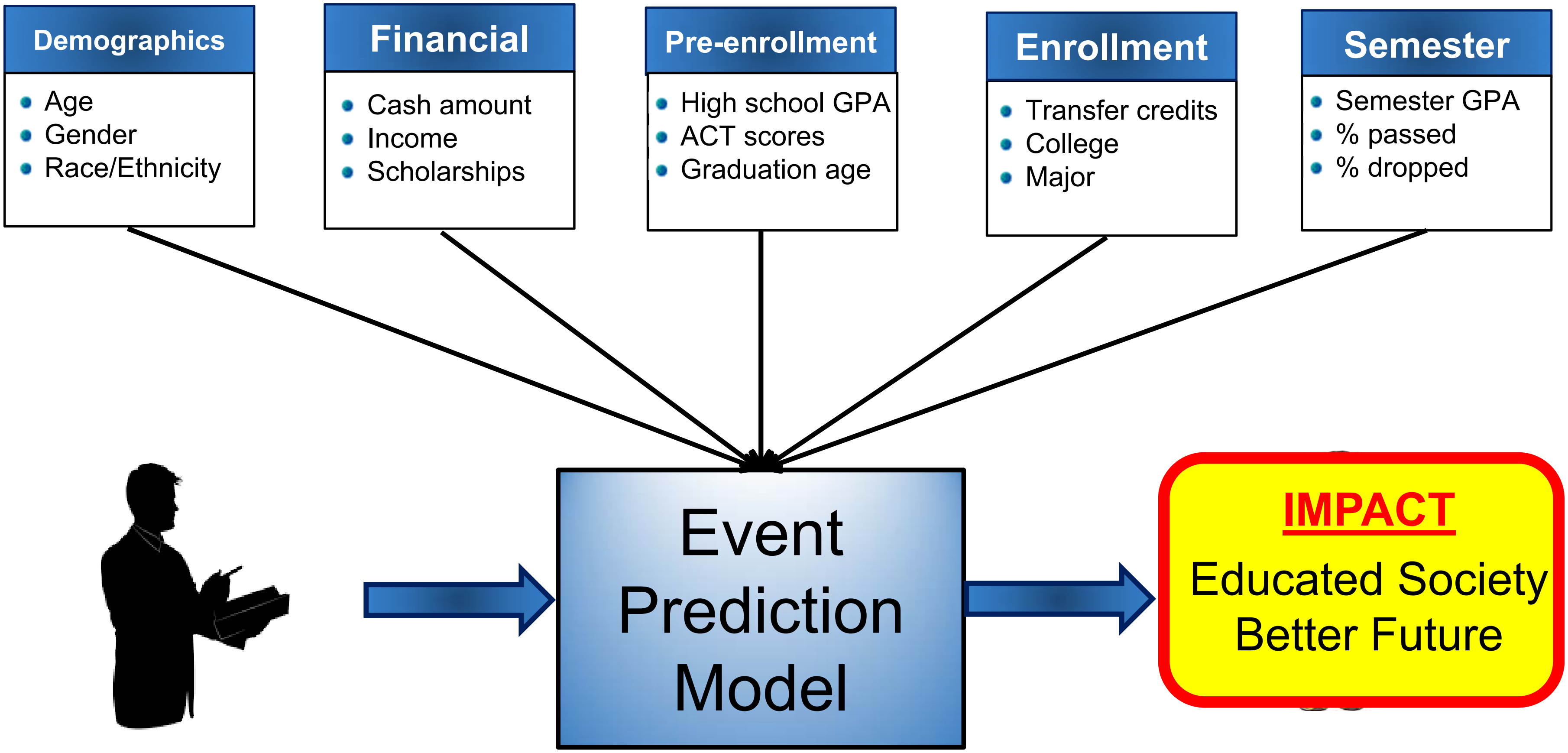
- For a given instance $i$, represented by a triplet $(X_i, y_i, \delta_i)$.
  - $X_i$ is the feature vector;
  - $\delta_i$ is the binary event indicator, i.e., $\delta_i = 1$ for an uncensored instance and $\delta_i = 0$ for a censored instance;
  - $y_i$ denotes the observed time and is equal to the survival time for an uncensored instance and for a censored instance, i.e.,

$$y_i = \begin{cases} T_i & \delta_i = 1 \\ C_i & \delta_i = 0 \end{cases}$$

- Note for $T_i$:
  - The value of $T_i$ will be both non-negative and continuous.
  - $T_i$ is latent for censored instances.

**Goal of survival analysis**: To estimate the time to the event of interest $T_j$ for a new instance $j$ with feature predictors denoted by $X_j$.

# Education



**Demographics**

- Age
- Gender
- Race/Ethnicity

**Financial**

- Cash amount
- Income
- Scholarships

**Pre-enrollment**

- High school GPA
- ACT scores
- Graduation age

**Enrollment**

- Transfer credits
- College
- Major

**Semester**

- Semester GPA
- % passed
- % dropped

Event Prediction Model

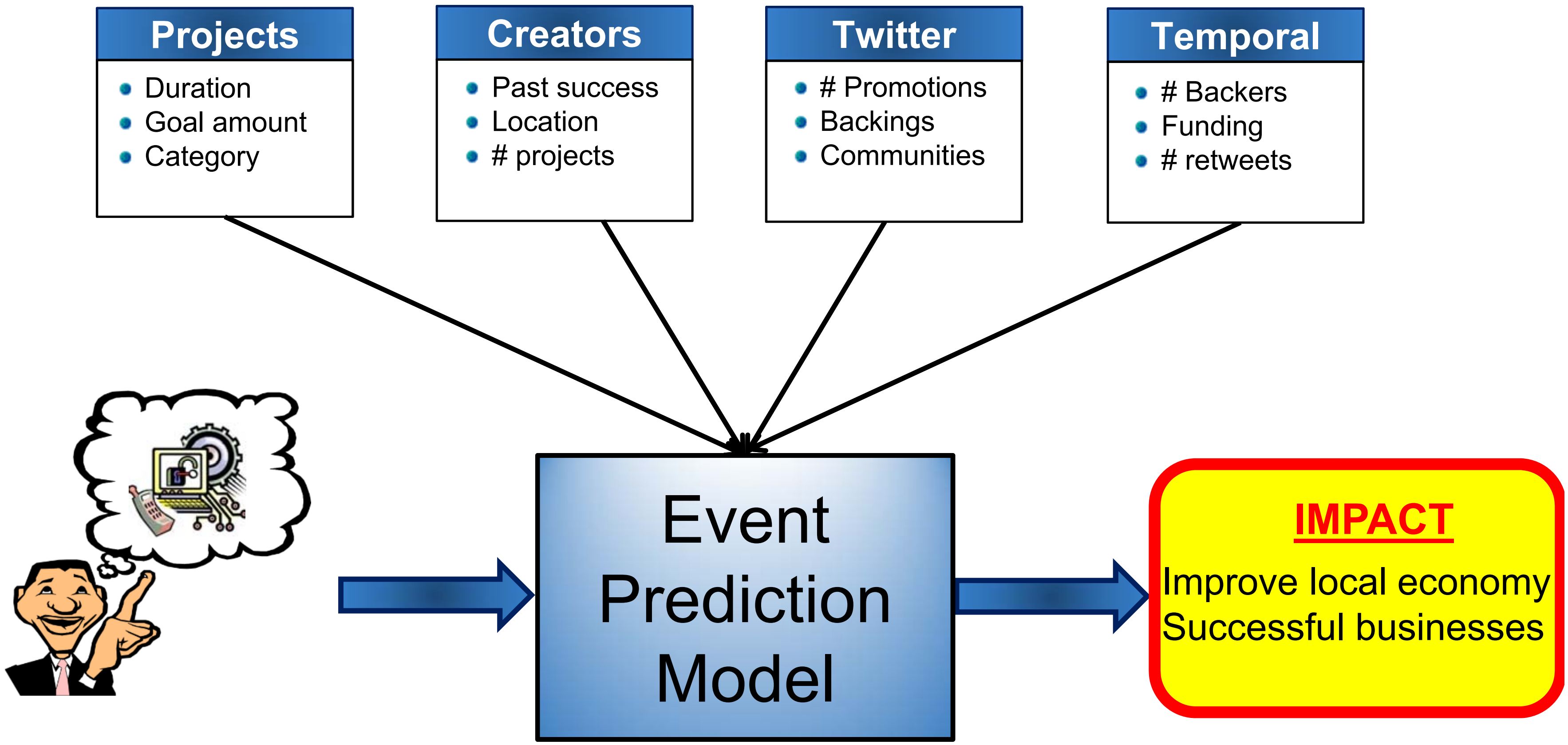
**IMPACT**

Educated Society Better Future

- **Event of Interest :** Student dropout
- **Outcome:** Likelihood of a student being dropout within t days

S. Ameri, M. J. Fard, R. B. Chinnam and C. K. Reddy, "**Survival Analysis based Framework for Early Prediction of Student Dropouts**", *CIKM* 2016.

# Crowdfunding



**Projects**
- Duration
- Goal amount
- Category

**Creators**
- Past success
- Location
- # projects

**Twitter**
- # Promotions
- Backings
- Communities

**Temporal**
- # Backers
- Funding
- # retweets

Event Prediction Model

**IMPACT**

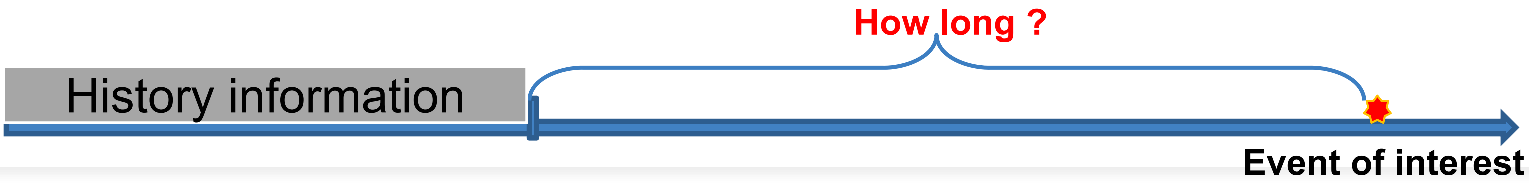
Improve local economy
Successful businesses

- **Event of Interest:** Project Success
- **Outcome:** Likelihood of a project being successful within t days

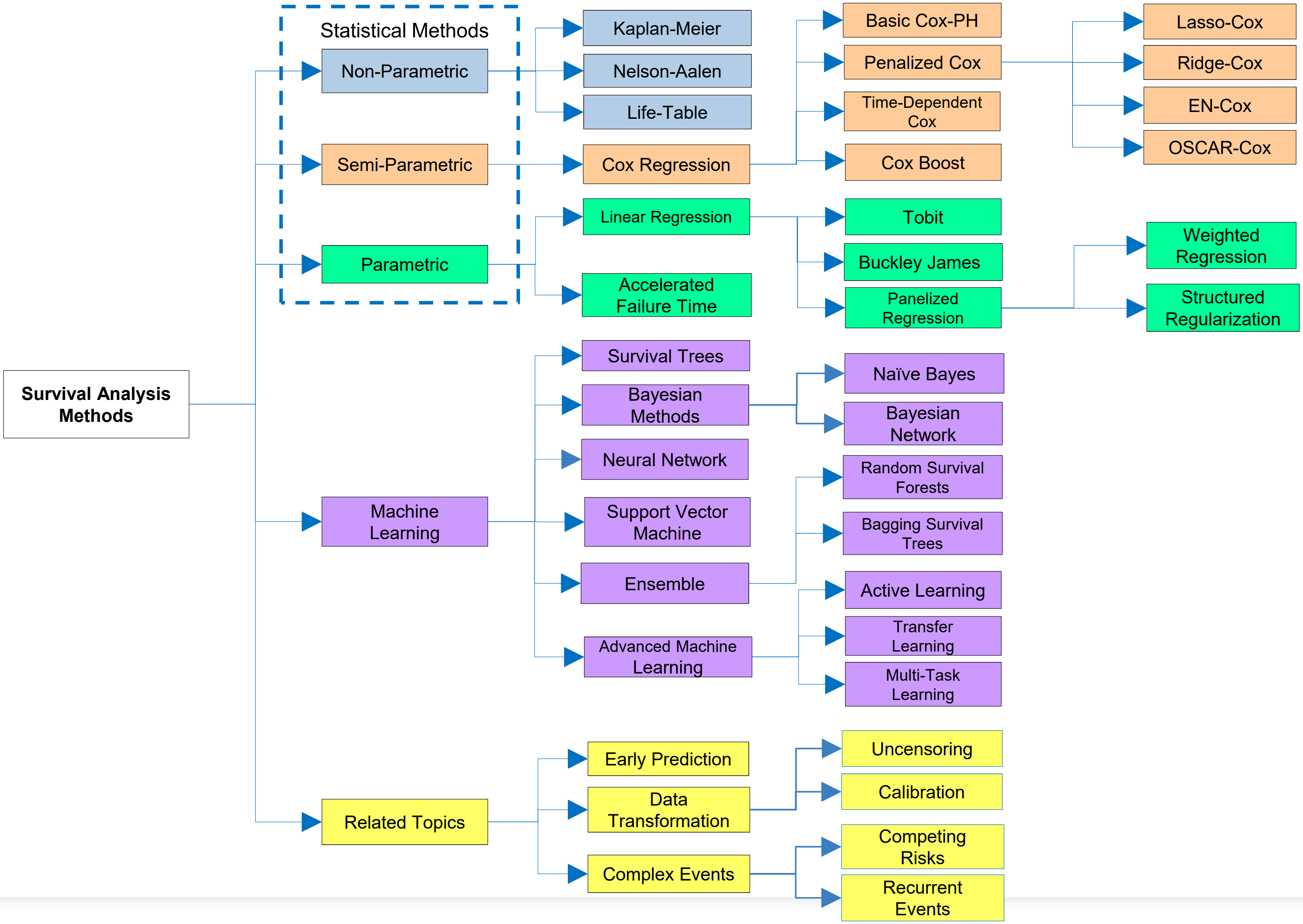
Y. Li, V. Rakesh, and C. K. Reddy, "**Project Success Prediction in Crowdfunding Environments**", *WSDM* 2016.

# Other Applications

- **Reliability**: Device Failure Modeling in Engineering
  - **Goal:** Estimate when a device will fail
  - **Features:** Product and manufacturer details, user reviews
- **Duration Modeling**: Unemployment Duration in Economics
  - **Goal:** Estimate the time people spend without a job (for getting a new job)
  - **Features:** User demographics and experience, Job details and economics
- **Click Through Rate**: Computational Advertising on the Web
  - **Goal:** Estimate when a web user will click the link of the ad.
  - **Features:** User and Ad information, website statistics
- **Customer Lifetime Value**: Targeted Marketing
  - **Goal:** Estimate the frequent purchase pattern for customers.
  - **Features:** Customer and store/product information.



# Taxonomy of Survival Analysis Methods



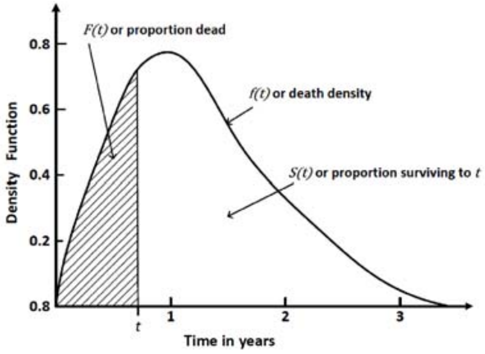
# Tutorial Outline

- Basic Concepts
- **Statistical Methods**
- Machine Learning Methods
- Related Topics

# Basics of Survival Analysis

- Main focuses is on time to event data. Typically, survival data are not fully observed, but rather are censored.
- Several important functions:
  - Survival function, indicating the probability that the instance can survive for longer than a certain time t.

    $$S(t) = \Pr(T \geq t)$$



  - Cumulative density function, representing the probability that the event of interest occurs earlier than t.

    $$F(t) = 1 - S(t)$$

Survival function

$$S(t) = \exp(-H(t))$$

  - Death density function:

    $$f(t) = dF(t)/dt = -dS(t)/dt$$

  - Hazard function: representing the probability the “event” of interest occurs in the next instant, given survival to time t.

    $$h(t) = \frac{f(t)}{S(t)} = -\frac{d[\ln S(t)]}{dt}$$

Cumulative hazard function

$$H(t) = \int_0^t h(u)du$$

Chandan K. Reddy and Yan Li, "**A Review of Clinical Prediction Models**", in *Healthcare Data Analytics*, Chandan K. Reddy and Charu C. Aggarwal (eds.), Chapman and Hall/CRC Press, 2015.

# Evaluation Metrics

- Due to the presence of the censoring in survival data, the standard evaluation metrics for regression such as root of mean squared error and $R^2$ are not suitable for measuring the performance in survival analysis.
- Three specialized evaluation metrics for survival analysis:
  - Concordance index (C-index)
  - Brier score
  - Mean absolute error

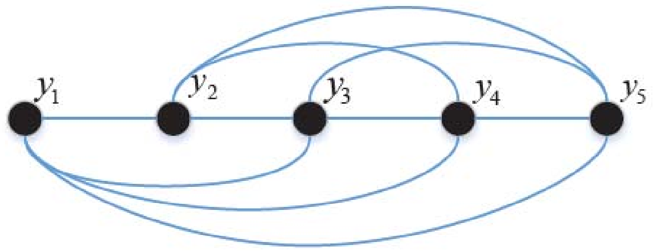
# Concordance Index (C-Index)

- It is a rank order statistic for predictions against true outcomes and is defined as the ratio of the concordant pairs to the total comparable pairs.
- Given the comparable instance pair $(i, j)$ with $t_i$ and $t_j$ are the actual observed times and S($t_i$) and S($t_j$) are the predicted survival times,
  - The pair $(i, j)$ is **concordant** if $t_i$ > $t_j$ and S($t_i$) > S($t_j$).
  - The pair $(i, j)$ is **discordant** if $t_i$ > $t_j$ and S($t_i$) < S($t_j$).
- Then, the concordance probability $c = \Pr\left(\hat{T}_i < \hat{T}_j \middle| T_i < T_j\right)$ measures the concordance between the rankings of actual values and predicted values.
- For a binary outcome, C-index is identical to the area under the ROC curve (AUC).

U. Hajime, et al. "On the C-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data." Statistics in medicine, 2011.

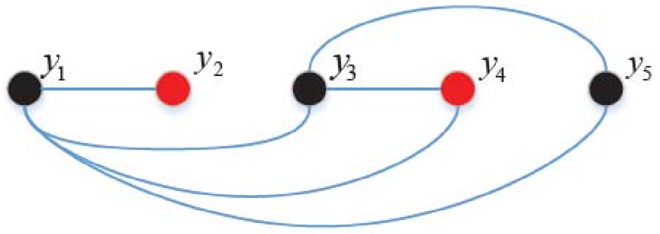
# Comparable Pairs

- The survival times of two instances can be compared if:
  - Both of them are uncensored;
  - The observed event time of the uncensored instance is smaller than the censoring time of the censored instance.



**Without Censoring**

A total of ${}^5C_2$ comparable pairs



**With Censoring**

Comparable only with events and with those censored after the events

H. Steck, B. Krishnapuram, C. Dehing-oberije, P. Lambin, and V. C. Raykar, “On ranking in survival analysis: Bounds on the concordance index”, *NIPS* 2008.

# C-index

- When the output of the model is the prediction of survival time:

$$\hat{c} = \frac{1}{num} \sum_{i:\delta_i=1} \sum_{j:\, y_i<y_j} I[S(\hat{y}_j|X_j) > S(\hat{y}_i|X_i)]$$

Where $S(\hat{y}_i|X_i)$ is the predicted survival probabilities, $num$ denotes the total number of comparable pairs.

- When the output of the model is the hazard ratio (Cox model):

$$\hat{c} = \frac{1}{num} \sum_{i:\delta_i=1} \sum_{j:\, y_i<y_j} I[X_i\hat{\beta} > X_j\hat{\beta}]$$

Where $I[\cdot]$ is the indicator function and $\hat{\beta}$ is the estimated parameters from the Cox based models. (The patient who has a longer survival time should have a smaller hazard ratio).

# C-index during a Time Period

- Area under the ROC curves (AUC) is

$$AUC = Pr\left(\hat{y}_i < \hat{y}_j \middle| y_i = 0, y_j = 1\right) = \frac{1}{num}\sum_{y_i=0}\sum_{y_j=1} I(\hat{y}_i < \hat{y}_j)$$

- In a possible survival time $t \in T_s$, $T_s$ is the set of all possible survival times, the time-specific AUC is defined as

$$AUC(t) = Pr\left(\hat{y}_i < \hat{y}_j \middle| y_i < t, y_j > t\right) = \frac{1}{num(t)}\sum_{i:\, y_i<t}\sum_{j:\, y_j>t} I(\hat{y}_i < \hat{y}_j)$$

$num(t)$ denotes the number of comparable pairs at time $t$.

- Then the C-index during a time period $(0, t^*)$ can be calculated as:

$$\boldsymbol{c_{t^*}} = \frac{\mathbf{1}}{\boldsymbol{num}}\sum_{i:\delta_i=1}\sum_{\boldsymbol{T_i<T_j}} \boldsymbol{I}\,(\hat{\boldsymbol{y}}_{\boldsymbol{i}} < \hat{\boldsymbol{y}}_{\boldsymbol{j}})$$

$$= \frac{\mathbf{1}}{\sum_{\boldsymbol{t\in T_s}} \boldsymbol{num(t)}}\sum_{\boldsymbol{t\in T_s}}\sum_{\boldsymbol{y_i<t}}\sum_{\boldsymbol{y_j>t}} \boldsymbol{I}(\hat{\boldsymbol{y}}_{\boldsymbol{i}} < \hat{\boldsymbol{y}}_{\boldsymbol{j}}) = \sum_{\boldsymbol{t\in T_s}} \boldsymbol{AUC(t)} \cdot \frac{\boldsymbol{num(t)}}{\boldsymbol{num}}$$

- C-index is a weighted average of the area under time-specific ROC curves (Time-dependent AUC).

# Brier Score

- Brier score is used to evaluate the prediction models where the outcome to be predicted is either binary or categorical in nature.
- The individual contributions to the empirical Brier score are reweighted based on the censoring information:

$$BS(t) = \frac{1}{N}\sum_{i=1}^{N} w_i(t)[\hat{y}_i(t) - y_i(t)]^2$$

$w_i(t)$ denotes the weight for the $i^{th}$ instance.

- The weights can be estimated by considering the Kaplan-Meier estimator of the censoring distribution $G$ on the dataset.

$$w_i(t) = \begin{cases} \delta_i / G(y_i) & if\ y_i \leq t \\ 1/G(y_i) & if\ y_i > t \end{cases}$$

  - The weights for the instances that are censored before $t$ will be $0$.
  - The weights for the instances that are uncensored at $t$ are greater than $1$.

E. Graf, C. Schmoor, W. Sauerbrei, and M. Schumacher, “Assessment and comparison of prognostic classification schemes for survival data”, Statistics in medicine, 1999.

# Mean Absolute Error

- For survival analysis problems, the mean absolute error (MAE) can be defined as an average of the differences between the predicted time values and the actual observation time values.

$$MAE = \frac{1}{n}\sum_{i=1}^{N}(\delta_i|y_i - \hat{y}_i|)$$

  where

  - $y_i$ -- the actual observation times.
  - $\hat{y}_i$ -- the predicted times.

- Only the samples for which the event occurs are being considered in this metric.
- Condition: MAE can only be used for the evaluation of survival models which can provide the event time as the predicted target value.

# Summary of Statistical methods

| Type | Advantages | Disadvantages | Specific methods |
|---|---|---|---|
| **Non-parametric** | More efficient when no suitable theoretical distributions known. | Difficult to interpret; yields inaccurate estimates. | Kaplan-Meier<br>Nelson-Aalen<br>Life-Table |
| **Semi-parametric** | The knowledge of the underlying distribution of survival times is not required. | The distribution of the outcome is unknown; not easy to interpret. | Cox model<br>Regularized Cox<br>CoxBoost<br>Time-Dependent Cox |
| **Parametric** | Easy to interpret, more efficient and accurate when the survival times follow a particular distribution. | When the distribution assumption is violated, it may be inconsistent and can give sub-optimal results. | Tobit<br>Buckley-James<br>Penalized regression<br>Accelerated Failure Time |

# Kaplan-Meier Analysis

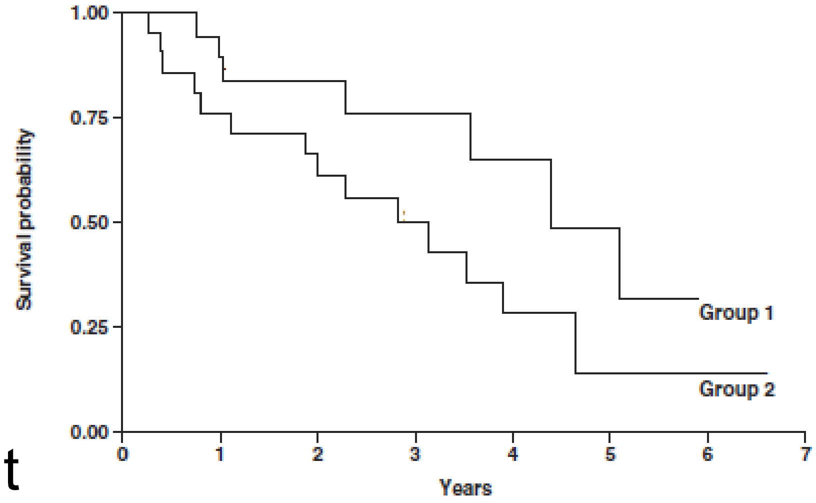
- Kaplan-Meier (KM) analysis is a nonparametric approach to survival outcomes. The survival function is:

$$\hat{S}(t) = \prod_{j:\, T_j<t} (1 - \frac{d_j}{r_j})$$

where

- $T_i \ldots T_K$ -- a set of distinct event times observed in the sample.
- $d_j$ -- number of events at $T_j$.
- $c_j$ -- number of censored observations between $T_j$ and $T_{j+1}$.
- $r_j$ -- number of individuals “at risk” right before the $j^{th}$ death.

$$r_j = r_{j-1} - d_{j-1} - c_{j-1}$$



E. Bradley. "Logistic regression, survival analysis, and the Kaplan-Meier curve." *JASA* 1988.

# Survival Outcomes

| Patient | Days | Status | Patient | Days | Status | Patient | Days | Status |
|---|---|---|---|---|---|---|---|---|
| 1 | 21 | 1 | 15 | 256 | 2 | 29 | 398 | 1 |
| 2 | 39 | 1 | 16 | 260 | 1 | 30 | 414 | 1 |
| 3 | 77 | 1 | 17 | 261 | 1 | 31 | 420 | 1 |
| 4 | 133 | 1 | 18 | 266 | 1 | 32 | 468 | 2 |
| 5 | 141 | 2 | 19 | 269 | 1 | 33 | 483 | 1 |
| 6 | 152 | 1 | 20 | 287 | 3 | 34 | 489 | 1 |
| 7 | 153 | 1 | 21 | 295 | 1 | 35 | 505 | 1 |
| 8 | 161 | 1 | 22 | 308 | 1 | 36 | 539 | 1 |
| 9 | 179 | 1 | 23 | 311 | 1 | 37 | 565 | 3 |
| 10 | 184 | 1 | 24 | 321 | 2 | 38 | 618 | 1 |
| 11 | 197 | 1 | 25 | 326 | 1 | 39 | 793 | 1 |
| 12 | 199 | 1 | 26 | 355 | 1 | 40 | 794 | 1 |
| 13 | 214 | 1 | 27 | 361 | 1 | | | |
| 14 | 228 | 1 | 28 | 374 | 1 | | | |

**Status**

1: Death

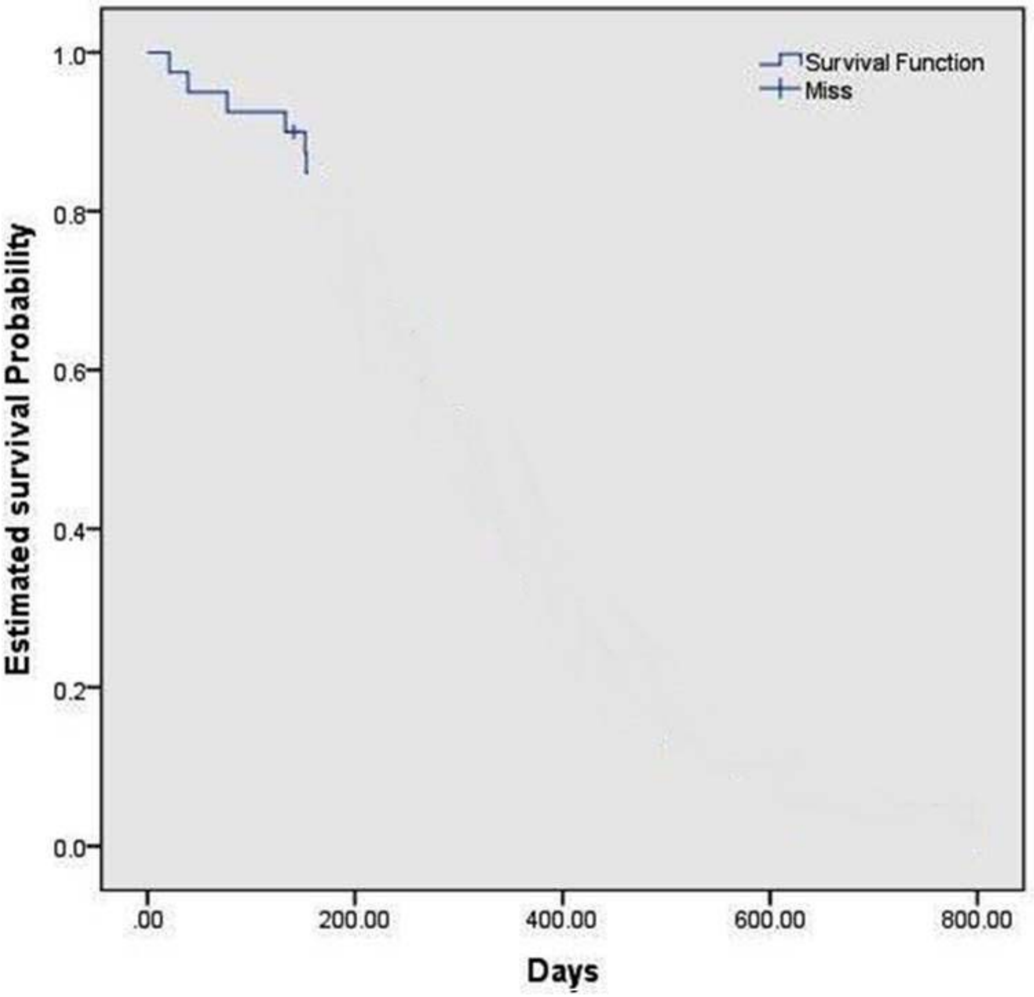
2: Lost to follow up

3: Withdrawn Alive

# Kaplan-Meier Analysis

Kaplan-Meier Analysis

| $i$ | Time | Status | $d_i$ | $c_i$ | $r_i$ | $S(t)$ |
|---|---|---|---|---|---|---|
| 1 | 21 | 1 | 1 | 0 | 40 | 0.975 |
| 2 | 39 | 1 | 1 | 0 | 39 | 0.95 |
| 3 | 77 | 1 | 1 | 0 | 38 | 0.925 |
| 4 | 133 | 1 | 1 | 0 | 37 | 0.9 |
| 5 | 141 | 2 | 0 | 1 | 36 | . |
| 6 | 152 | 1 | 1 | 0 | 35 | 0.874 |
| 7 | 153 | 1 | 1 | 0 | 34 | 0.849 |



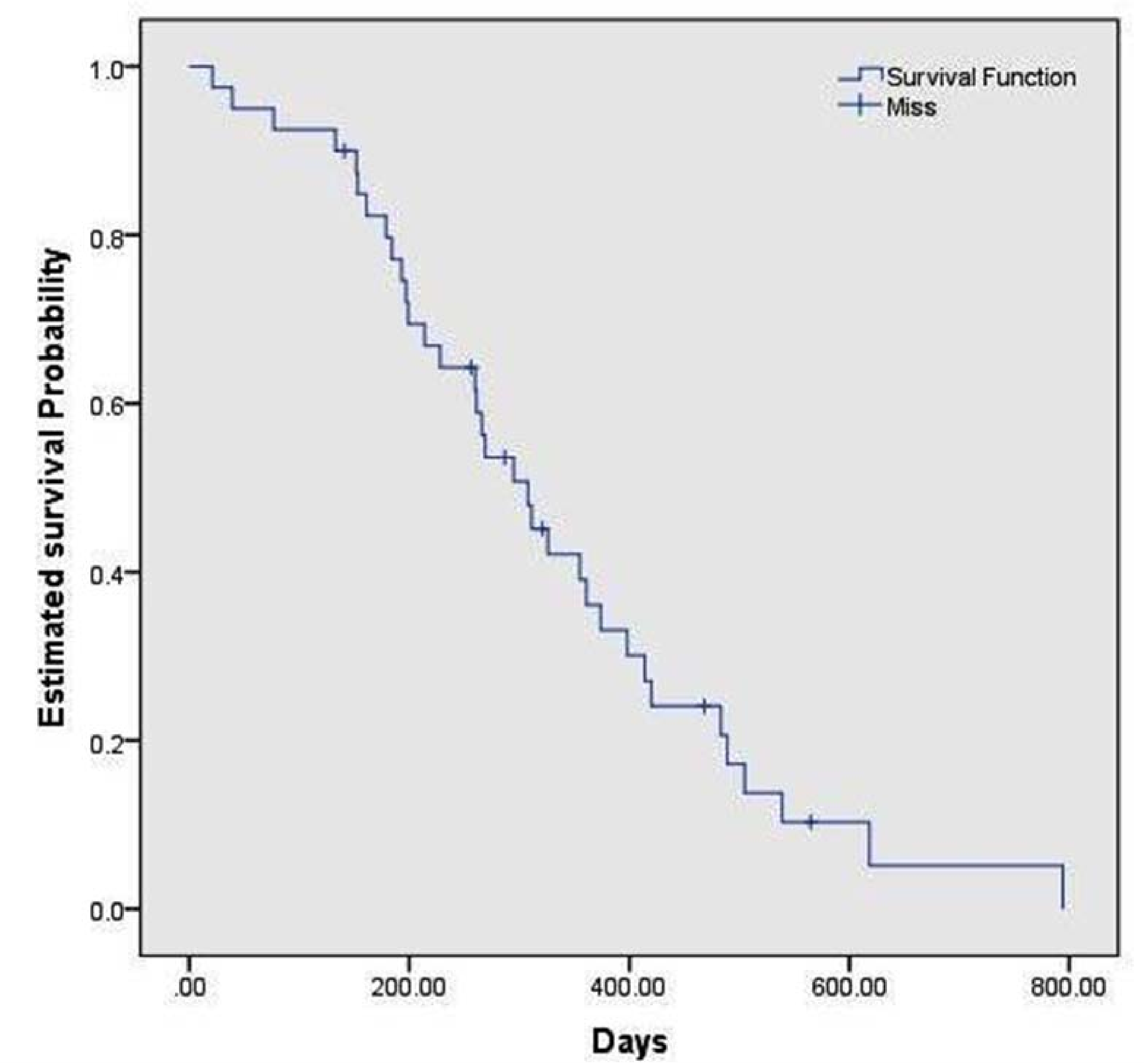
**KM Estimator:**

$$\hat{S}(t) = \prod_{j:T_j<t}(1-\frac{d_j}{r_j})$$

# Kaplan-Meier Analysis

KM Estimator:

| $i$ | Time | Statu | rror | $\sum d_i$ | $r_i$ |
|---|---|---|---|---|---|
| 1 | 21 | 1 | | 18 | 20 |
| 2 | 39 | 1 | 81 | 19 | 19 |
| 3 | 77 | 1 | 81 | 20 | 18 |
| 4 | 133 | 1 | 81 | 21 | 17 |
| 5 | 141 | 2 | | 21 | 16 |
| 6 | 152 | 1 | 81 | 22 | 15 |
| 7 | 153 | 1 | 81 | 23 | 14 |
| 8 | 161 | 1 | )8 | 24 | 13 |
| 9 | 179 | 1 | 79 | 25 | 12 |
| 10 | 184 | 1 | 77 | 26 | 11 |
| 11 | 193 | 1 | 75 | 27 | 10 |
| 12 | 197 | 1 | 72 | 28 | 9 |
| 13 | 199 | 1 | | 28 | 8 |
| 14 | 214 | 1 | )7 | 29 | 7 |
| 15 | 228 | 1 | 66 | 30 | 6 |
| 16 | 256 | 2 | 61 | 31 | 5 |
| 17 | 260 | 1 | 55 | 32 | 4 |
| 18 | 261 | 1 | | 32 | 3 |
| 19 | 266 | 1 | 46 | 33 | 2 |
| 20 | 269 | 1 | | 34 | 1 |



# Nelson-Aalen Estimator

- Nelson-Aalen estimator  is a non-parametric estimator of the cumulative hazard function (CHF) for censored data.
- Instead of estimating the survival probability as done in KM estimator, NA estimator directly estimates the hazard probability.
- The Nelson-Aalen estimator of the cumulative hazard function:

$$\widehat{H}(t) = \sum_{t_j \leq t} \frac{d_j}{r_j}$$

  - $d_j$ -- the number of deaths at time $t_j$
  - $r_j$ -- the number of individuals at risk at $t_j$

- The cumulative hazard rate function can be used to estimate the survival function and its variance.

$$\hat{S}(t) = e^{-\widehat{H}(t)} = \exp\left[-\sum_{t_j \leq t} \frac{d_j}{j}\right]$$

- The NA and KM estimators are asymptotically equivalent.

W. Nelson. “Theory and applications of hazard plotting for censored failure data.” Technometrics, 1972.
O. Aalen. “Nonparametric inference for a family of counting processes.” The Annals of Statistics, 1978.

# Clinical Life Tables

- Clinical life tables applies to grouped survival data from studies in patients with specific diseases, it focuses more on the conditional probability of dying within the interval.

**The $j^{th}$ time interval is $[t_{j-1}, t_j)$ VS.**
**$T_i \dots T_M$ is a set of distinct death times**

The survival function is:

$$\hat{S}(t_j) = \prod\nolimits_{\iota<j}\left(1 - \frac{d_\iota}{r'_\iota}\right)$$

**Nonparametric**

Assumption:

- at the beginning of each interval: $r'_j = r_j - c_j$
- at the end of each interval: $r'_j = r_j$
- on average halfway through the interval: $r'_j = r_j - c_j/2$

KM analysis suits small data set with a more accurate analysis,
Clinical life table suit for large data set with a relatively approximate result.

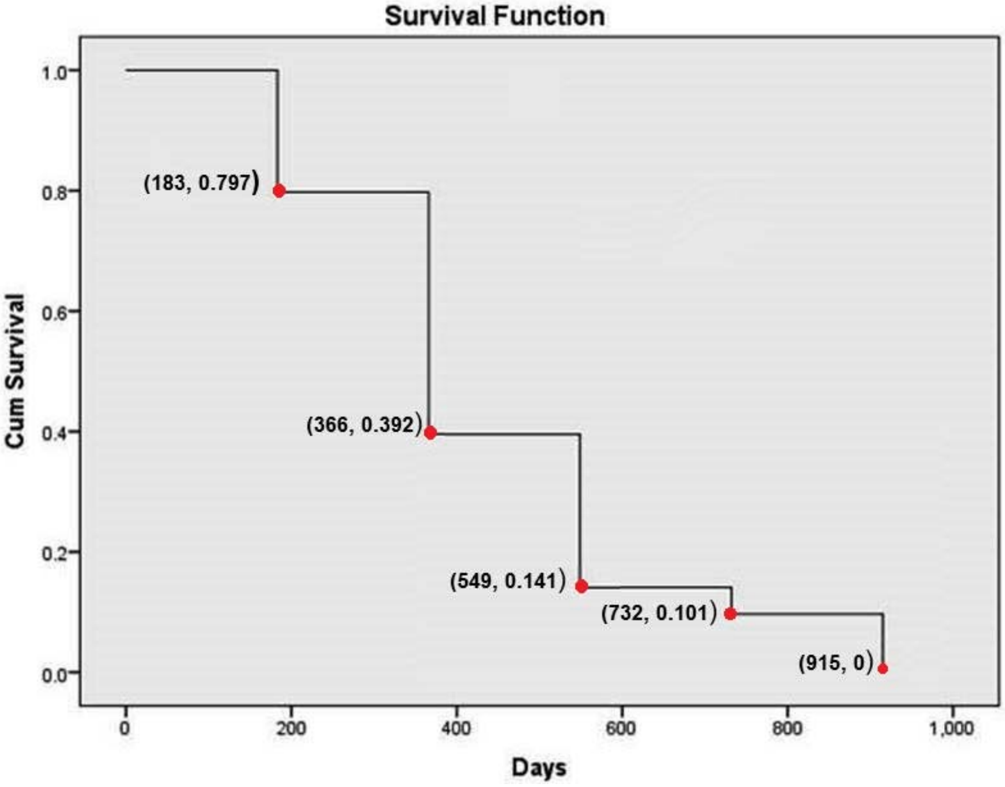
Cox, David R. "Regression models and life-tables", Journal of the Royal Statistical Society. Series B (Methodological), 1972.

# Clinical Life Tables

**NOTE** :
The length of interval is half year(183 days)

Clinical Life Table

| | Interval | Interval | | | | | | Std. Error |
|---|---|---|---|---|---|---|---|---|
| Interval | Start Time | End Time | $r_i$ | $c_i$ | $r_i'$ | $d_i$ | $S(t)$ | of $S(t)$ |
| 1 | 0 | 182 | 40 | 1 | 39.5 | 8 | 0.797 | 0.06 |
| 2 | 183 | 365 | 31 | 3 | 29.5 | 15 | 0.392 | 0.08 |
| 3 | 366 | 548 | 13 | 1 | 12.5 | 8 | 0.141 | 0.06 |
| 4 | 549 | 731 | 4 | 1 | 3.5 | 1 | 0.101 | 0.05 |
| 5 | 732 | 915 | 2 | 0 | 2 | 2 | 0 | 0 |



Clinical Life Table :

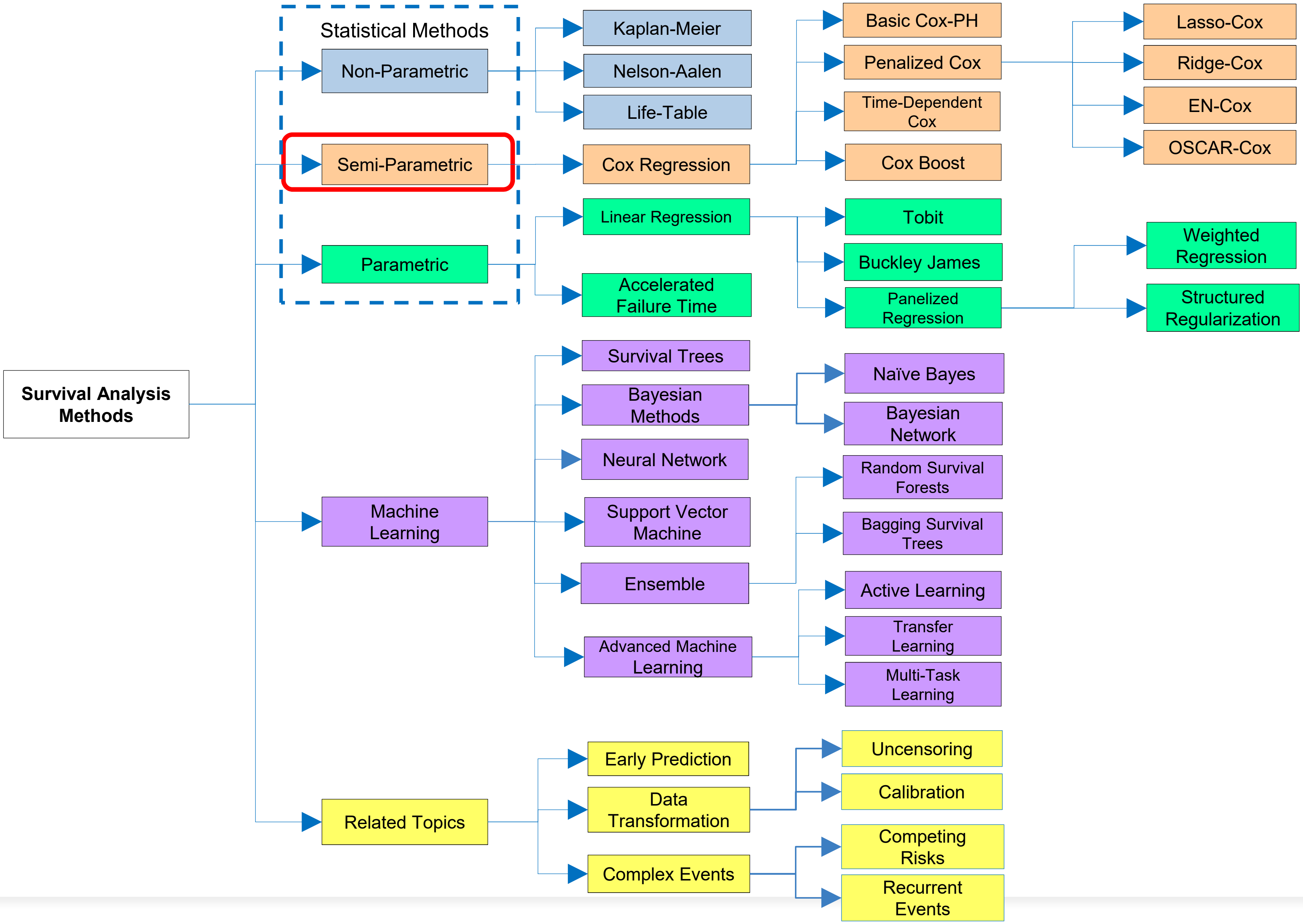
$$\hat{S}(t_j) = \prod_{\iota<j}(1 - \frac{d_\iota}{r_\iota'})$$

On average halfway through the interval: $r_j' = r_j - c_j/2$

# Statistical methods

| Type | Advantages | Disadvantages | Specific methods |
|---|---|---|---|
| **Non-parametric** | More efficient when no suitable theoretical distributions known. | Difficult to interpret; yields inaccurate estimates. | Kaplan-Meier<br>Nelson-Aalen<br>Life-Table |
| **Semi-parametric** | The knowledge of the underlying distribution of survival times is not required. | The distribution of the outcome is unknown; not easy to interpret. | Cox model<br>Regularized Cox<br>CoxBoost<br>Time-Dependent Cox |
| **Parametric** | Easy to interpret, more efficient and accurate when the survival times follow a particular distribution. | When the distribution assumption is violated, it may be inconsistent and can give sub-optimal results. | Tobit<br>Buckley-James<br>Penalized regression<br>Accelerated Failure Time |

# Taxonomy of Survival Analysis Methods



- **Survival Analysis Methods**
  - Statistical Methods
    - Non-Parametric
      - Kaplan-Meier
      - Nelson-Aalen
      - Life-Table
    - Semi-Parametric
      - Cox Regression
        - Basic Cox-PH
        - Penalized Cox
          - Lasso-Cox
          - Ridge-Cox
          - EN-Cox
          - OSCAR-Cox
        - Time-Dependent Cox
        - Cox Boost
    - Parametric
      - Linear Regression
        - Tobit
        - Buckley James
        - Panelized Regression
          - Weighted Regression
          - Structured Regularization
      - Accelerated Failure Time
  - Machine Learning
    - Survival Trees
    - Bayesian Methods
      - Naïve Bayes
      - Bayesian Network
    - Neural Network
    - Support Vector Machine
    - Ensemble
      - Random Survival Forests
      - Bagging Survival Trees
    - Advanced Machine Learning
      - Active Learning
      - Transfer Learning
      - Multi-Task Learning
  - Related Topics
    - Early Prediction
    - Data Transformation
      - Uncensoring
      - Calibration
    - Complex Events
      - Competing Risks
      - Recurrent Events

# Cox Proportional Hazards Model

- The Cox proportional hazards model is the most commonly used model in survival analysis.
- Hazard Function $h(t)$, sometimes called an instantaneous failure rate, shows the event rate at time $t$ conditional on survival until time $t$ or later.

$$h(t, X_i) = h_0(t)\exp(X_i\beta) \quad \Rightarrow \quad \log\left(\frac{h_i(t,X_i)}{h_0(t)}\right) = X_i\beta$$

A linear model for the log of the hazard ratio.

where

- $X_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ is the covariate vector.
- $h_0(t)$ is the baseline hazard function, which can be an arbitrary non-negative function of time.

- The Cox model is a semi-parametric algorithm since the baseline hazard function is unspecified.

D. R. Cox, "Regression models and life tables". Journal of the Royal Statistical Society, 1972.

# Cox Proportional Hazards Model

- The Proportional Hazards assumption means that the hazard ratio of two instances $X_1$ and $X_2$ is constant over time (independent of time).

$$\widehat{HR} = \frac{h(t, X_1)}{h(t, X_2)} = \frac{h_0(t)\exp(X_1\beta)}{h_0(t)\exp(X_2\beta)} = \exp[(X_1 - X_2)\beta]$$

- The survival function in Cox model can be computed as follows:

$$S(t) = \exp(-H_0(t)\exp(X\beta)) = S_0(t)^{\exp(X\beta)}$$

  - $H_0(t)$ is the cumulative baseline hazard function;
  - $S_0(t) = \exp\big(-H_0(t)\big)$ represents the baseline survival function.

- The Breslow's estimator is the most widely used method to estimate $H_0(t)$, which is given by:

$$\widehat{H}_0(t) = \sum_{t_i \le t} \hat{h}_0(t_i)$$

  - $\hat{h}_0(t_i) = \frac{1}{\sum_{j \in R_i} e^{X_j\beta}}$ if $t_i$ is an event time, otherwise $\hat{h}_0(t_i) = 0$.
  - $R_i$ represents the set of subjects who are at risk at time $t_i$.

# Optimization of Cox model

- Not possible to fit the model using the standard likelihood function
  - Reason: the baseline hazard function is not specified.
- Cox model uses partial likelihood function:
  - Advantage: depends only on the parameter of interest and is free of the nuisance parameters (baseline hazard).
- Conditional on the fact that the event occurs at $T_j$, the individual probability corresponding to covariate $X_j$ can be formulated as:

$$\frac{h\left(T_j, X_j\right)dt}{\sum_{i \in R_j} h\left(T_j, X_i\right)dt}$$

  - $J\ (J \leq N)$ -- the total number of events of interest that occurred during the observation period for $N$ instances.
  - $T_1 < T_2 < \cdots < T_J$ -- the distinct ordered time to event of interest.
  - $X_j$ -- the covariate vector for the subject who has the event at $T_j$.
  - $R_j$ -- the set of risk subjects at $T_j$.

# Partial Likelihood Function

- The partial likelihood function of the Cox model will be:

$$L(\beta) = \prod_{j=1}^{N} \left[ \frac{\exp(X_j\beta)}{\sum_{i\in R_j} \exp(X_i\beta)} \right]^{\delta_j}$$

  - If $\delta_j = 1$, the $j^{th}$ term in the product is the conditional probability;
  - if $\delta_j = 0$, the corresponding term is $1$, which means that the term will not have any effect on the final product.

- The coefficient vector is estimated by minimizing the negative log-partial likelihood:

$$LL(\beta) = -\sum_{j=1}^{N} \delta_j \left\{ X_j\beta - log \left[ \sum_{i\in R_j} \exp(X_i\beta) \right] \right\}$$

- The maximum partial likelihood estimator (MPLE) can be used along with the numerical Newton-Raphson method to iteratively find an estimator $\hat{\beta}$ which minimizes $LL(\beta)$.

D. R. Cox, Regression models and life tables, Journal of the Royal Statistical Society, 1972.

# Regularized Cox Models

- Regularized Cox regression methods:

$$\hat{\beta} = argmin_{\beta}\, LL(\beta) + \lambda * P(\beta)$$

$P(\beta)$ is a sparsity inducing norm and $\mu$ is the regularization parameter.

| Method | Penalty Term Formulation |
|---|---|
| LASSO | $\sum_{k=1}^{p} \lvert\beta_k\rvert$ |
| Ridge | $\sum_{k=1}^{p} {\beta_k}^2$ |
| Elastic Net (EN) | $\mu \sum_{k=1}^{p} \lvert\beta_k\rvert + (1-\mu) \sum_{k=1}^{p} {\beta_k}^2$ |
| Adaptive LASSO (AL) | $\sum_{k=1}^{p} w_k \lvert\beta_k\rvert$ |
| Adaptive Elastic Net (AEN) | $\mu \sum_{k=1}^{p} w_k \lvert\beta_k\rvert + (1-\mu) \sum_{k=1}^{p} {\beta_k}^2$ |
| OSCAR | $\lambda_1 \parallel \beta \parallel_1 + \lambda_2 \parallel T\beta \parallel_1$ |

Promotes Sparsity

Handles Correlation

Sparsity + Correlation

Adaptive Variants are slightly more effective

Sparsity + Feature Correlation Graph

# Lasso-Cox and Ridge-Cox

- Lasso performs feature selection and estimates the regression coefficients simultaneously using a $\ell_1$-norm regularizer .
- Lasso-Cox model incorporates the $\ell_1$-norm into the log-partial likelihood and inherits the properties of Lasso.
- Extensions of Lasso-Cox method:
  - Adaptive Lasso-Cox - adaptively weighted $\ell_1$-penalties on regression coefficients.
  - Fused Lasso-Cox - coefficients and their successive differences are penalized.
  - Graphical Lasso-Cox - $\ell_1$-penalty on the inverse covariance matrix is applied to estimate the sparse graphs .
- Ridge-Cox is Cox regression model regularized by a $\ell_2$-norm
  - Incorporates a $\ell_2$-norm regularizer to select the correlated features.
  - Shrink their values towards each other.

N. Simon et al., "Regularization paths for Coxs proportional hazards model via coordinate descent", *JSS* 2011.

# EN-Cox and OSCAR-Cox

- EN-Cox method uses the Elastic Net penalty term (combining the $\ell_1$ and squared $\ell_2$ penalties) into the log-partial likelihood function.
  - Performs feature selection and handles correlation between the features.
- Kernel Elastic Net Cox (KEN-Cox) method builds a kernel similarity matrix for the feature space to incorporate the pairwise feature similarity into the Cox model.
- OSCAR-Cox uses Octagonal Shrinkage and Clustering Algorithm for Regression regularizer within the Cox framework.

$$P(\beta) = \lambda_1 \parallel \beta \parallel_1 + \lambda_2 \parallel T\beta \parallel_1$$

  - $T$ is the sparse symmetric edge set matrix from a graph constructed by features.
  - Performs the variable selection for highly correlated features in regression.
  - Obtain equal coefficients for the features which relate to the outcome in similar ways.

B. Vinzamuri and C. K. Reddy, "Cox Regression with Correlation based Regularization for Electronic Health Records", *ICDM* 2013.

# CoxBoost

- CoxBoost method can be applied to fit the sparse survival models on the high-dimensional data by considers some mandatory covariates explicitly in the model.

CoxBoost **VS.** Regular gradient boosting approach (RGBA)

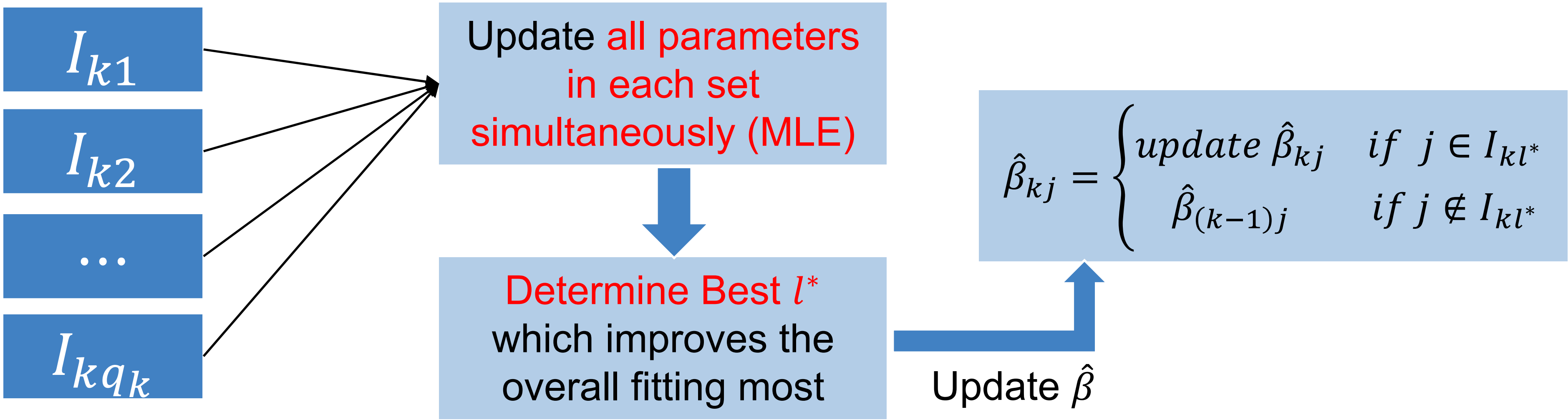
- **Similar goal:** estimate the coefficients in Cox model.
- **Differences:**
  - RGBA: updates in component-wise boosting or fits the gradient by using all covariates in each step.
  - CoxBoost: considers a flexible set of candidate variables for updating in each boosting step.

H. Binder and M. Schumacher, “Allowing for mandatory covariates in boosting estimation of sparse high-dimensional survival models”, BMC bioinformatics, 2008.

# CoxBoost

## How to update in each iteration of CoxBoost?

- Assume that $\hat{\beta}_{k-1} = \left(\hat{\beta}_{(k-1)1}, \cdots, \hat{\beta}_{(k-1)P}\right)^T$ being the actual estimate of the overall parameter vector $\beta$ after step $k-1$ of the algorithm and $q_k$ predefined candidate sets of features in step $k$ with $I_{kl} \subset \{1, \cdots, P\}$, $l = 1, \cdots, q_k$.



$I_{k1}$

$I_{k2}$

$\cdots$

$I_{kq_k}$

Update all parameters in each set simultaneously (MLE)

Determine Best $l^*$ which improves the overall fitting most

Update $\hat{\beta}$

$$\hat{\beta}_{kj} = \begin{cases} update\ \hat{\beta}_{kj} & if\ \ j \in I_{kl^*} \\ \hat{\beta}_{(k-1)j} & if\ j \notin I_{kl^*} \end{cases}$$

Special case:

Component-wise CoxBoost: $I_k = \{\{1\}, \cdots, \{P\}\}$ in each step $k$.

# TD-Cox Model

- Cox regression model is also effectively adapted to time-dependent Cox model to handle time-dependent covariates.
- Given a survival analysis problem which involves both time-dependent and time-independent features, the variables at time $t$ can be denoted as:

$$X(t) = (\underbrace{\boxed{X_{\cdot 1}(t), X_{\cdot 2}(t), \ldots, X_{\cdot P_1}(t)}}_{\text{Time-dependent}}, \underbrace{\boxed{X_{\cdot 1}, X_{\cdot 2}, \ldots, X_{\cdot P_2}}}_{\text{Time-independent}})$$

- The TD-Cox model can be formulated as:

$$h\big(t, \{X\}(t)\big) = h_0(t)\exp\left[\sum_{j=1}^{P_1} \underbrace{\alpha_j X_{\cdot j}(t)}_{\text{Time-dependent}} + \sum_{i=1}^{P_2} \underbrace{\beta_i X_{\cdot i}}_{\text{Time-independent}}\right]$$

# TD-Cox Model

- For the two sets of predictors at time $t$:

$$X_1(t) = (X_{11}(t), X_{12}(t), \dots, X_{1P_1}(t), X_{11}, X_{12}, \dots, X_{1P_2})$$

$$X_2(t) \ \ = (X_{21}(t), X_{22}(t), \dots, X_{2P_1}(t), X_{\cdot}^{*}{}_{1}, X_{\cdot}^{*}{}_{2}, \dots, X_{2P_2})$$

- The hazard ratio for TD-Cox model can be computed as follows:

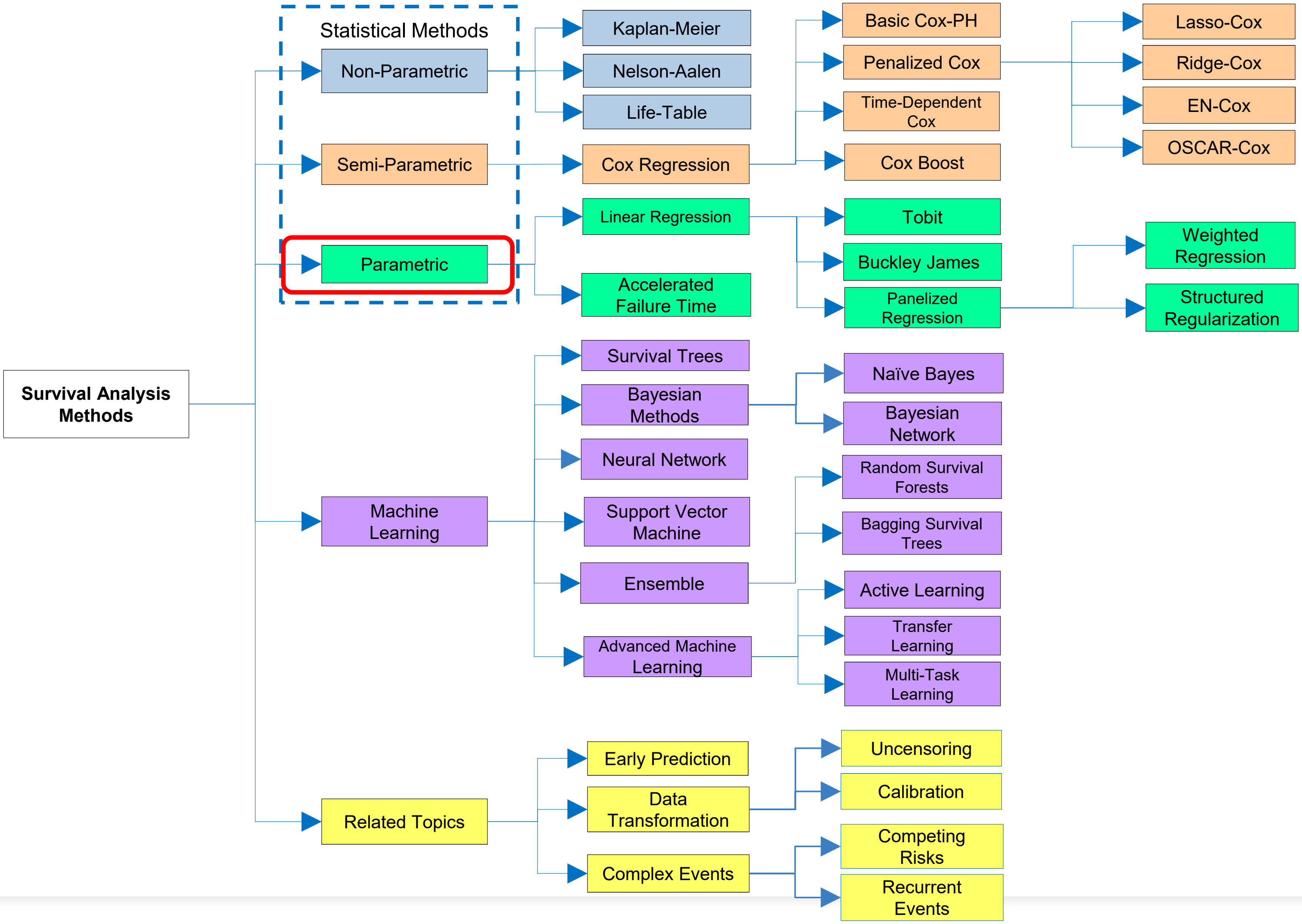
$$\widehat{HR}(t) = \frac{\hat{h}(t, X_2(t))}{\hat{h}(t, X_1(t))} = exp\boxed{\Bigg[\sum_{j=1}^{P_1} \alpha_j [X_{2j}(t) - X_{1j}(t)]} + \sum_{j=1}^{P_2} \beta_j [X_{2j} - X_{1j}]\Bigg]$$

  - Since the first component in the exponent is time-dependent, we can consider the hazard ratio in the TD-Cox model as a function of time $t$.
  - This means that it does not satisfy the PH assumption mentioned in the standard Cox model.

# Counting Process Example

| ID | Gender (0/1) | Weight (lb) | Smoke (0/1) | Start Time (days) | Stop Time (days) | Status |
|---|---|---|---|---|---|---|
| $S_1$ | 1 (F) | 125 | 0 | 0 | 20 | 1 |
| $S_2$ | 0 (M) | 171 | 1 | 0 | 20 | 0 |
| $S_2$ | 0 | 180 | 0 | 20 | 30 | 1 |
| $S_3$ | 0 | 165 | 1 | 0 | 20 | 0 |
| $S_3$ | 0 | 160 | 0 | 20 | 30 | 0 |
| $S_3$ | 0 | 168 | 0 | 30 | 50 | 0 |
| $S_4$ | 1 | 130 | 0 | 0 | 20 | 0 |
| $S_4$ | 1 | 125 | 1 | 20 | 30 | 0 |
| $S_4$ | 1 | 120 | 1 | 30 | 80 | 1 |

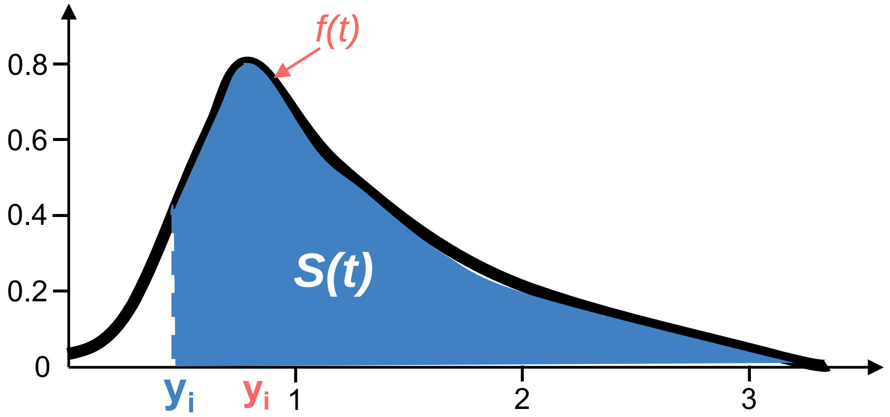
# Taxonomy of Survival Analysis Methods



# Statistical Methods

| Type | Advantages | Disadvantages | Specific methods |
|---|---|---|---|
| **Non-parametric** | More efficient when no suitable theoretical distributions known. | Difficult to interpret; yields inaccurate estimates. | Kaplan-Meier<br>Nelson-Aalen<br>Life-Table |
| **Semi-parametric** | The knowledge of the underlying distribution of survival times is not required. | The distribution of the outcome is unknown; not easy to interpret. | Cox model<br>Regularized Cox<br>CoxBoost<br>Time-Dependent Cox |
| **Parametric** | Easy to interpret, more efficient and accurate when the survival times follow a particular distribution. | When the distribution assumption is violated, it may be inconsistent and can give sub-optimal results. | Tobit<br>Buckley-James<br>Penalized regression<br>Accelerated Failure Time |

# Parametric Censored Regression



- Event density function $f(t)$: rate of events per unit time
  - $\prod_{\delta_i=1} f(y_i,\ \theta)$: The joint probability of uncensored instances.
- Survival function $S(t) = \Pr(T \geq t)$: the probability that the event did not happen up to time $t$
  - $\prod_{\delta_i=0} S(y_i,\ \theta)$: The joint probability of censored instances.

❖ Likelihood function

$$L(\theta) = \prod_{\delta_i=1} f(y_i, \theta) \prod_{\delta_i=0} S(y_i, \theta)$$

# Parametric Censored Regression

- Generalized Linear Model

$$z_i = X_i\beta + \sigma\varepsilon_i \qquad\qquad \varepsilon_i \sim f$$

Where $z_i = \begin{cases} T_i & (Linear\ Model) \\ \log(T_i) & (Accelerated\ Failure\ Time\ Model) \end{cases}$

$$L = \prod_{\delta_i=1} f(\varepsilon_i/\sigma) \prod_{\delta_i=0} [1 - F(\varepsilon_i)]$$

- Negative log-likelihood

$$\min_{\beta,\sigma} -\frac{2}{n}\left\{\sum_{\delta_i=1}\left[\log\bigl(f(\varepsilon_i)\bigr) - \log(\sigma)\right] + \sum_{\delta_i=0} \log\bigl(1 - F(\varepsilon_i)\bigr)\right\}$$

**Uncensored Instances** **censored Instances**

# Optimization

- Use second order second-order Taylor expansion to formulate the log-likelihood as a reweighted least squares

$$
\begin{aligned}
l(\beta) &\approx l(\tilde{\beta}) + (\beta - \tilde{\beta})^T l^{'}(\tilde{\beta}) + \frac{(\beta - \tilde{\beta})^T l^{''}(\tilde{\beta})(\beta - \tilde{\beta})}{2} \\
&= l(\tilde{\beta}) + (X\beta - \tilde{\eta})^T l^{'}(\tilde{\eta}) + \frac{(X\beta - \tilde{\eta})^T l^{''}(\tilde{\eta})(X\beta - \tilde{\eta})}{2} \\
&= \frac{1}{2}(z(\tilde{\eta}) - X\beta)^T l^{'}(\tilde{\eta})(z(\tilde{\eta}) - X\beta) + C(\tilde{\eta}, \tilde{\beta})
\end{aligned}
$$

where $\tilde{\eta} = X\tilde{\beta}, Z(\tilde{\eta}) = \tilde{\eta} - \frac{l''(\tilde{\eta})}{l'(\tilde{\eta})}$. The first-order derivative $l'(\tilde{\eta})$, second-order derivative $l''(\tilde{\eta})$, and other components in optimization share the same formulation with respect to $f(\cdot), f'(\cdot), f''(\cdot), \text{and } \mathrm{F}(\cdot)$.

- In addition, we can add some regularization term to encode some prior assumption.

$$
\min_{\beta} \frac{1}{2}(z(\tilde{\eta}) - X\beta)^T l^{'}(\tilde{\eta})(z(\tilde{\eta}) - X\beta) + R(\beta)
$$

Y. Li, K. S. Xu, C. K. Reddy, “Regularized Parametric Regression for High-dimensional Survival Analysis“, 2016. *SDM*

# Pros and Cons

- Advantages:
  - Easy to interpret.
  - Rather than Cox model, it can directly predict the survival(event) time.
  - More efficient and accurate when the time to event of interest is follow a particular distribution.

- Disadvantages:
  - The model performance strongly relies on the choosing of distribution, and in practice it is very difficult to choose a suitable distribution for a given problem.

Li, Yan, Vineeth Rakesh, and Chandan K. Reddy. "Project success prediction in crowdfunding environments." *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*. ACM, 2016.

# Commonly Used Distributions

| Distributions | PDF $f(t)$ | Survival $S(t)$ | Hazard $h(t)$ |
|---|---|---|---|
| **Exponential** | $\lambda\exp(-\lambda t)$ | $\exp(-\lambda t)$ | $\lambda$ |
| **Weibull** | $\lambda k t^{k-1}\exp(-\lambda t^{k})$ | $\exp(-\lambda t^{k})$ | $\lambda k t^{k-1}$ |
| **Logistic** | $\frac{e^{-(t-\mu)/\sigma}}{\sigma(1+e^{-(t-\mu)/\sigma})^{2}}$ | $\frac{e^{-(t-\mu)/\sigma}}{1+e^{-(t-\mu)/\sigma}}$ | $\frac{1}{\sigma(1+e^{-(t-\mu)/\sigma})}$ |
| **Log-logistic** | $\frac{\lambda k t^{k-1}}{(1+\lambda t^{k})^{2}}$ | $\frac{1}{1+\lambda t^{k}}$ | $\frac{\lambda k t^{k-1}}{1+\lambda t^{k}}$ |
| **Normal** | $\frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{(t-\mu)^{2}}{2\sigma^{2}})$ | $1-\Phi(\frac{t-\mu}{\sigma})$ | $\frac{1}{\sqrt{2\pi}\sigma(1-\Phi(\frac{t-\mu}{\sigma})\ )}\exp(-\frac{(t-\mu)}{2\sigma^{2}}$ |
| **Log-normal** | $\frac{1}{\sqrt{2\pi}\sigma t}\exp(-\frac{(\log(t)-\mu)^{2}}{2\sigma^{2}})$ | $1-\Phi(\frac{\log(t)-\mu}{\sigma})$ | $\frac{\frac{1}{\sqrt{2\pi}\sigma t}\exp(-\frac{(\log(t)-\mu)^{2}}{2\sigma^{2}})}{1-\Phi(\frac{\log(t)-\mu}{\sigma})}$ |

# Tobit Model

- Tobit model is one of the earliest attempts to extend linear regression with the Gaussian distribution for data analysis with censored observations.
- In Tobit model, a latent variable $y^*$ is introduced and it is assumed to linearly depend on $X$ as:

$$\mathrm{y}^* = X\beta \ + \epsilon, \epsilon \sim \ N(0, \sigma^2)$$

where $\epsilon$ is a normally distributed error term.

- For the $i^{th}$ instance, the observable variable $y_i$ will be $y_i^*$ if $y_i^* > 0$, otherwise it will be $0$. This means that if the latent variable is above zero, the observed variable equals to the latent variable and zero otherwise.
- The parameters in the model can be estimated with maximum likelihood estimation (MLE) method.

J. Tobin, Estimation of relationships for limited dependent variables. Econometrica: Journal of the Econometric Society, 1958.

# Buckley-James Regression Method

- The Buckley-James (BJ) regression is a AFT model.

$$\log(T_i) = X_i\beta + \varepsilon_i \qquad y_i = \begin{cases} T_i & \delta_i = 1 \\ C_i & \delta_i = 0 \end{cases}$$

The estimated target value

$$\log(y_i^*) = \begin{cases} \log(y_i) & \delta_i = 1 \\ E(\log(T_i)\,|\,\log(T_i) > \log(y_i),\ X_i) & \delta_i = 0 \end{cases}$$

- The key point is to calculate $E(\log(T_i)\,|\,\log(T_i) > \log(y_i),\ X_i)$:

$$E(\log(T_i)\,|\,\log(T_i) > \log(y_i), X_i) = X_i\beta + E(\varepsilon_i|\varepsilon_i > \log(y_i) - X_i\beta)$$

$$= X_i\beta + \int_{\log(y_i)-X_i\beta}^{\infty} t \cdot \frac{f(t)}{1 - F(\log(y_i) - X_i\beta)}$$

Rather than a selected closed formed theoretical distribution, the Kaplan-Meier (KM) estimation method are used to approximate the F(·).

J. Buckley and I. James, Linear regression with censored data. Biometrika, 1979.

# Buckley-James Regression Method

- The Least squares is used as the empirical loss function

$$\min_{\beta} \frac{1}{2} \sum_{i=1}^{n} (\log(y_i^*) - X_i\beta)^2$$

Where $\log(y_i^*)$=$\delta_i \log(y_i) +$

$$(1-\delta_i)\left\{X_i\beta^{(m-1)} + \int_{\log(y_i)-X_i\beta^{(m-1)}}^{\infty} t \cdot \frac{f(t)}{1 - F(\log(y_i) - X_i\beta^{(m-1)})}\right\}$$

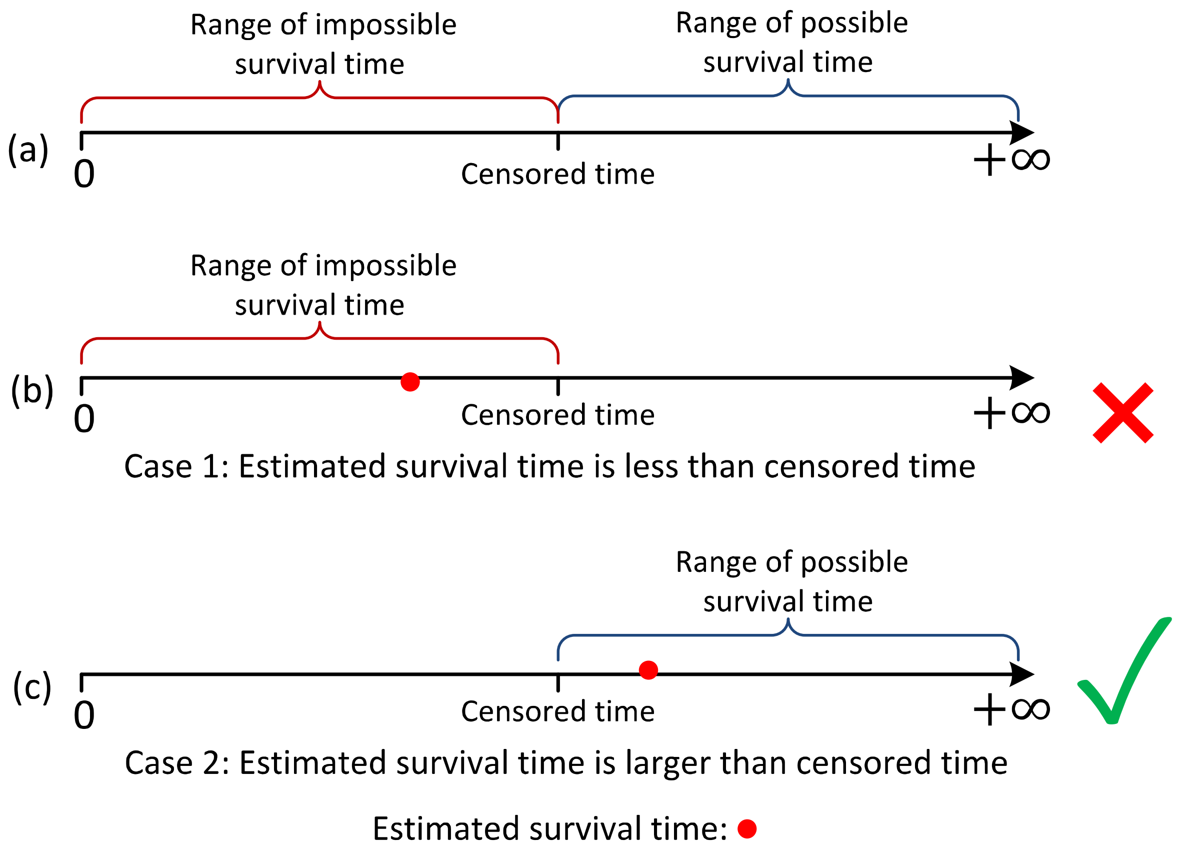
- The Elastic-Net regularizer also has been used to penalize the BJ-regression (EN-BJ) to handle the high-dimensional survival data.

$$\min_{\beta} \frac{1}{2} \sum_{i=1}^{n} (\log(y_i^*) - X_i\beta)^2 + \lambda \left(\alpha\|\beta\|_1 + \frac{1-\alpha}{2}\, \|\beta\|_2^2\right)$$

  - To estimate of $\beta$ of BJ and EN-BJ models, we just need to calculate $\log(y_i^*)$ based on the $\beta$ of pervious iteration and then minimize the lest square or penalized lest square via standard algorithms.

Wang, Sijian, et al. “Doubly Penalized Buckley–James Method for Survival Data with High-Dimensional Covariates.” *Biometrics*, 200

# Regularized Weighted Linear Regression


Range of impossible survival time
Range of possible survival time
(a)
0
Censored time
+∞
Range of impossible survival time
(b)
0
Censored time
+∞
✗
Case 1: Estimated survival time is less than censored time
Range of possible survival time
(c)
0
Censored time
+∞
✓
Case 2: Estimated survival time is larger than censored time
Estimated survival time: ●

- **Induce more penalize to case 1 and less penalize to case 2**

Y. Li, B. Vinzamuri, and C. K. Reddy, "Regularized Weighted Linear Regression for High-dimensional Censored Data", *SDM* 2016.

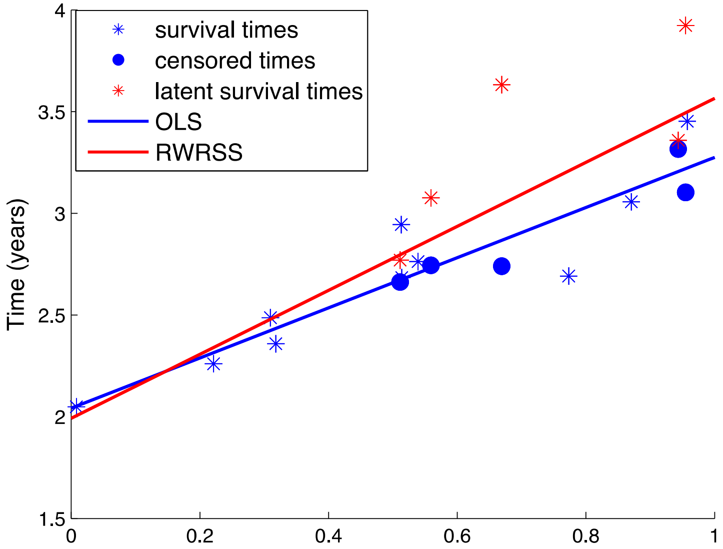
# Weighted Residual Sum-of-Squares

- More weight to the censored instances whose estimated survival time is lesser than censored time
- Less weight to the censored instances whose estimated survival time is greater than censored time.
- Weighted residual sum-of-squares

$$WRSS = \frac{1}{2}\sum_{i=1}^{N}(y_i - X_i\beta)^2 w_i$$

where weight $w_i$ is defined as follows:

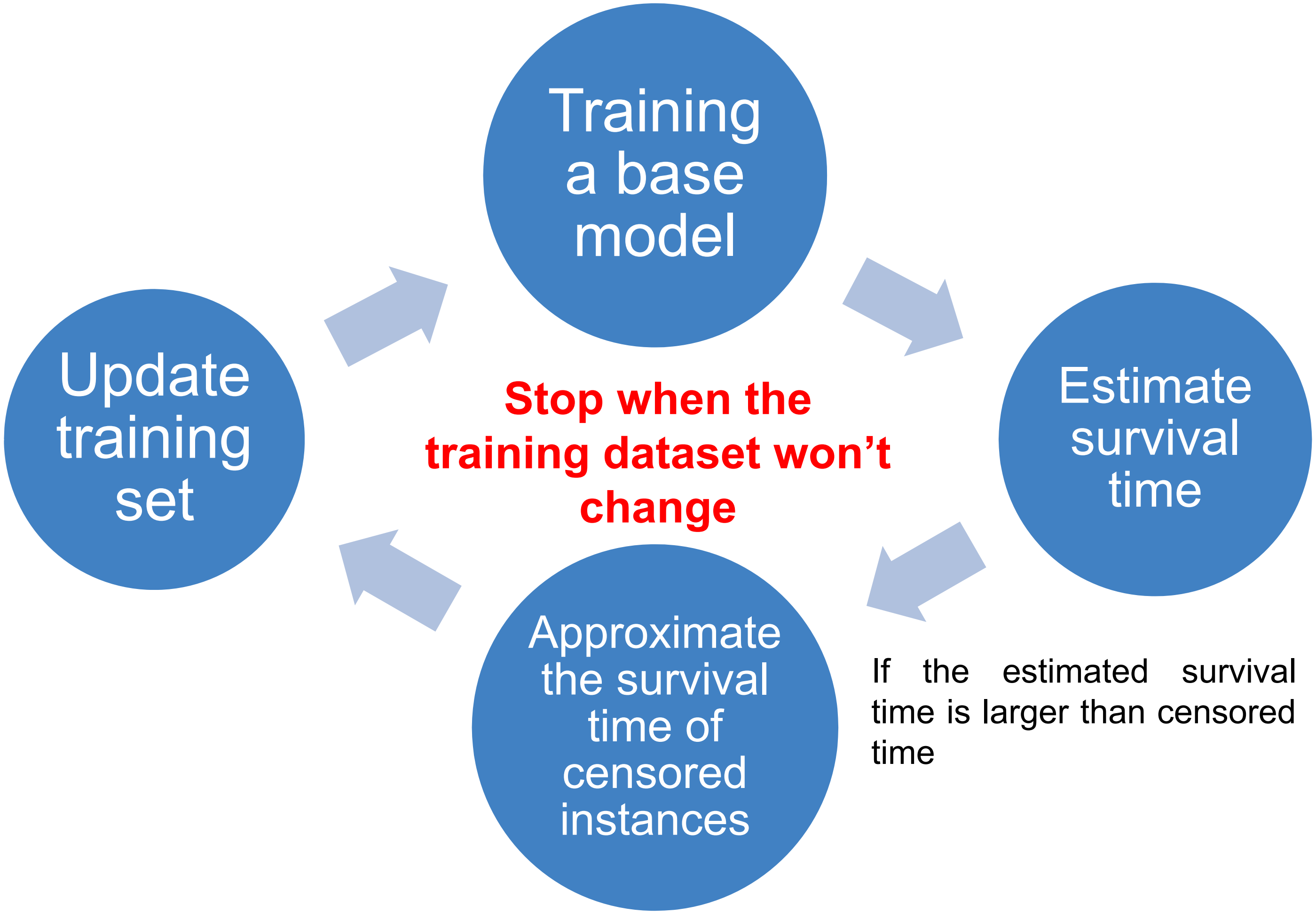
$$w_i = \begin{cases} 1 & if \;\; \delta_i = 1 \\ \tau & if \;\; \delta_i = 0 \;\; and \;\; y_i \geq X_i\beta \\ 0 & if \;\; \delta_i = 0 \;\; and \;\; y_i < X_i\beta \end{cases}$$



A demonstration of linear regression model for dataset with right censored observations.

# Self-Training Framework

***Self-training: training the model by using its own prediction***



**Stop when the training dataset won't change**

If the estimated survival time is larger than censored time

# Bayesian Survival Analysis

- Penalized regression encode assumption via regularization term, while Bayesian approach encode assumption via prior distribution.
- Bayesian Paradigm
  - Based on observed data $D$, one can build a likelihood function $L(\theta|D)$. (likelihood estimator)
  - Suppose $\theta$ is random and has a prior distribution denote by $\pi(\theta)$.
  - Inference concerning $\theta$ is based on the posterior distribution

$$\pi(\theta|D) = \frac{L(\theta|D)\pi(\theta)}{\int_\Theta L(\theta|D)\pi(\theta)d\theta} \qquad m(D) = \int_\Theta L(\theta|D)\pi(\theta)d\theta$$

  - $m(D)$ usually does not have an analytic closed form, requires methods like MCMC to sample from $\pi(\theta|D)$ and methods to estimate $m(D)$.
  - Posterior predictive distribution of a future observation vector $x$ given D

$$\pi(x|D) = \int_\Theta f(x|\theta)\pi(\theta|D)d\theta$$

where $f(x|\theta)$ denotes the sampling density function of $x$

Ibrahim, Joseph G., Ming-Hui Chen, and Debajyoti Sinha. *Bayesian survival analysis*. John Wiley & Sons, 2005.

# Bayesian Survival Analysis

- Under the Bayesian framework the lasso estimate can be viewed as a Bayesian posterior mode estimate under independent Laplace priors for the regression parameters.

Lee, Kyu Ha, Sounak Chakraborty, and Jianguo Sun. "Bayesian variable selection in semiparametric proportional hazards model for high dimensional survival data." *The International Journal of Biostatistics* 7.1 (2011): 1-32.

- Similarly based on the mixture representation of Laplace distribution, the Fused lasso prior and group lasso prior can be also encode based on a similar scheme.

Lee, Kyu Ha, Sounak Chakraborty, and Jianguo Sun. "Survival prediction and variable selection with simultaneous shrinkage and grouping priors." *Statistical Analysis and Data Mining: The ASA Data Science Journal* 8.2 (2015): 114-127.

- A similar approach can also be applied in the parametric AFT model.

Komarek, Arnost. *Accelerated failure time models for multivariate interval-censored data with flexible distributional assumptions*. Diss. PhD thesis, PhD thesis, Katholieke Universiteit Leuven, Faculteit Wetenschappen, 2006.

# Deep Survival Analysis

- Deep Survival Analysis is a hierarchical generative approach to survival analysis in the context of the EHR
- Deep survival analysis models covariates and survival time in a Bayesian framework.
- It can easily handle both missing covariates and model survival time.
- Deep exponential families (DEF) are a class of multi-layer probability models built from exponential families. Therefore, they are capable to model the complex relationship and latent structure to build a joint model for both the covariates and the survival times.

$$z_n \sim DEF(\mathbf{W})$$

$z_n$ is the output of DEF network, which can be used to generate the observed covariates and the time to failure.

R. Ranganath, A. Perotte, N. Elhadad, and D. Blei. "Deep survival analysis." Machine Learning for Healthcare, 2016.

# Deep Survival Analysis

$$z_n \sim DEF(\mathbf{W})$$
$$x_n \sim p(\cdot|\beta, z_n)$$
$$t_n \sim Weibull(\log(1+\exp(z_n^\top a+b), k))$$
$$b \sim Normal(0, \sigma_b)$$
$$a \sim Normal(0, \sigma_W)$$

- $x_n$ is the feature vector, which is supposed can be generated from a prior distribution.
- The Weibull distribution is used to model the survival time.
- a and b are drawn from normal distribution, they are parameter related to survival time.
- Given a feature vector x, the model makes predictions via the posterior predictive distribution:

$$p(t|x) = \int_z p(t|z)p(z|x)dz$$

# Tutorial Outline

- Basic Concepts
- Statistical Methods
- **Machine Learning Methods**
- Related Topics

# Machine Learning Methods

- **Basic ML Models**
  - Survival Trees
    - Bagging Survival Trees
    - Random Survival Forest
  - Support Vector Regression
  - Deep Learning
  - Rank based Methods
- **Advanced ML Models**
  - Active Learning
  - Multi-task Learning
  - Transfer Learning

# Survival Tree

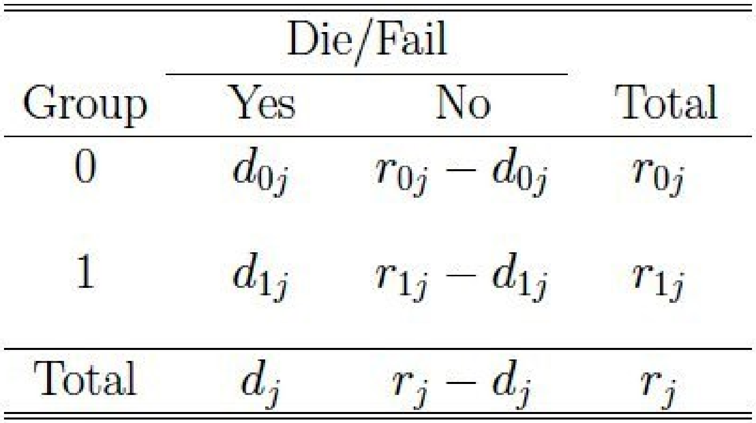
- Survival trees is similar to decision tree which is built by recursive splitting of tree nodes. A node of a survival tree is considered “pure” if all the patients in the node survive for an identical span of time.

- The logrank test is most commonly used dissimilarity measure that estimates the survival difference between two groups. For each node, examine every possible split on each feature, and then select the best split, which maximizes the survival difference between two children nodes.

LeBlanc, M. and Crowley, J. (1993). Survival Trees by Goodness of Split. Journal of the American Statistical Association 88, 457–467.

# Logrank Test

The logrank test is obtained by constructing a (2 X 2) table at each distinct death time, and comparing the death rates between the two groups, conditional on the number at risk in the groups. Let $t_1, \dots, t_K$ represent the $K$ ordered, distinct death times. At the $j$-th death time, we have the following:



|  | Die/Fail |  |  |
|---|---|---|---|
| Group | Yes | No | Total |
| 0 | $d_{0j}$ | $r_{0j} - d_{0j}$ | $r_{0j}$ |
| 1 | $d_{1j}$ | $r_{1j} - d_{1j}$ | $r_{1j}$ |
| Total | $d_j$ | $r_j - d_j$ | $r_j$ |

$$\boldsymbol{X^2_{logrank}} = \frac{\left[\sum_{j=1}^{K}\left(d_{0j} - r_{0j} \times d_j/r_j\right)\right]^2}{\sum_{j=1}^{K} \frac{r_{1j} r_{0j} d_j (r_j - d_j)}{r_j^2 (r_j - 1)}}$$

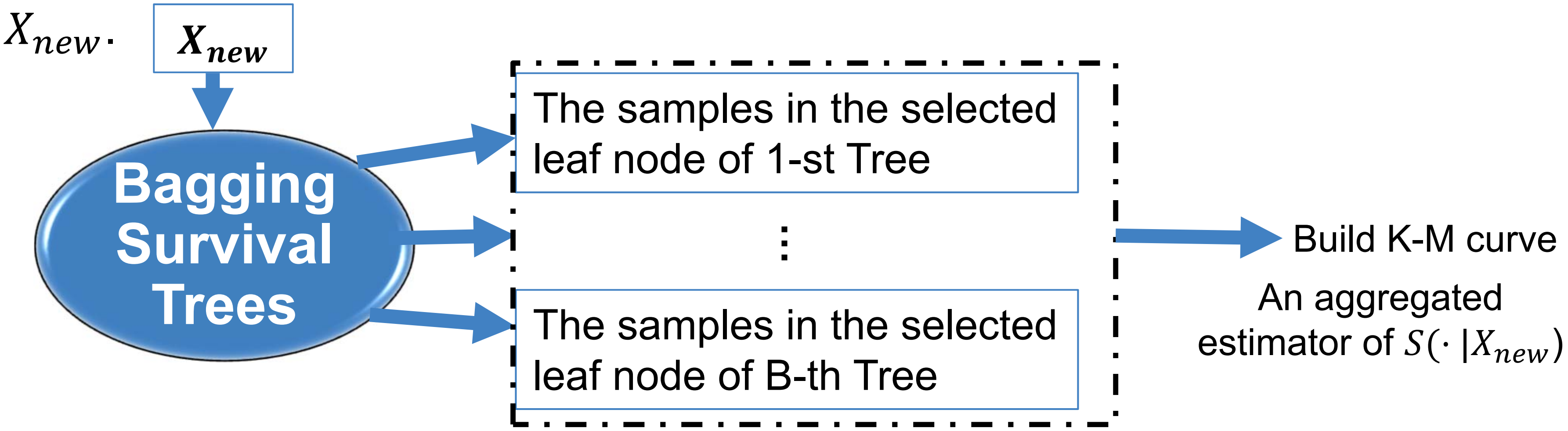
- the numerator is the squared sum of deviations between the observed and expected values. The denominator is the variance of the $d_{0j}$ (Patnaik ,1948).
- The test statistic, $X^2_{logrank}$, gets bigger as the differences between the observed and expected values get larger, or as the variance gets smaller.
- It follows a $\mathcal{X}^2$ distribution asymptotically under the null hypothesis.

Segal, Mark Robert. "Regression trees for censored data." Biometrics (1988): 35-47.

# Bagging Survival Trees

Bagging + Survival Tree = Bagging Survival Trees

- Draw B bootstrap samples from the original data.
- Grow a survival tree for each bootstrap sample based on all features. Recursively spitting the node using the feature that maximizes survival difference between daughter nodes.
- Compute the bootstrap aggregated survival function for a new observation $X_{new}$.



$\boldsymbol{X_{new}}$

Bagging Survival Trees

The samples in the selected leaf node of 1-st Tree

⋮

The samples in the selected leaf node of B-th Tree

Build K-M curve

An aggregated estimator of $S(\cdot\,|X_{new})$

Hothorn, Torsten, et al. "Bagging survival trees." *Statistics in medicine* 23.1 (2004): 77-91.

# Random Survival Forests

Random Forests + Survival Tree = RSF

1. Draw B bootstrap samples from the original data (63% in the bag data, 37% Out of bag data(OOB)).
2. Grow a survival tree for each bootstrap sample based on randomly select $p$ candidate features, and splits the node using feature from the selected candidate features that maximizes survival difference between daughter nodes.
3. Grow the tree to full size, each terminal node should have no less than $d_0 > 0$ unique deaths.
4. Calculate a Cumulative Hazard Function (CHF) for each tree. Average to obtain the bootstrap ensemble CHF.
5. Using OOB data, calculate prediction error for the OOB ensemble CHF.

H. Ishwaran, U. B. Kogalur, E. H. Blackstone and M. S. Lauer, “Random Survival Forests”. Annals of Applied Statistics, 2008

# Random Survival Forests

- The cumulative hazard function (CHF) in random survival forests is estimated via Nelson-Aalen estimator:

$$\widehat{H_h}(t) = \sum_{t_{l,h}<t} \frac{d_{l,h}}{r_{l,h}}$$

where $t_{l,h}$ is the $l$-th distinct event time of the samples in leaf $h$, $d_{l,h}$ is the number events at $t_{l,h}$, and $r_{l,h}$ is the number of individuals at risk at $t_{l,h}$.

- OOB ensemble CHF ($H_e^{**}(t|x_i)$) and bootstrap ensemble CHF ($H_e^{*}(t|x_i)$)

$$H_e^{**}(t|x_i) = \frac{\sum_{b=1}^{B} I_{i,b} H_b^{*}(t|x_i)}{\sum_{b=1}^{B} I_{i,b}}, \qquad H_e^{*}(t|x_i) = \frac{1}{B}\sum_{b=1}^{B} H_b^{*}(t|x_i)$$

where $H_b^{*}(t|x_i)$ is the CHF of the node in b-th bootstrap which $x_i$ belongs to. $I_{i,b} = 1$ if i is an OOB case for b; otherwise, set $I_{i,b} = 0$. Therefore OOB ensemble CHF is the average over bootstrap samples which i is OOB, and bootstrap ensemble CHF is the average of all B bootstrap.

O. O. Aalen, "Nonparametric inference for a family of counting processes", *Annals of Statistics* 1978.

# Support Vector Regression (SVR)

- Once a model has been learned, it can be applied to a new instance $X$ through

$$f(x) = W \bullet \Phi(x) + b$$

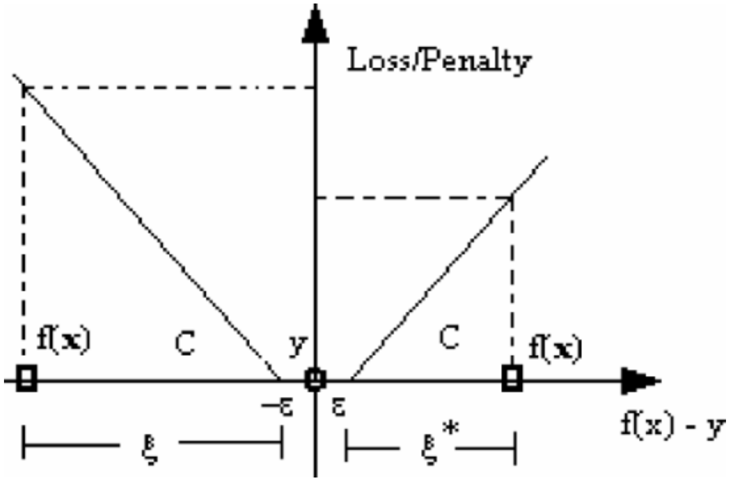
$\Phi(x)$ is a kernel, and the SVR algorithm can abstractly be considered as a linear algorithm

$$\min_{W,b} \frac{1}{2}||W||^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*)$$

$$y_i - (W \bullet \Phi(x_i) + b) \leq \varepsilon + \xi_i$$

$$(W \bullet \Phi(x_i) + b) - y_i \leq \varepsilon + \xi_i^*$$

$$\xi_i, \xi_i^* \geq 0, \qquad i = 1 \ldots n$$



$\varepsilon$: margin of error

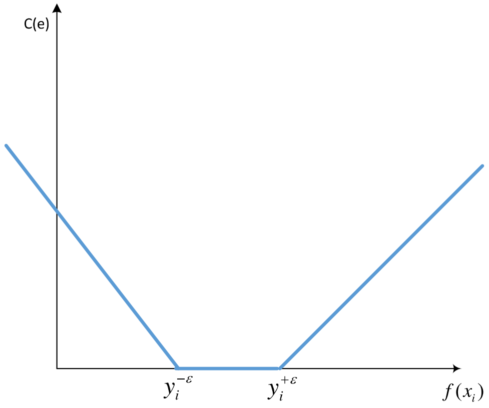
$C$: regularization parameter

$\xi_i, \xi_i^*$: slack variables $|\xi|_\varepsilon = \begin{cases} 0 & \text{if } |\xi| \leq \varepsilon \\ |\xi| - \varepsilon & \text{if otherwise} \end{cases}$

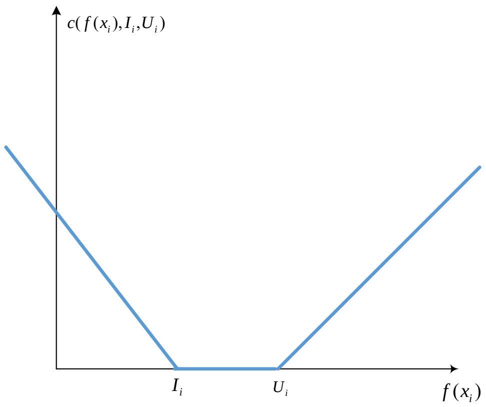
# Support Vector Approach for Censored Data

- Interval Targets: These are samples for which we have both an upper and a lower bound on the target. The tuple $(x_i,l_i,u_i)$ with $y_i - \varepsilon_i = l_i < u_i = y_i + \varepsilon_i$. As long as the output $f(x_i)$ is between $l_i$ and $u_i$, there is no empirical error.
- Right censored sample is written as $(x_i,\ l_i\ +\infty)$ whose survival time is greater than $l_i \in \mathbb{R}$, but the upper bound is unknown.

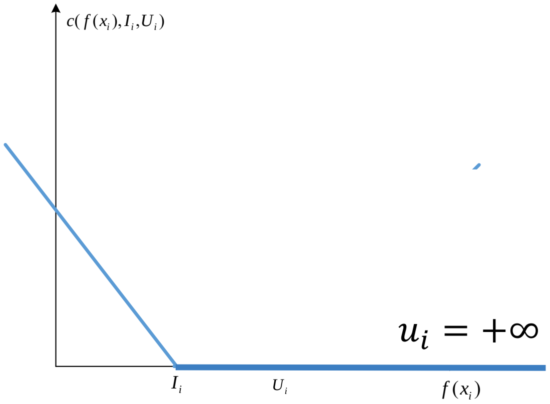
Graphical representation of Loss functions



SVR loss



SVRC loss in general

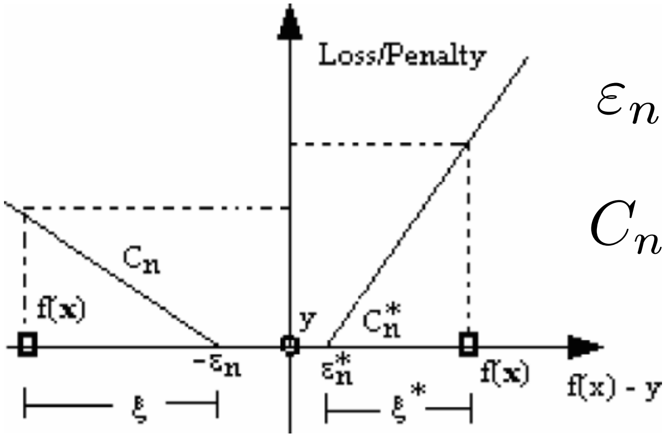


SVRC loss for right censored

P. K. Shivaswamy, W. Chu, and M. Jansche. "A support vector approach to censored targets", *ICDM* 2007.

# Support Vector Regression for Censored Data

- A graphical representation of the SVRc parameters for events.

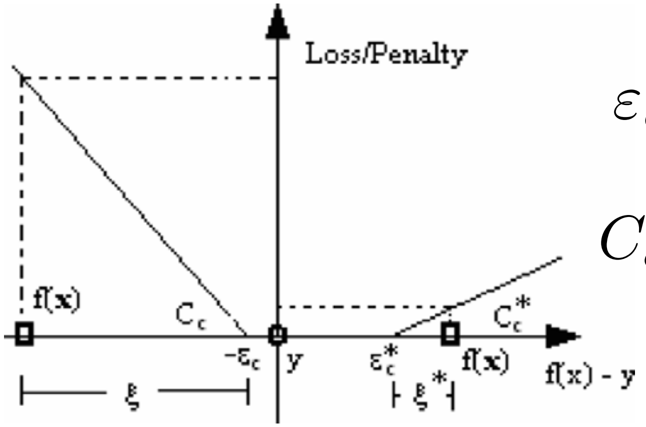


$\varepsilon_n > \varepsilon_n^*$ Lesser acceptable margin when the predicted value is grater than the event time

$C_n < C_n^*$ Greater penalty rate when the predicted value is greater than the censored time

Predicting a high risky patient will survive longer is more gangrenous than predicting a low risky patient will survive shorter

- Graphical representation of the SVRc parameters for censored data.



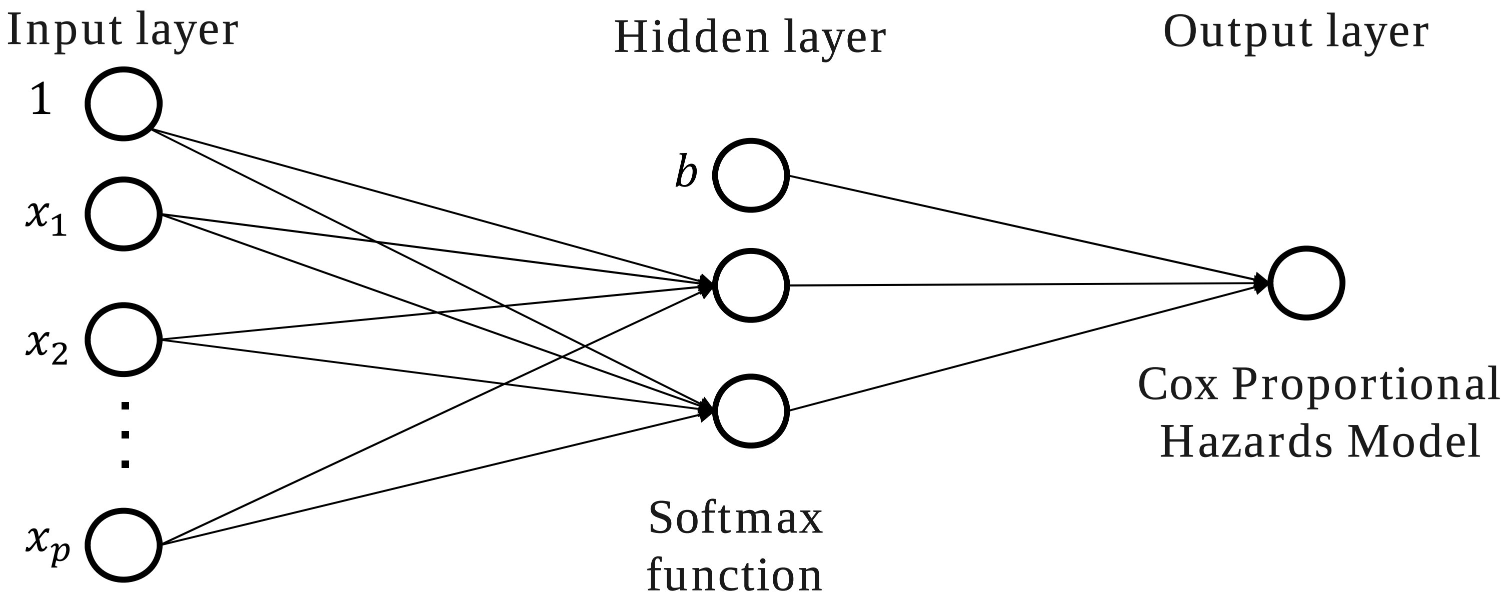
$\varepsilon_c < \varepsilon_c^*$ Greater acceptable margin when the predicted value is greater than the censored time

$C_c > C_c^*$ Less penalty rate when the predicted value is greater than the censored time

The possible survival time of censored instances should be grater than or equal to the corresponding censored time.

F. M. Khan and V. B. Zubek. "Support vector regression for censored data (SVRc): a novel tool for survival analysis." *ICDM 2008*

# Neural Network Model



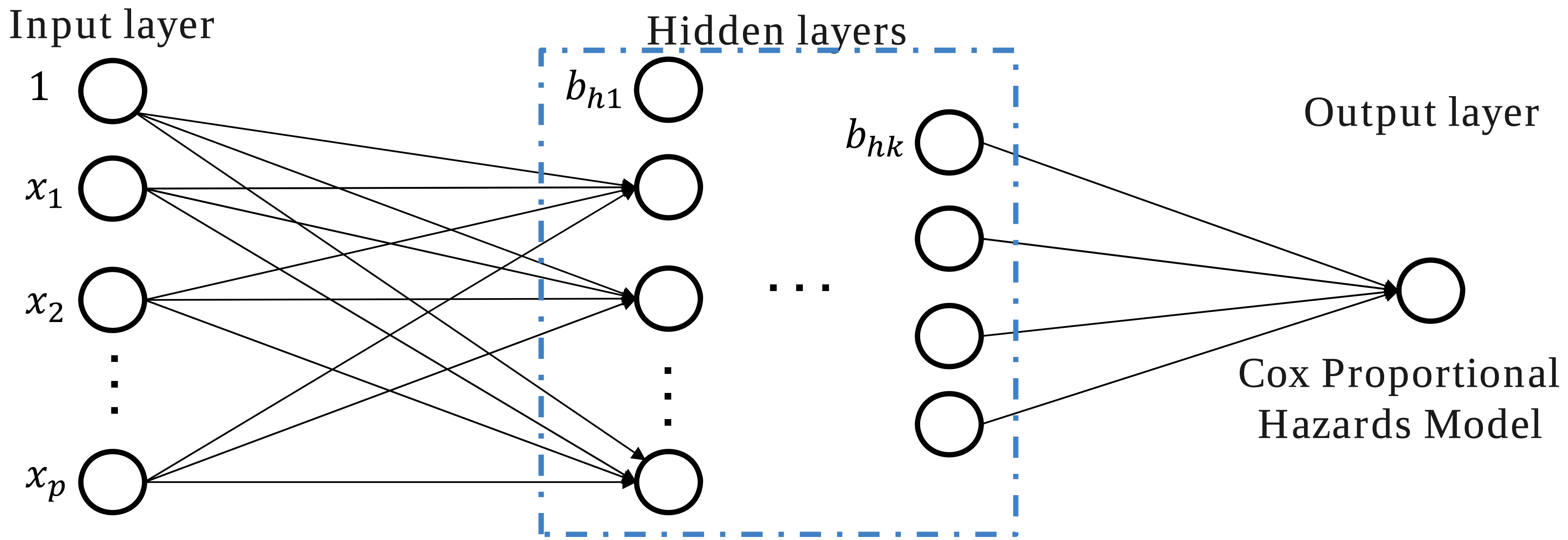
- Hidden layer takes softmax $g(x, w)$ as active function.

$$(\boldsymbol{w}) = -\sum_{\{\boldsymbol{i}:\boldsymbol{D_i}=\boldsymbol{1}\}} \boxed{g(\boldsymbol{x_i}, \boldsymbol{w})} + \boldsymbol{log} \sum_{\{\boldsymbol{j}:\boldsymbol{t_j} \geq \boldsymbol{t_i}\}} \boldsymbol{exp}\left(\boxed{g(\boldsymbol{x_j}, \boldsymbol{w})}\right)$$

No longer to be a linear function

D. Faraggi and R. Simon. "A neural network model for survival data." *Statistics in medicine*, 1995.

# Deep Survival: A Deep Cox Proportional Hazards Network



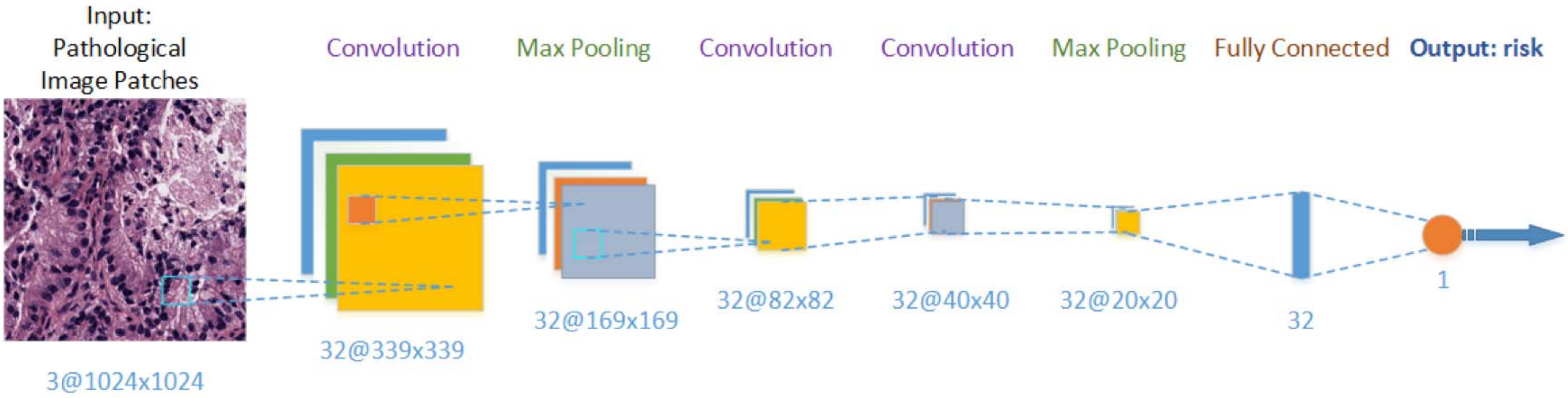
- Takes some modern deep learning techniques such as Rectified Linear Units (ReLU) active function, Batch Normalization, dropout.

$$\boldsymbol{l(w) = -\sum_{\{i:D_i=1\}} \boxed{h^{last}(x_i, w)} + log \sum_{\{j:t_j \geq t_i\}} exp\left(\boxed{h^{last}(x_j, w)}\right)}$$

No longer to be a linear function

Katzman, Jared, et al. "Deep Survival: A Deep Cox Proportional Hazards Network." *arXiv , 2016*.

# Deep Convolutional Neural Network



$$\boldsymbol{l(w) = -\sum_{\{i:D_i=1\}} \boxed{h^{last}(x_i, w)} + log \sum_{\{j:t_j \geq t_i\}} exp\left(\boxed{h^{last}(x_j, w)}\right)}$$

No longer to be a liner function

$x_i$: image patch from $i$-th patient
$w$: the deep model

Pos: Directly built deep model for survival analysis from images input

X. Zhu, J. Yao, and J. Huang. "Deep convolutional neural network for survival analysis with pathological images“, *BIBM* 2016.

# Ranking based Models

- C-index is a pairwise ranking based evaluation metric. Boosting concordance index (BoostCI) is an approach which aims at directly optimize the C-index.

$$C = \frac{\sum_{i,j} (\hat{G}_n^L(y_i))^{-2} I(y_i < y_j) I(\hat{y}_i > \hat{y}_j) \delta_i}{\sum_{i,j} (\hat{G}_n^L(y_i))^{-2} I(y_i < y_j) \delta_i}$$

$\hat{G}_n^L(\cdot)$ is the kaplan-Meier estimator, and as the existence of $I(\cdot)$ the above definition is non-smooth and nonconvex, which is hart to optimize.

- In BoostCI, a sigmoid function is used to provide a smooth approximation for indicator function.

$$I(\hat{y}_i > \hat{y}_j) \approx \frac{1}{1 + exp(\frac{\hat{y}_j - \hat{y}_i}{\sigma})}$$

Therefore, we have the smoothed version

$$C_{smooth} = \sum_{i,k} w_{ik} \frac{1}{1 + exp(\frac{\hat{y}_k - \hat{y}_i}{\sigma})}$$ weights $$w_{ij} := \frac{\delta_i (\hat{G}_n^L(y_i))^{-2} I(y_i < y_j)}{\sum_{i,j} \delta_i (\hat{G}_n^L(y_i))^{-2} I(y_i < y_j)}$$

A. Mayr and M. Schmid, “Boosting the concordance index for survival data–a unified framework to derive and evaluate biomarker combinations”, *PloS one*, 2014.

# BoostCI Algorithm

- The component-wise gradient boosting algorithm is used to optimize the smoothed C-index.

**Learning Step:**

1. **Initialize** the estimate of the marker combination $\hat{y}$ with offset values, and set maximum number ($m_{max}$) of iteration, and set $m = 1$.
2. **Compute** the negative gradient vector of smoothed C-index.
3. **Fit** the negative gradient vector separately to each of the components of $X$ via the base-learners $\hat{b}_l(X_{(:,l)})$.
4. **Select** the component that best fits the negative gradient vector, and the selected index of base-learn is denote as $l^*$
5. **Update** the marker combination $\hat{y}$ for this component

$$\hat{y}^{[m]} \leftarrow \hat{y}^{[m+1]} + \alpha \hat{b}_{l^*}(X_{(:,l^*)}).$$

6. **Stop** if $m = m_{max}$. Else increase $m$ by one and go back to step 2

# Machine Learning Methods

- **Basic ML Models**
  - Survival Trees
    - Bagging Survival Trees
    - Random Survival Forest
  - Support Vector Machine
  - Deep Learning
  - Rank based Methods
- **Advanced ML Models**
  - Active Learning
  - Multi-Task Learning
  - Transfer Learning

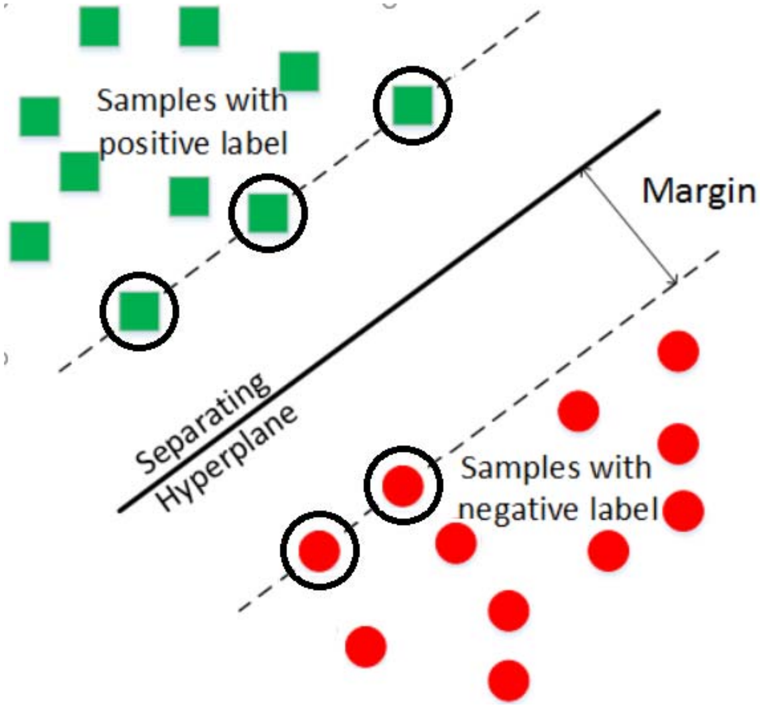
# Active Learning for Survival Data

- **Objective:** Identify the representative samples in the data

**Outcome:** Allow the Model to select instances to be included. It can minimize the training cost and complexity of the model and obtain a good generalization performance for Censored data.

Our sampling method chooses that particular instance which maximizes the following criterion.

$$X^* = \arg\max_{X \in pool} \sum_{k=1}^{K} h(T_k \mid X) \left\| \frac{\partial L_X(\beta)}{\partial \beta} \right\|$$

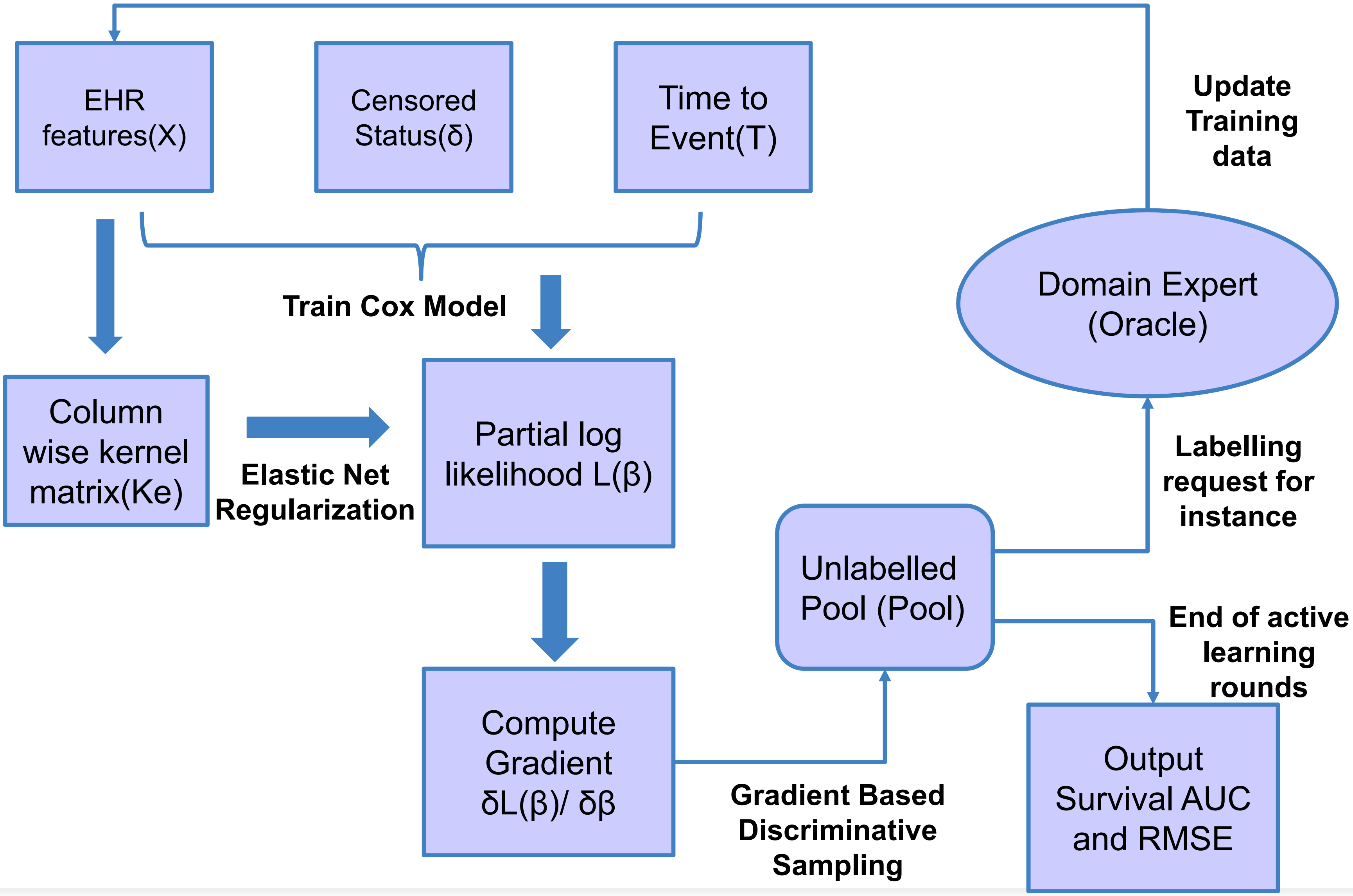


- Active learning based framework for the survival regression using a novel **model discriminative gradient based sampling** procedure.

- Helps clinicians to understand more about the most representative patients.

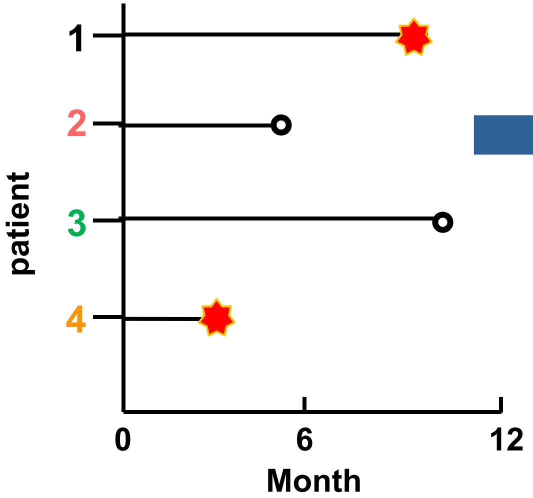
B. Vinzamuri, Y. Li, C. Reddy, "Active Learning Based Survival Regression for Censored Data", *CIKM 2014*.

# Active Learning with Censored Data


EHR features(X)
Censored Status(δ)
Time to Event(T)
Update Training data
Domain Expert (Oracle)
Train Cox Model
Column wise kernel matrix(Ke)
Elastic Net Regularization
Partial log likelihood L(β)
Labelling request for instance
Unlabelled Pool (Pool)
End of active learning rounds
Compute Gradient δL(β)/ δβ
Gradient Based Discriminative Sampling
Output Survival AUC and RMSE

# Multi-task Learning Formulation

**Advantage:** The model is general, no assumption on either survival time or survival function.



| Y | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *1* | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| *2* | 1 | 1 | 1 | 1 | 1 | ? | ? | ? | ? | ? | ? | ? |
| *3* | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | ? | ? |
| *4* | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**1: Alive** **0: Death** **?: Unknown**

- **Similar tasks:** All the binary classifiers aim at predicting the life status of each patient.
- **Temporal smoothness**: For each patient, the life statuses of adjacent time intervals are mostly same.
- **Not reversible:** Once a patient is dead, he is impossible to be alive again.

# Multi-task Learning Formulation

| Y | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| D1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| D2 | 1 | 1 | 1 | 1 | 1 | ? | ? | ? | ? | ? | ? | ? |
| D3 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | ? | ? |
| D4 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| W | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| D1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| D2 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| D3 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| D4 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

How to deal with the "?" in Y

The Proposed objective function:

$$\min_{XB\in P} \frac{1}{2}\|\Pi_W(Y-XB)\|_F^2 + \frac{\lambda_1}{2}\|B\|_F^2 + \lambda_2\|B\|_{2,1}$$

Where

$$(\Pi_W(U))_{ij} = \begin{cases} U_{ij} & if \quad W_{ij} = 1 \\ 0 & if \quad W_{ij} = 0 \end{cases}$$

**Handling Censored**

**Similar tasks:** select some common features across all the task via $l_{2,1}$-norm.

**Temporal smoothness & Irreversible:**

Y and $\hat{Y}$ should follow a non-negative non-increasing list structure

$$P = \{Y \geq 0, Y_{ij} \geq Y_{il} | j \leq l, \forall j = 1, \ldots, k, \forall l = 1, \ldots, k\}$$

Yan Li, Jie Wang, Jieping Ye and Chandan K. Reddy "A Multi-Task Learning Formulation for Survival Analysis". *KDD* 2016

# Multi-task Learning Formulation

$$\min_{M \in P} \frac{1}{2} \|\Pi_W(Y - M)\|_F^2 + \frac{\lambda_1}{2} \|B\|_F^2 + \lambda_2 \|B\|_{2,1}$$

Subject to: $M = XB$

ADMM:

$$M^{t+1} = \min_{M \in P} \left( \frac{1}{2} \|\Pi_W(Y - M)\|_F^2 + \frac{\rho}{2} \|M - XB^t + u^t\|_F^2 \right)$$

**Solving the non-negative non-increasing list structure by max-heap projection**

$$B^{t+1} = \min_{B \in \mathbb{R}^{p \times k}} \left( \frac{\lambda_1}{2} \|B\|_F^2 + \lambda_2 \|B\|_{2,1} + \frac{\rho}{2} \|M^{t+1} - XB + u^t\|_F^2 \right)$$

**Solving the $l_{2,1}$-norm by using FISTA algorithm**

$$u^{t+1} = u^t + M^{t+1} - XB^{t+1}$$

## An adaptive variant model

**Too many time intervals, non-negative non-increasing list will be so strong that will overfit the model. Relaxation of the above model:**

$$\min_{B \in \mathbb{R}^{p \times k}} \frac{1}{2} \|\Pi_W(Y - XB)\|_F^2 + \frac{\lambda_1}{2} \|B\|_F^2 + \lambda_2 \|B\|_{2,1}$$

# Multi-Task Logistic Regression

- Model survival distribution via a sequence of dependent regressions.
- Consider a simpler classification task of predicting whether an individual will survive for more than $t$ months.

$$S_\beta(t) = P_\beta(T \geq t|x) = \frac{1}{1 + \exp(x\beta + b)}$$

- Consider a serious of time points $(t_1, t_2, t_3, \ldots, t_k)$, we can get a series of logistic regression models

$$P_{\beta_{(:,i)}}(T \geq t_i|x) = \frac{1}{1 + \exp(x\beta_{(:,i)} + b_i)}, \quad 1 \geq i \geq m$$

- The model should enforce the dependency of the outputs by predicting the survival status of a patient at each of the time snapshots, let $(y_1, y_2, y_3, \ldots, y_k)$ where $y_i = 0$ (no death event yet ), and $y_i = 1$ (death)

$$P(Y = (y_1, y_2, \cdots, y_k)|x) = \frac{\exp(\sum_{i=1}^{k} y_k(x\beta_{(:,i)} + b_i))}{\sum_{i=0}^{k} \exp(\sum_{l=i+1}^{k}(x\beta_{(:,l)} + b_l))} \quad \text{for} \quad 0 \leq l \leq k$$

C. Yu et al. "Learning patient-specific cancer survival distributions as a sequence of dependent regressors." *NIPS* 2011.

# Multi-Task Logistic Regression

$$P(Y=(y_1,y_2,\cdots,y_k)|x) = \frac{\exp(\sum_{i=1}^{k} y_k(x\beta_{(:,i)}+b_i))}{\sum_{i=0}^{k}\exp(\sum_{l=i+1}^{k}(x\beta_{(:,l)}+b_l))} \quad \text{for} \quad 0 \leq l \leq k$$

- A very similar idea as cox model:

$\exp\left(\sum_{l=i+1}^{k}\left(x\beta_{(:,l)}+b_l\right)\right) = \exp\left(\sum_{l=i+1}^{k} y_l\left(x\beta_{(:,l)}+b_l\right)\right)$ with $y_l = 1\ \forall\ l = i+1,\dots,k.$

is the score of sequence with the event occurring in the interval $[t_i, t_{i+1})$. But different from cox model the coefficient is different in different time interval. So no proportional hazard assumption.

- For censored instances:

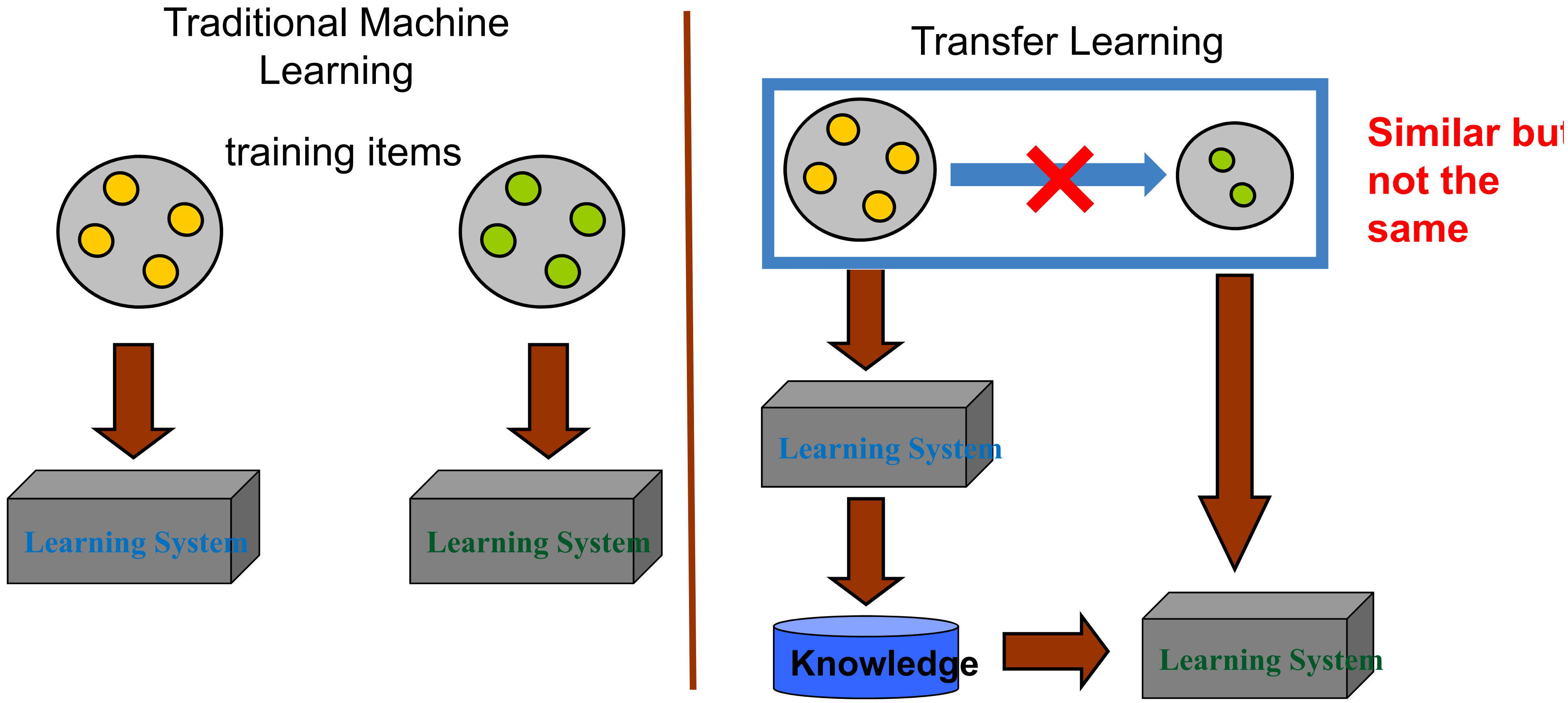
$$P(T \geq t_c)|x) = \frac{\sum_{i=c}^{k}\exp(\sum_{l=i+1}^{k}(x\beta_{(:,l)}+b_l))}{\sum_{i=0}^{k}\exp(\sum_{l=i+1}^{k}(x\beta_{(:,l)}+b_l))}$$

The numerator is the score of the death will happen after $t_c$

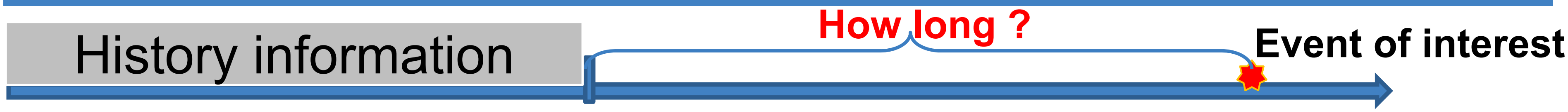
- In the model add $\sum_{l=0}^{k-1}\left\|\beta_{(:,l+1)}-\beta_{(:,l)}\right\|^2$ regularization term to achieve temporary smoothness.

# Knowledge Transfer

- Transfer learning models aim at using auxiliary data to augment learning when there are insufficient number of training samples in target dataset.



# Transfer Learning for Survival Analysis

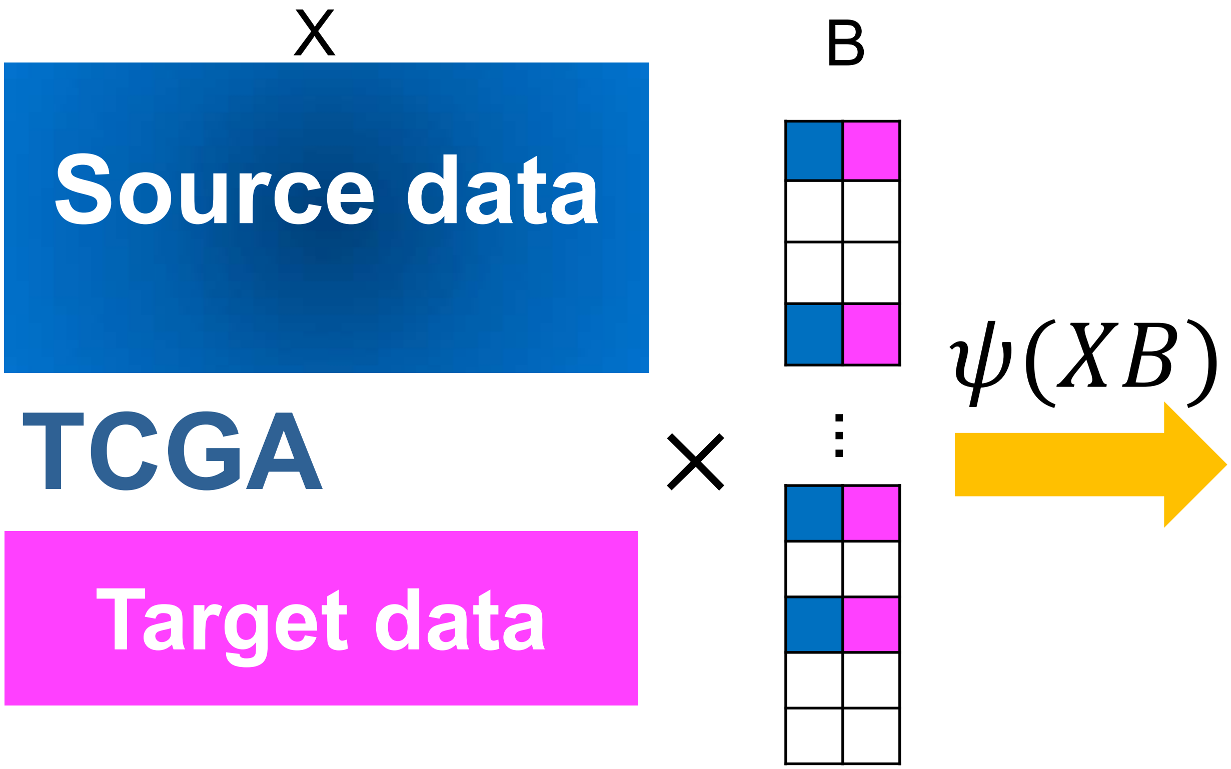


History information

How long ?

Event of interest

**Labeling the time-to-event data is very time consuming!**



X

B

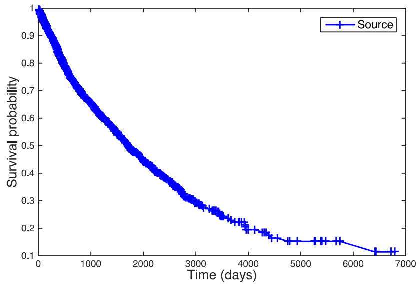
Source data

TCGA

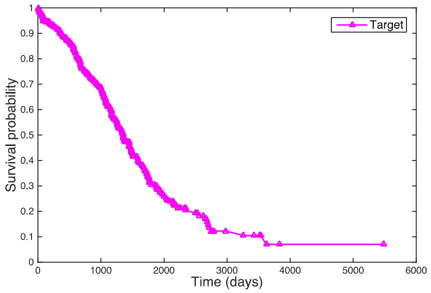
Target data

$\times$

$\psi(XB)$



Source Task



Target Task

- Both source and target tasks are survival analysis problem.
- There exist some features which are important among all correlated disease.

Yan Li, Lu Wang, Jie Wang, Jieping Ye and Chandan K. Reddy "Transfer Learning for Survival Analysis via Efficient L2,1-norm Regularized Cox Regression". *ICDM* 2016.

# Transfer-Cox Model

The Proposed objective function:

$$\min_{\beta_S,\beta_T} \boxed{\frac{1}{N_S} l_S(\beta_S) + \frac{w}{N_T} l_T(\beta_T)} + \frac{\mu}{2}\|B\|_F^2 + \boxed{\lambda\|B\|_{2,1}}$$

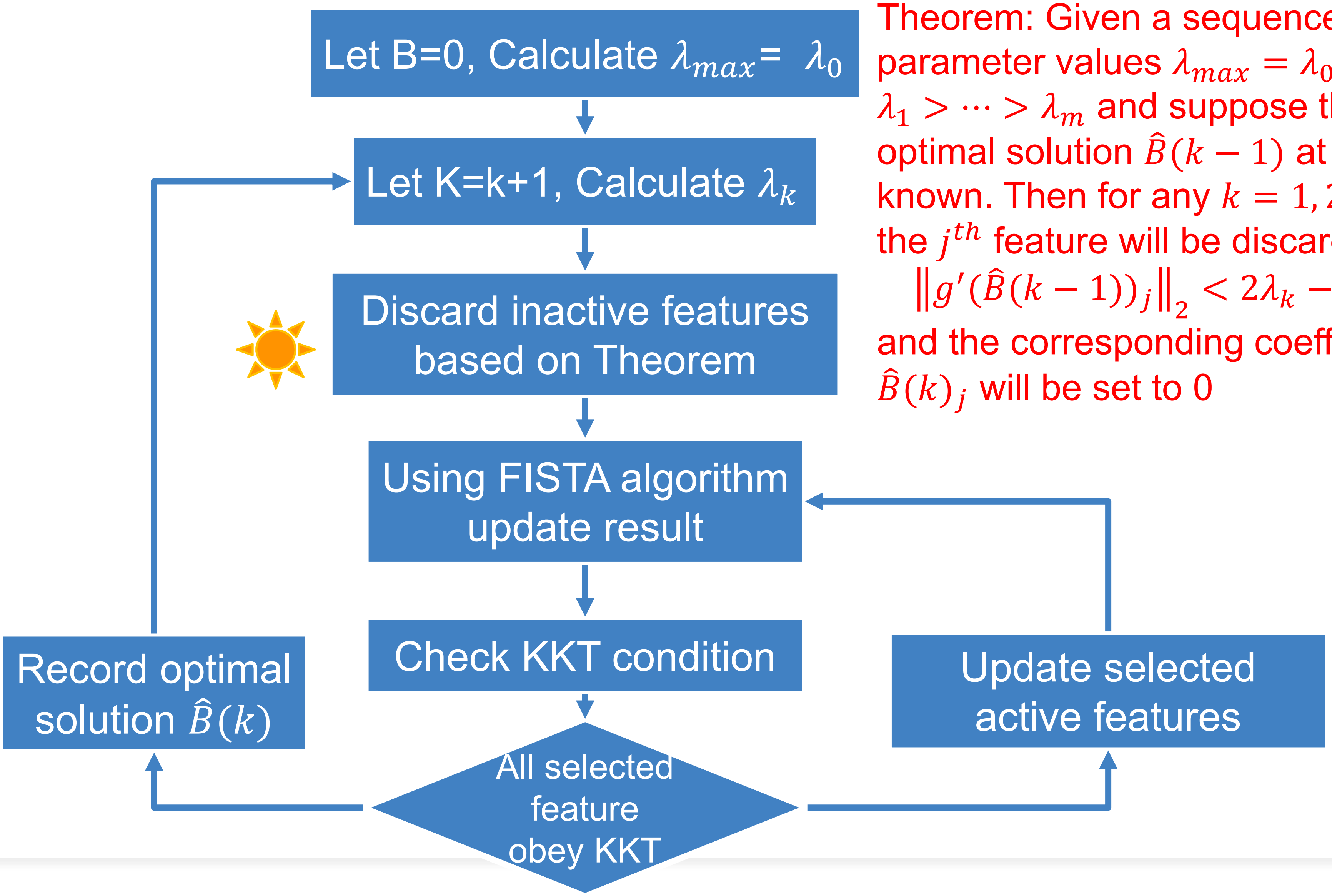
Where $\beta_S$, $l_S(\beta_S)$, $\beta_T$, and $l_T(\beta_T)$ denote the coefficient vector and negative partial log-likelihood,

$$l(\beta) = n^{-1}\sum_{i=1}^{K} -X_i^T\beta + \log\left(\sum_{j\epsilon R_\mathrm{i}} exp\left(X_j^T \beta\right)\right),$$

of source take and target take, respectively. And $B = (\beta_S, \beta_T)$.

- L2,1 norm can encourage group sparsity; therefore, it selects some common features across all the task.
- We propose a FISTA based algorithm to solve the problem with a linear scalability.

# Using Strong Rule in Learning Process



Let B=0, Calculate $\lambda_{max}= \ \lambda_0$

Let K=k+1, Calculate $\lambda_k$

Discard inactive features based on Theorem

Using FISTA algorithm update result

Check KKT condition

All selected feature obey KKT

Record optimal solution $\hat{B}(k)$

Update selected active features

Theorem: Given a sequence of parameter values $\lambda_{max} = \lambda_0 > \lambda_1 > \cdots > \lambda_m$ and suppose the optimal solution $\hat{B}(k-1)$ at $\lambda_{k-1}$ is known. Then for any $k = 1, 2, \ldots, \mathrm{m}$ the $j^{th}$ feature will be discarded if

$$\left\| g'(\hat{B}(k-1))_j \right\|_2 < 2\lambda_k - \lambda_{k-1}$$

and the corresponding coefficient $\hat{B}(k)_j$ will be set to 0

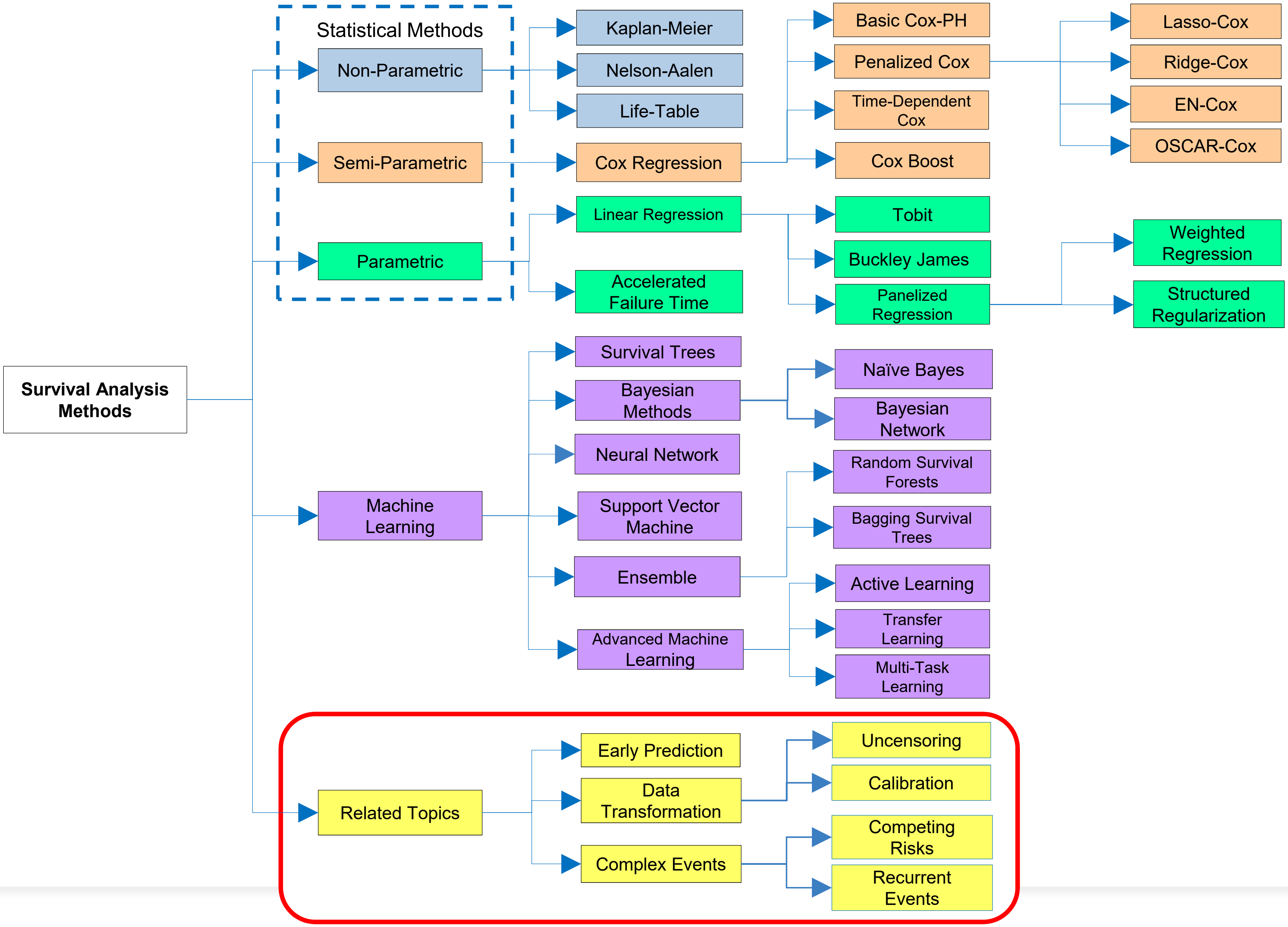
# Summary of Machine Learning Methods

- **Basic ML Models**
  - Survival Trees
    - Bagging Survival Trees
    - Random Survival Forest
  - Support Vector Regression
  - Deep Learning
  - Rank based Methods
- **Advanced ML Models**
  - Active Learning
  - Multi-Task Learning
  - Transfer Learning

# Tutorial Outline

- Basic Concepts
- Statistical Methods
- Machine Learning Methods
- **Related Topics**

# Taxonomy of Survival Analysis Methods

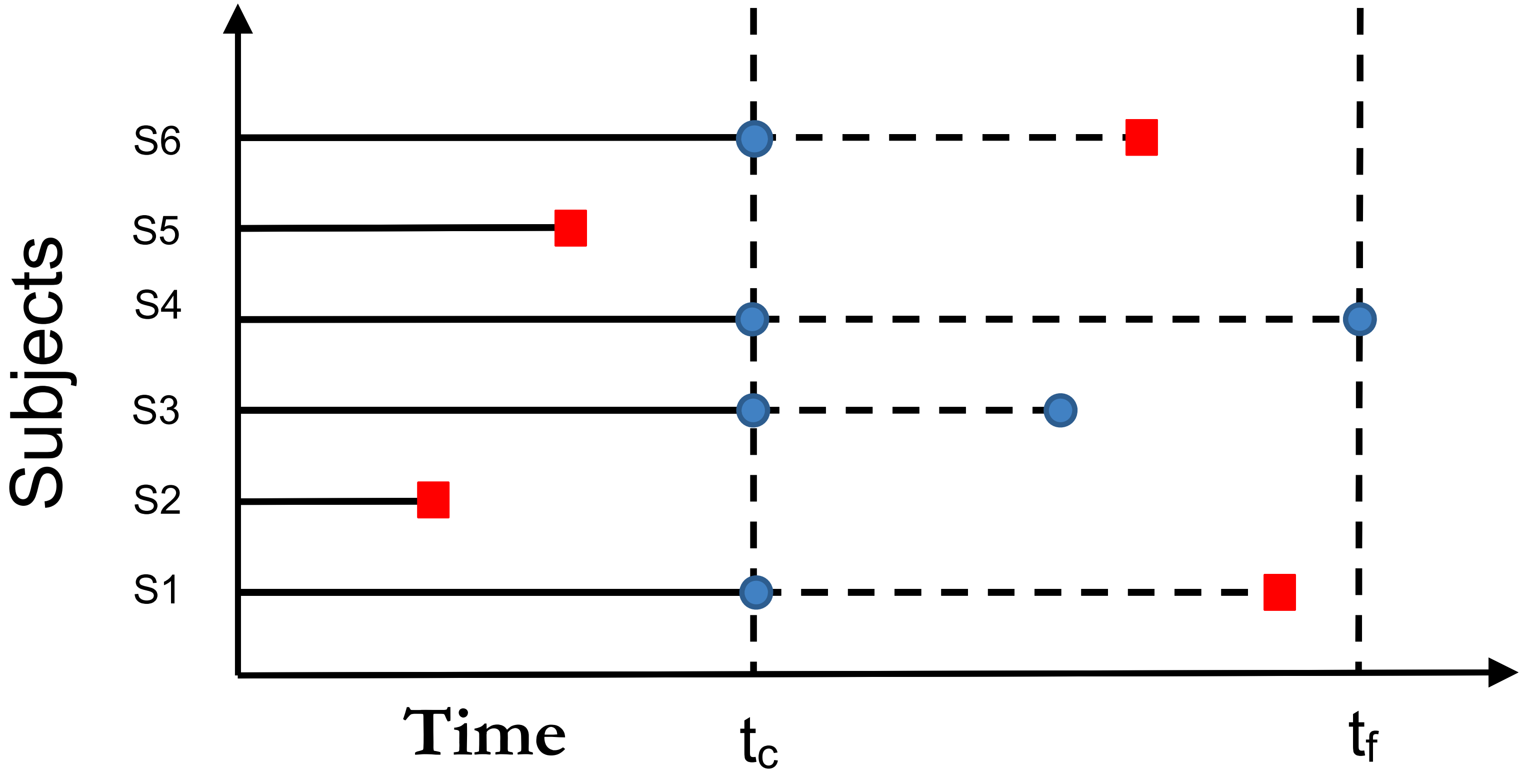


# Related Topics

- Early Prediction
- Data Transformation
  - Uncensoring
  - Calibration
- Complex Events
  - Competing Risks
  - Recurrent Events

# Early Stage Event Prediction

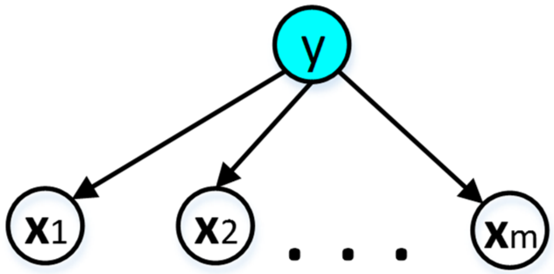
- Collecting data for survival analysis is very “time” consuming.



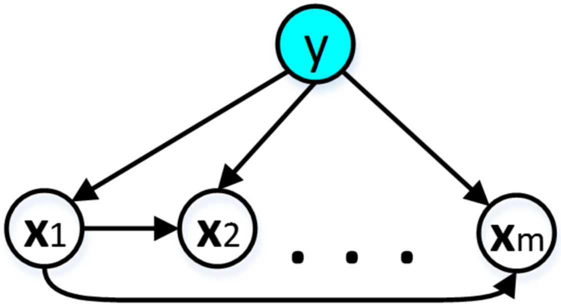
- Any existing survival model can predict only until $t_c$
- Develop a Bayesian approach for early stage prediction.

M. J Fard, P. Wang, S. Chawla, and C. K. Reddy, “A Bayesian perspective on early stage event prediction in longitudinal data”, *TKDE* 2016.

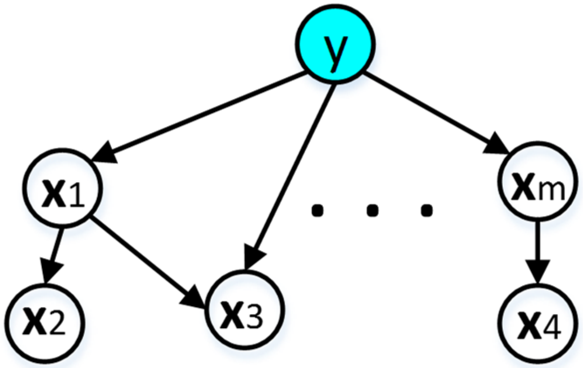
# Bayesian Approach



Naïve Bayes (NB)



Tree-Augmented Naïve Bayes (TAN)



Bayesian Networks (BN)

$$\prod_{j=1}^{m} P\left(x_j \mid y\left(t_c\right)=1\right) \qquad \prod_{j=1}^{m} P\left(x_j \mid y\left(t_c\right)=1, x_p\left(j\right)\right) \qquad \prod_{j=1}^{m} P\left(x_j \mid y\left(t_c\right)=1, Pa\left(x_j\right)\right)$$

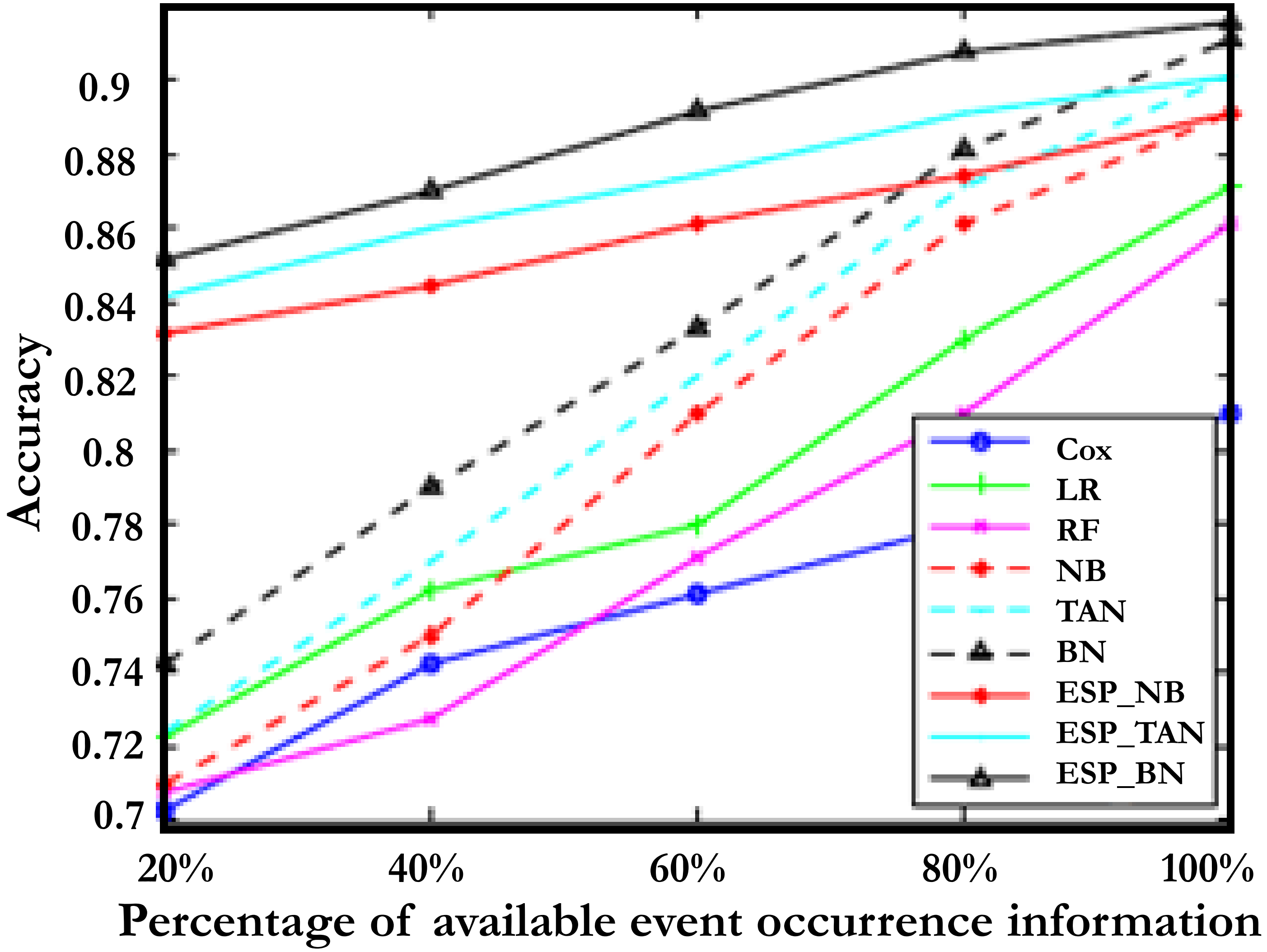
**Probability of Event Occurrence**

$$P\left(y\left(t_f\right)=1 \mid x, t \le t_f\right)=\frac{\boxed{\text{Prior}}\ X\ \boxed{Likelihood}}{P\left(x, t \le t_f\right)}$$

**Extrapolation of Prior**

$$\boxed{Weibull: F\left(t_c\right)=1-e^{-\left(t_c/b\right)^a}} \qquad \boxed{\text{Log-logistic}: F\left(t_c\right)=\frac{1}{1+\left(t_c/b\right)^{-a}}}$$

# Early Stage Prediction



# Data Transformation

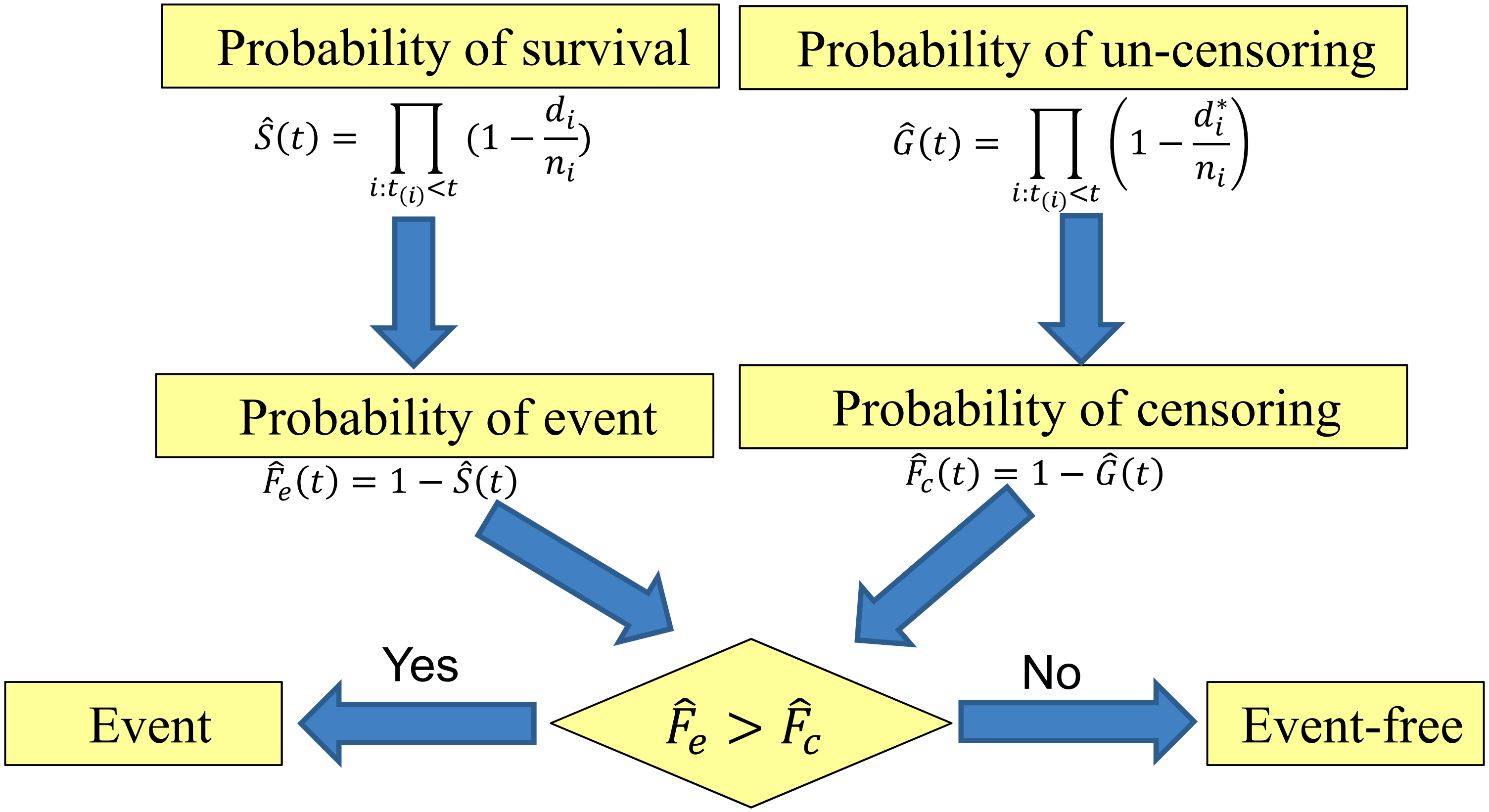
- Two data transformation techniques that will be useful for data pre-processing in survival analysis.
  - Uncensoring approach
  - Calibration
- Transform the data to a more conducive form so that other survival-based (or sometimes even the standard algorithms) can be applied effectively.

# Uncensoring Approach

- The censored instances actually have partial informative labeling information which provides the possible range of the corresponding true response (survival time).
- Such censored data have to be handled with special care within any machine learning method in order to make good predictions.
- Two naive ways of handling such censored data:
  - Delete the censored instances.
  - Treating censoring as event-free.

# Uncensoring Approach I

- For each censored instance, estimate the probability of event and probability of being censored (considering censoring as a new event) using Kaplan-Meier estimator. Give a new class label based on these probability values.



Probability of survival

$\hat{S}(t) = \prod_{i:t_{(i)}<t} (1 - \frac{d_i}{n_i})$

Probability of un-censoring

$\hat{G}(t) = \prod_{i:t_{(i)}<t} \left(1 - \frac{d_i^*}{n_i}\right)$

Probability of event

$\hat{F}_e(t) = 1 - \hat{S}(t)$

Probability of censoring

$\hat{F}_c(t) = 1 - \hat{G}(t)$

$\hat{F}_e > \hat{F}_c$

Yes

Event

No

Event-free

M. J Fard, P. Wang, S. Chawla, and C. K. Reddy, “A bayesian perspective on early stage event prediction in longitudinal data”, TKDE 2016.

# Uncensoring Approach II

- Group the instances in the given data into three categorizes:
  - (i) Instances which experience the event of interest during the observation will be labeled as event.
  - (ii) Instances whose censored time is later than a predefined time point are labeled as event-free.
  - (iii) Instances whose censored time is earlier than a predefined time point,
    - A copy of these instances will be labeled as event.
    - Another copy of the same instances will be labeled as event-free.
    - These instances will be weighted by a marginal probability of event occurrence estimated by the Kaplan-Meier method.

B. Zupan, J. DemsAr, M. W. Kattan, R. J. Beck, and I. Bratko, "Machine learning for survival analysis: a case study on recurrence of prostate cancer", Artificial intelligence in medicine, 2000.

# Calibration

- **Motivation**
  - Inappropriately labeled censored instances in survival data cannot provide much information to the survival algorithm.
  - The censoring depending on the covariates may lead to some bias in standard survival estimators.
- **Approach -** Regularized inverse covariance based imputed censoring
  - Impute an appropriate label value for each censored instance, a new representation of the original survival data can be learned effectively.
  - It has the ability to capture correlations between censored instances and correlations between similar features.
  - Estimates the calibrated time-to-event values by exploiting row-wise and column-wise correlations among censored instances for imputing them.

B. Vinzamuri, Y. Li, and C. K Reddy, “Pre-processing censored survival data using inverse covariance matrix based calibration”, *TKDE* 2017.

# Complex Events

- Until now, the discussion has been primarily focused on survival problems in which each instance can experience only a single event of interest.
- However, in many real-world domains, each instance may experience different types of events and each event may occur more than once during the observation time period.
- Since this scenario is more complex than the survival problems discussed so far, we consider them to be complex events.
  - Competing risks
  - Recurrent events

# Stratified Cox Model

- The stratified Cox model is a modification of the regular Cox model which allows for control by stratification of the predictors which do not satisfy the PH assumption in Cox model.
  - Variables $Z_1, Z_2, \dots, Z_k$ do not satisfy the PH assumption.
  - Variables $X_1, X_2, \dots, X_p$ satisfy the PH assumption.
- Create a single new variable $Z^*$:
  - (1) categorize each $Z_i$; (2) form all the possible combinations of categories;

    (3) the strata are the categories of $Z^*$.
- The general stratified Cox model will be:

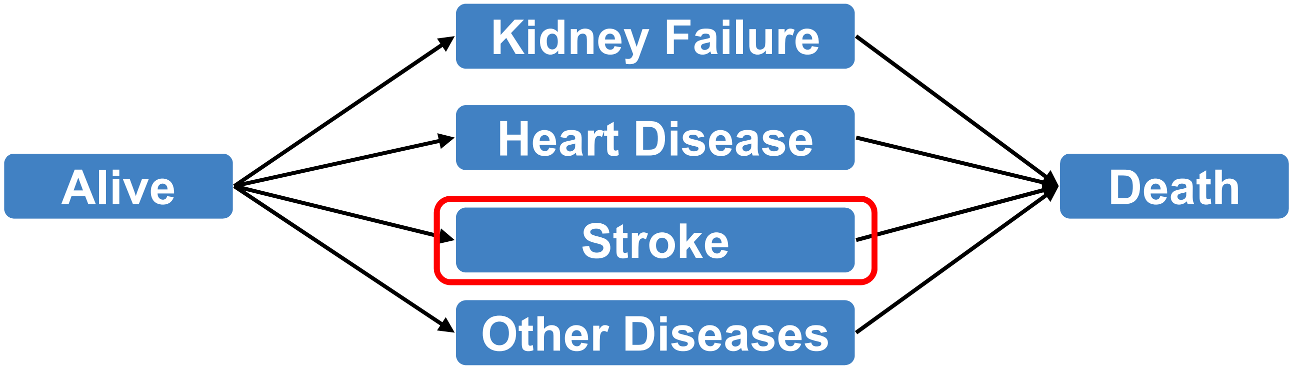
$$h_g(t,X) = \underbrace{\boxed{h_{0g}(\mathrm{t})}}_{\text{Can be different for each strata}} \times \underbrace{\boxed{\exp[\beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p]}}_{\text{Coefficients are the same for each strata}}$$

where $g = 1,2,\cdots,k^*$, strata defined from $Z^*$.

- The coefficients are estimated by maximizing the partial likelihood function obtained by multiplying likelihood functions for each strata.

# Competing Risks

- The competing risks will only exist in survival problems with more than one possible event of interest, but only one event will occur at any given time.



- In this case, competing risks are the events that prevent an event of interest from occurring which is different from censoring.
  - In the case of censoring, the event of interest still occurs at a later time, while the event of interest is impeded.
- Cumulative Incidence Curve (CIC) and Lunn-McNeil (LM)

# Cumulative Incidence Curve (CIC)

- The cumulative incidence curve is one of the main approaches for competing risks which estimates the marginal probability of each event $q$. The CIC is defined as

$$CIC_q(t) = \sum_{j:t_j \leq t} \hat{S}(t_{j-1})\hat{h}_q(t_j) = \sum_{j:t_j \leq t} \hat{S}(t_{j-1})\frac{n_{qj}}{n_j}$$

where

- $\hat{h}_q(t_j)$ represents the estimated hazard at time $t_j$ for event $q$.
- $n_{qj}$ is the number of events for the event $q$ at $t_j$.
- $n_j$ denotes the number of instances who are at the risk of experiencing events at $t_j$.
- $\hat{S}(t_{j-1})$ denotes the survival probability at last time point $t_{j-1}$.

H. Putter, M. Fiocco, and R. B. Geskus, “Tutorial in biostatistics: competing risks and multi-state models”, Statistics in medicine, 2007.

# Lunn-McNeil (LM)

- Lunn-McNeil fits a single Cox PH model which considers all the events $(E_1, \mathrm{E}_2, \dots, \mathrm{E}_\mathrm{c})$ in competing risks rather than separate models for each event.
- LM approach is implemented using an augmented data, in which a dummy variable is created for each event to distinguish different competing risks.

***The augmented data for the $i^{th}$ subject at time $t_i$.***

| ID | Time | Status | $E_1$ | $E_2$ | … | $E_c$ | $X_1$ | … | $X_P$ |
|---|---|---|---|---|---|---|---|---|---|
| i | $t_i$ | $\delta_1$ | 1 | 0 | … | 0 | $X_{11}$ | … | $X_{1P}$ |
| i | $t_i$ | $\delta_2$ | 0 | 1 | … | 0 | $X_{11}$ | … | $X_{1P}$ |
| … | … | … | … | … | … | … | … | … | … |
| i | $t_i$ | $\delta_C$ | 0 | 0 | … | 1 | $X_{11}$ | … | $X_{1P}$ |

Only one of them equals to 1.

**Dummy variables**

**Features**

M. Lunn and D. McNeil, “Applying Cox regression to competing risks”, Biometrics, 1995.

# Recurrent Events

- In many application domains, the event of interest can occur for each instance more than once during the observation time period.

- In survival analysis, we refer to such events which occur more than once as recurrent events, which is different from the competing risks.
  - If all the recurring events for each instance are of the same type.
    - Method: counting process (CP) algorithm.
  - If there are different types of events or the order of the events is the main goal.
    - Method: methods using stratified Cox (SC) approaches, including stratified CP, Gap Time and Marginal approach.

# Software Resources

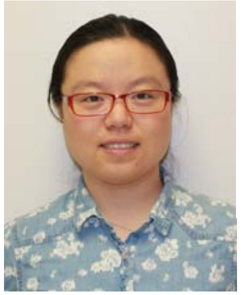
| Algorithm | Software | Language | Link |
|---|---|---|---|
| Kaplan-Meier | survival | R | https://cran.r-project.org/web/packages/survival/index.html |
| Nelson-Aalen | | | |
| Life-Table | | | |
| Basic Cox | survival | R | https://cran.r-project.org/web/packages/survival/index.html |
| TD-Cox | | | |
| Lasso-Cox | fastcox | R | https://cran.r-project.org/web/packages/fastcox/index.html |
| Ridge-Cox | | | |
| EN-Cox | | | |
| Oscar-Cox | RegCox | R | https://github.com/MLSurvival/RegCox |
| CoxBoost | CoxBoost | R | https://cran.rproject.org/web/packages/CoxBoost/ |
| Tobit | survival | R | https://cran.r-project.org/web/packages/survival/index.html |
| BJ | bujar | R | https://cran.rproject.org/web/packages/bujar/index.html |
| AFT | survival | R | https://cran.r-project.org/web/packages/survival/index.html |

# Software Resources

| Algorithm | Software | Language | Link |
|---|---|---|---|
| Baysian Methods | BMA | R | https://cran.rproject.org/web/packages/BMA/index.html |
| RSF | randomForestSRC | R | https://cran.rproject.org/web/packages/randomForestSRC/ |
| BST | ipred | R | https://cran.rproject.org/web/packages/ipred/index.html |
| Boosting | mboost | R | https://cran.rproject.org/web/packages/mboost/ |
| Active Learning | RegCox | R | https://github.com/MLSurvival/RegCox |
| Transfer Learning | TransferCox | C++ | https://github.com/MLSurvival/TransferCox |
| Multi-Task Learning | MTLSA | Matlab | https://github.com/MLSurvival/MTLSA |
| Early Prediction | ESP | R | https://github.com/MLSurvival/ESP |
| Uncensoring | | | |
| Calibration | survutils | R | https://github.com/MLSurvival/survutils |
| Competing Risks | survival | R | https://cran.r-project.org/web/packages/survival/index.html |
| Recurrent Events | survrec | R | https://cran.r-project.org/web/packages/survrec/ |

# Acknowledgements

## Graduate Students


Ping Wang

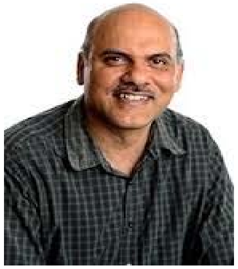
Bhanu Vinzamuri

Mahtab Fard

Vineeth Rakesh

## Collaborators

Jieping Ye
Univ. of Michigan


Sanjay Chawla
Univ. of Sydney

Charu Aggarwal
IBM Research

Naren Ramakrishnan
Virginia Tech

## Funding Agencies

NSF

NIH

DEPARTMENT OF TRANSPORTATION
UNITED STATES OF AMERICA

SUSAN G. KOMEN FOR THE CURE

BlueCross BlueShield

# Thank You

## Questions and Comments

Feel free to email questions or suggestions to

reddy@cs.vt.edu

http://www.cs.vt.edu/~reddy/