SOCR 2020 MDP Project Summaries

The one-page summaries below describe the main SOCR MDP R&D Projects for 2020 (January-December)

GDrive: https://drive.google.com/drive/folders/1PBsa89I9EEiE_6aKlya-NfYtGrdB21WJ
GSlides: https://drive.google.com/open?id=1dWGyKrc6nI3_LEmfLslEF0kDNRswPhe64pDzxE48UaA
SOCR Project Leaders:

- Programming: Alex Kalinin, Syed Husain, Ivo Dinov
- Methods (CBDA & DataSifter): Simeone Marino & Nina Zhou
- Analytics: Jess Wild, Jared Chai, Nina Zhou, Ivo Dinov
- Data Science Fundamentals: Yongkai, Zhe, Ivo Dinov

SOCR Trainees/Students
TBD

Project Summaries

Project Area	Skills	Likely Majors
Programming Subteam: SOCRAT (Charts, Wrangler, Modeler, Analyses, Tools) (2-3 students)	UI/UX design, HTML5, JavaScript, Adobe Illustrator, Canvas	Computer Science (CSE/CS-LSA) School of Information (SI)
TensorFlow.JS	https://js.tensorflow.org https://js.tensorflow.org/api/latest/ https://codepen.io/pen?&editors=1011	Computer Science (CSE/CS-LSA)
Methods (CBDA & DataSifter) (4 students) DataSifter& CBDA	Technical math background, R-computing	
Analytics (4 students) TDA Biomed/Health Applications (see Case-Studies)	R/Python, statistical modeling, high-throughput data analytics, machine learning	Statistics, Biostatistics, Bioinformatics Math Computer Science (CSE/CS-LSA)
Data Science Fundamentals (New sub-team – will work with the PI directly) (4 students)	Information measures, entropy KL divergence, PDEs, Dirac's bra-ket operators. See *The Enigmatic Kime: Time Complexity in Data Science* at the University of Michigan Institute for Data Science (MIDAS) Seminar Series, Slidedeck, YouTube video of this seminar	Physics, math or engineering background is preferred
498/599 Programming 3-6 students will tackle interesting ML, web-services and Visualization problems DVT, BlueML,	See above and **TensorFlow.JS**	Computer Science (CSE/CS-LSA) Statistics, Biostatistics, Bioinformatics Math, Physics, Engineering School of Information (SI)

SOCR 2020 MDP Project: SOCRAT

SOCR Project Leaders: Alex Kalinin / Syed Husain / Ivo Dinov
Website: http://socr.umich.edu/HTML5/SOCRAT/
GitHub: https://github.com/SOCR/SOCRAT
Training Modules: https://github.com/SOCR/socr-tutorials
GDrive: https://drive.google.com/drive/folders/1UrNpNDI5sWoXW61YwP02NSv3PBbxfvpC

Description

The Statistics Online Computational Resource Analytics Toolbox (SOCRAT) is a Dynamic Web Toolbox for Interactive Data Processing, Analysis, and Visualization. It's purely built using HTML5 standards and JavaScript (core library) as well as node.js,

Student Skills

- EECS, Computer Science (CSE/CS-LSA) and School of Information (SI)
- UI/UX design, HTML5, JavaScript

Project Goals

- Go through the Training Modules, practice HTML/JS/Angular/Node programming
- Get your GitHub domain going and pull current SOCRAT branch
- Choose 1-2 deliverables, go over current design, start expansion, include unit tests, pilot development
- Coordinate with team

Deliverables

- Expanded collection of Charts
- Expanded collection of Data-Modelers
- Expanded collection of (parametric and non-parametric) Statistical Analyses
- Expanded collection of machine learning classification, prediction, clustering and analytics modules.

Team Activities

- Weekly team BlueJeans meetings
- Code review (pull/push Github requests)
- Join/present the SOCR All-Hands Calls (twice a month, SOCR BlueJeans channel)

References

- Review the websites
- Alexandr A. Kalinin, Selvam Palanimalai, and Ivo D. Dinov. 2017. SOCRAT Platform Design: A Web Architecture for Interactive Visual Analytics Applications. In Proceedings of HILDA'17, Chicago, IL, USA, May 14, 2017, 6 pages. DOI:10.1145/3077257.3077262

SOCR 2020 MDP Project: Methods: CBDA

SOCR Project Leaders: Simeone Marino

Website: http://socr.umich.edu/HTML5/CBDA/
GitHub: https://github.com/SOCR/CBDA
C-RAN Package: https://cran.r-project.org/web/packages/CBDA
Training Modules: http://socr.umich.edu/HTML5/CBDA/
GDrive: https://drive.google.com/drive/folders/1hjwtgz64A_lUsnRK1gv7mGSJ3HdBHaRW

Description

The SOCR Compressive Big Data Analytics (CBDA) Project conducts research and implements efficient computational algorithms to tackle the Big Data problems of representation and analysis of complex heterogeneous information. Big Data cannot be loaded and processed as a whole. CBDA implements a real-time efficient divide-and-conquer strategy to deconstruct the Big Data into meaningful pieces of information that can be eventually reconstructed for actionable knowledge and predictive analytics.

Student Skills

- Probability, stats, math, numerical methods, optimization
- R programming with RStudio (IDE) experience

Project Goals

- Go through the provided materials and references
- Download the CBDA Package
- Practice with test-cases (https://umich.instructure.com/courses/38100/files/folder/Case_Studies)
- Identify specific R&D direction to go deeper into an meaningfully contribute to CBDA
- Coordinate with team

Deliverables

- New CBDA methods
- Expanded collection of machine learning forecasting, prediction, classification, clustering methods to expand the available CBDA algorithms
- Release new versions of CBDA R package and publish CBDA #2 manuscript
- Python/Perl scripts to speed up the subsampling strategy with Big Data > 100Gb-1Tb

Team Activities

- Weekly team face-to-face/BlueJeans meetings
- Code review (pull/push Github requests)
- Join/present the SOCR All-Hands Calls (twice a month, SOCR BlueJeans channel)

References

- Review the websites
- Marino S, Xu J, Zhao Y, Zhou N, Zhou Y, Dinov, ID. (2018) Controlled feature selection and compressive big data analytics: Applications to biomedical and health studies, PLoS ONE 13(8): e0202674, DOI: 10.1371/journal.pone.0202674

SOCR 2020 MDP Project: Methods: DataSifter

SOCR Project Leaders: Nina Zhou / Simeone Marino

Website: http://DataSifter.org
GitHub: https://github.com/SOCR/DataSifter
C-RAN Package: (lite version pending)
Training Modules: http://DataSifter.org
GDrive: https://drive.google.com/drive/folders/1jVT5pTa_n8xHjUszn1u5gwTzyvPLtszj

Description

The SOCR DataSifter is a novel method, and an efficient R package, for on-the-fly de-identification of structured Clinical/Epic/PHI data. This approach provides complete administrative control over the risk of data identification when sharing large clinical cohort-based medical data. At the extremes, the data-governor may specify that either null data or completely identifiable data is generated and shared with the data-requester. This decision may be based on data-governor determined criteria about access level, research needs, etc. For instance, to stimulate innovative pilot studies, the data office may dial up the level of protection (which may naturally devalue the information content in the data), whereas for more established and trusted investigators, the data governors may provide a more egalitarian dataset that balances preservation of information content and sensitive-information protection.

Student Skills

- Probability, stats, math, numerical methods, optimization
- R programming with RStudio (IDE) experience

Project Goals

- Go through the provided materials and references
- Download the DataSifter-lite Package
- Practice with test-cases (https://umich.instructure.com/courses/38100/files/folder/Case_Studies)
- Identify specific R&D direction to go deeper into and meaningfully contribute to DataSifter methods, implementation and/or validation
- Coordinate with team

Deliverables

- New DataSifter methods/algorithms (e.g., addressing text, time-varying, graph data organizations)
- Release new versions of DataSifter R package
- Coordinate/support collaborators

Team Activities

- Weekly team face-to-face/BlueJeans meetings
- Code review (pull/push Github requests)
- Join/present the SOCR All-Hands Calls (twice a month, SOCR BlueJeans channel)

References

- Review the websites
- Marino, S, Zhou, N, Zhao, Yi, Wang, L, Wu Q., and Dinov, ID. (2018) DataSifter: Statistical Obfuscation of Electronic Health Records and Other Sensitive Datasets, Journal of Statistical Computation and Simulation, pp: 1-23, DOI: 10.1080/00949655.2018.1545228.

SOCR 2020 MDP Project: Webapps & Data Analytics

SOCR Project Leaders: Jared Chai, Simeone Marino, Ivo Dinov

Website: <many, e.g., http://socr.umich.edu/HTML5>
GitHub: https://github.com/SOCR <many, e.g., https://github.com/SOCR/ALS_PA>
Training Modules: http://DSPA.predictive.space
GDrive: https://drive.google.com/drive/folders/1sN1fLYA0oLf1I4e1REJRthaMD0jXBs7w

Description

The SOCR Webapp & Data Analytics projects are focused on interrogating massive amounts of complex biomedical and health data. Each project tackles multiple case-studies using R/RMD/RStudio, RShiny Services, and Python/Jupyter Notebook and the SOCR-Flux Compute Server (https://docs.google.com/document/d/1UmBq_BMiMeUcijvKUCzPeG3tKZaWkinVtKrVWenPK1Y). The webapp development will use R markdown notebook, RShiny web-servers, and Google BigQuery Datasets.

Student Skills

- Biostats, quantitative analytics, probability, stats, math, numerical methods, optimization
- R programming with RStudio (IDE) experience, and/or Python/Jupyter Notebook

Project Goals

- Go through the provided materials and references
- Review the SOCR Data Analytics Publications (http://socr.umich.edu/people/dinov/publications.html)
- Review the SOCR R-environment (https://drive.google.com/file/d/1-u9adsMIYmMkcPD9W_6BbfC1IMETsHF_/)
- Practice with test-cases (https://umich.instructure.com/courses/38100/files/folder/Case_Studies)
- Identify specific case-study and an R&D direction to go deeper into an meaningfully contribute
- Coordinate with team

Deliverables

- New SOCR end-to-end data analytics protocols
- Analytical results, abstracts, publications, presentations, research findings, etc.
- MIMIC-III analytics
- Baby-growth and mother-obesity relations
- Data Value Metric (DVM)
- European Economics Indicators (longitudinal analytics)
- 2D, 3D, 4D Visualization of complex data
- Coordinate/support collaborators

Team Activities

- Weekly team face-to-face/BlueJeans meetings
- Code review (pull/push Github requests)
- Join/present the SOCR All-Hands Calls (twice a month, SOCR BlueJeans channel)

References

- Review the websites and listed resources
- https://shiny.med.umich.edu/apps/dinov/RShinyApp_PIPM/
- https://socr.shinyapps.io/RShinyApp_PIPM/

SOCR 2020 MDP Project: Data Analytics - MIMIC-III

SOCR Project Leaders: Jess WIld, Jared Chai, Ivo Dinov

Website: TBD
GitHub: https://github.com/SOCR
Training Modules:
- Data Science & Predictive Analytics: http://DSPA.predictive.space
- Previous SOCR Data Analytics Publications: http://socr.umich.edu/people/dinov/publications.html
- Gaining access to the dataset requires an online training module; see onboarding materials below https://drive.google.com/drive/u/1/folders/1Y6Yqq1CuTkHQ5rZg-C9r8_je18nM886l

GDrive: https://drive.google.com/drive/folders/1sN1fLYA0oLf1I4e1REJRthaMD0jXBs7w

Description

This SOCR Data Analytics project is focused on interrogating the MIMIC-III database, a large collection of ~43,000 critical care patients from an ICU in Boston, MA. We will use R/RStudio, Python/Jupyter, and the SOCR-Flux Compute Server[1] to digest the vital signs, laboratory results, free-text data, and waveforms available in this unique dataset and predict clinical outcomes via statistical modeling tools.

[1]SOCR-Flux Compute server:
https://docs.google.com/document/d/1UmBq_BMiMeUcijvKUCzPeG3tKZaWkinVtKrVWenPK1Y

Student Skills
- Biostats, quantitative analytics, probability, stats, math, numerical methods
- Programming experience in R (with RStudio) or Python (with Jupyter Notebook)
- Relational databases & structured query language (SQL)

Project Goals
- Review the provided materials and references (see above)
- Request access to the MIMIC-III dataset (https://mimic.physionet.org/gettingstarted/access/)
    - This involves an online but comprehensive human subjects research ethics course
- Practice with demo dataset (https://physionet.org/works/MIMICIIIClinicalDatabaseDemo/) and the MIMIC Query Builder (https://querybuilder-lcp.mit.edu/dashboard.cgi)
- Identify specific research aims and questions of interest to the team
- Coordinate with team to create a reproducible, accessible answer to these specific aims

Deliverables
- New SOCR end-to-end data analytics protocols
- Data extraction & time-alignment tools for the MIMIC-III dataset
- Build statistical models to predict meaningful clinical outcomes
- Analytical results, abstracts, publications, presentations, research findings, etc.
- Visualization of complex, multidimensional data

Team Activities
- Weekly team face-to-face/BlueJeans meetings
- Code review (pull/push Github requests)
- Join/present the SOCR All-Hands Calls (twice a month, SOCR BlueJeans channel)

SOCR 2020 MDP Project: SOCR TensorFlow/TensorBoard Apps

SOCR Project Leader: Syed Husain, Alex Kalinin, Chiranjeevi Vegi <vegi@umich.edu>, Ivo Dinov

Website: http://socr.umich.edu/HTML5/SOCR_TensorBoard_UKBB
GitHub: https://github.com/SOCR/97-tensorflowjs-quick-start
Training Modules: https://js.tensorflow.org/tutorials/
GDrive: https://drive.google.com/drive/folders/1wJY8539tpLmYiJc_vKZvI6oDVDAHTQu9

Description

The SOCR TensofFlowJS/TensorBoardJS project aims to design, built, validate and release new webapps based on the ML TensorFlow framework. For example, students will dive deep into TensorFlow.JS (https://js.tensorflow.org, https://js.tensorflow.org/api/latest/, https://codepen.io/pen?&editors=1011) and TensorBoard.JS (https://github.com/tensorflow/tensorboard, https://www.tensorflow.org/guide/summaries_and_tensorboard).

Student Skills

- EECS, Computer Science (CSE/CS-LSA) and School of Information (SI)
- AngularJS, TensorFlowJS, TensorBoard, JavaScript, HTML5

Project Goals

- Go through the Training Modules, practice HTML/JS/Angular/Node programming
- Get your GitHub domain going and start pilot testing various applications
- Use SOCR Data to experiment
- Review Vegi's SOCR t-SNE TensorFlow Webapp (http://socr.umich.edu/HTML5/SOCR_TensorBoard_UKBB)
- Coordinate with team
- Rapid RDD (research, development and deployment) is needed in this project

Deliverables

- 2-5 new SOCR TF/TB Apps
- ...

Team Activities

- Weekly team BlueJeans meetings
- Code review (pull/push Github requests)
- Join/present the SOCR All-Hands Calls (twice a month, SOCR BlueJeans channel)

References

- Review the websites

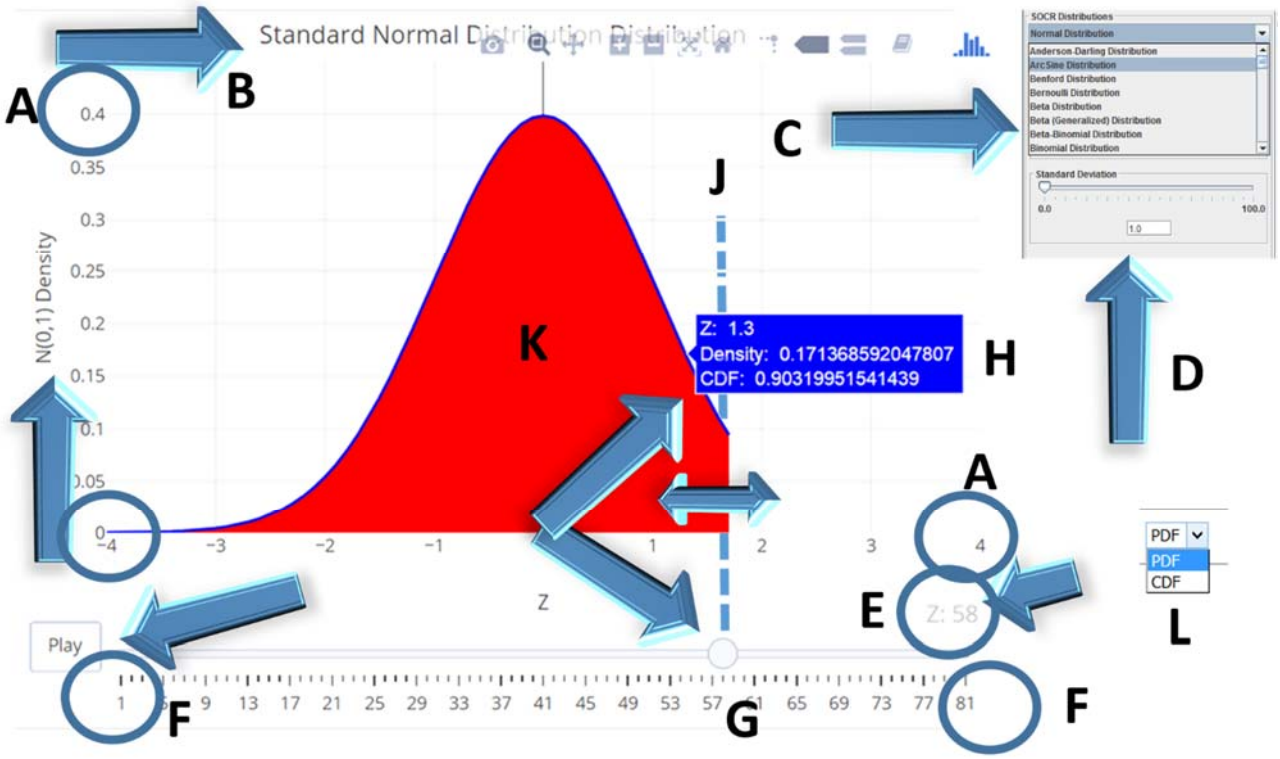
SOCR 2020 MDP Project:
Interactive Graphical Probability Distribution Calculator

SOCR Project Leader: Jared Chai, Ivo Dinov

Website: http://Distributome.org
GitHub: https://github.com/distributome
Training Modules: https://github.com/SOCR/socr-tutorials & http://dspa.predictive.space/
GDrive: https://drive.google.com/drive/folders/184p8VNSOumYEG_SOxlo4MyLVtanq9xLY
Goal: Stand-alone RMD source and a demo HTML (RMD-output) that address the goal of the challenge, provide the desired functionality, and implement it efficiently, without any back-end support (e.g., no shiny server apps).
Deliverables: Stand-alone RMD source and a demo HTML (RMD-output) that address the goal of the challenge, provide the desired functionality, and implement it efficiently.
Background:

- Review DSPA Chapter 2 (for probability distributions) (Links to an external site.) and Chapter 5 (for plot_ly demos) (Links to an external site.)
- Review, experiment and play with the Probability Distributome Calculators (Links to an external site.). Try at least 2-dozen distributions - what works well and what can be improved there? Mind the selection of parameters and the choice for function to plot (PDF, CDF).

Desired Functionality: The schematic below illustrates the core functionality of the interactive probability distribution calculator. Be creative in your solution.



- Include a drop-down list for the user to select the distribution
- Include an effective strategy to specify the parameters of the selected distribution, mind that the parameter number, interpretation and values will be different for different distributions.
- Make sure you keep the interactive aspect of the interface (*plot_ly* style interactivity)
- Make sure all axes are appropriately scaled, labeled and drawn.

Functionality Annotations

- **A**: x and y axes ranges and labels
- **B**: Appropriate title (placed to avoid overlaps
- **C**: Selection of the specific distribution - should include at least 20 distributions, see class notes the Distributome Calculators.
- **D**: appropriate section of the specific distribution parameters

- **E**: cut off of the critical value (Z)
- **F**: Ranges of the animation slider should match the x-axis range (Z-values range)
- **G**: Play button and the animation point provide the manual user control over the critical value cut off
- **H**: Report the appropriate Z-values, density curve height, and cumulative distribution up to Z (i.e., P(X<Z))
- **J**: There should be a light-colored vertical line at the animation index == Z-value and extending up to the corresponding density height
- **K**: shaded area represents the integral CDF value .....
- **L**: Drop-down selected for plotting PDF, CDF or inverse-CDF (quantile) function to plot.

Starting R Code: The basic skeleton of one solution (using "**plot_ly**") is included below. Many solutions are possible and you can start with anything you like, including this initial script.

```
library(magrittr); library(plotly)
select the right user-specified distribution (drop down list)
# Assuming Std Normal N(0,1) going down
# define the range
z<-seq(-4, 4, 0.1)
# points from -4 to 4 in 0.1 steps
# Define the quantile levels for the inverse-CDF (quantile) function)
q<-seq(0.001, 0.999, 0.01)
# probability quantile values from 0.1% to 99.9% in 0.1% steps
# define a DF containing Z, PDF and CDF
dStandardNormal <- data.frame(Z=z, Density=dnorm(z, mean=0, sd=1),
Distribution=pnorm(z, mean=0, sd=1))
# define an index feature
dStandardNormal$ID <- seq.int(nrow(dStandardNormal))
# Aggregate frames for interactive plot
aggregate_by <- function(dataset, feature) {
feature <- lazyeval::f_eval(feature, dataset)

levels <- plotly:::getLevels(feature)
aggData <- lapply(seq_along(levels), function(x) {
cbind(dataset[feature %in% levels[seq(1, x)], ], frame = levels[[x]])
})
dplyr::bind_rows(aggData)
}

# Apply the aggregate to ID index
dStandardNormal <- dStandardNormal %>% aggregate_by(~ID)

# generate the Plot_ly object
plotMe <- dStandardNormal %>% plot_ly( x = ~Z, y = ~Density, frame = ~frame,
type = 'scatter', mode = 'lines', fill = 'tozeroy', fillcolor="red",
line = list(color = "blue"), text = ~paste("Z: ", Z, "
Density: ", Density, "CDF: ", Distribution), hoverinfo = 'text' ) %>%
layout( title = "Standard Normal Distribution Distribution",
# Specify the right distribution and its parameters!!!
yaxis = list( title = "N(0,1) Density", range = c(0,0.45),
zeroline = F, tickprefix = "" # density value
),
xaxis = list( title = "Z", range = c(-4,4), zeroline = T, showgrid = T ) ) %>%
animation_opts( frame = 100, transition = 1, redraw = FALSE ) %>%
animation_slider( currentvalue = list( prefix = "Z: " ) )
# display interactive plot
plotMe
```

The optimal solution will include RMD (source) and HTML output (webapp).

SOCR 2020 MDP Project: Data Science Fundamentals

SOCR Project Leader: Yongkai, Zhe, Jinwen, Yufei, Milen Velev, Ivo Dinov

Website: http://tciu.predictive.space
GitHub: NA
Training Modules: ODE/PDE, Kaluza-Klein Theory (https://en.wikipedia.org/wiki/Kaluza–Klein_theory)
GDrive: https://drive.google.com/drive/folders/1PMMBR2bzBPubYMpywLkcTkJPyxOKQ4Aq

Description

The SOCR Data Science Fundamentals project will explore new theoretical representation and analytical strategies to understand large and complex data. It will utilize information measures, entropy KL divergence, PDEs, Dirac's bra-ket operators. This fundamentals of data science research project will explore time-complexity and inferential uncertainty in modeling, analysis and interpretation of large, heterogeneous, multi-source, multi-scale, incomplete, incongruent, and longitudinal data.

See The Enigmatic Kime: Time Complexity in Data Science (https://midas.umich.edu/event/midas-seminar-series-presents-ivo-d-dinov-phd-university-of-michigan/) at the University of Michigan Institute for Data Science (MIDAS) Seminar Series, Slidedeck (http://socr.umich.edu/docs/uploads/2018/Dinov_TCIU_Kime_MIDAS_2018.pdf).

Student Skills

- Physics, math or engineering background
- R programming with RStudio (IDE) experience, and/or Python/Jupyter Notebook

Project Goals

- Go through the provided materials and references
- Review the current platform (will be provided)
- Perform 3D and 4D Plot_Ly visualization of complex manifolds, including 5D space-kime and 2D-curved Kime.
- Identify specific case-study and an R&D direction to go deeper into an meaningfully contribute
- Coordinate with team

Deliverables

- Visualization protocols
- Math proofs of various physics properties in 5D Minkowski spacekime

Team Activities

- Weekly team face-to-face/BlueJeans meetings
- Code review Join/present the SOCR All-Hands Calls (twice a month, SOCR BlueJeans channel)

Key points

- *What is the problem?* Use complex-time physics to formulate data science theory & practice
- *Why is it important?* There is currently no canonical theory for Big Data discovery science
- *What is the SOCR Solution?* Blend transdisciplinary knowledge to build a new Data Analytic method
- *It's real; here it is (in a pilot form) … demo …* See TCIU Video
- *Why should you consider joining this SOCR-MDP Project?* High-risk/high-potential yield project.

References

- Review the websites and listed resources
- TCIU Website: http://tciu.predictive.space/
- TCIU GitHub: https://github.com/SOCR/TCIU/
- www.SpaceKime.org

SOCR 2020 MDP Project: Synthetic Data

SOCR Project Leaders: Simeone Marino, Johnny Liu, Ronak Shetty, Ivo Dinov
Website: http://gray-rain.com
GitHub: ...
Training Modules: https://github.com/SOCR/socr-tutorials
GDrive: https://drive.google.com/drive/folders/1hvzoihZMO-6uJIPbObI4rVc_QDp9i5tD

Description

This project involves developing a Virtual Hospital (VH) and Synthetic Patient (SP) capability to simulate realistic electronic health record including categorical, discrete, continuous, imaging, text and biomedical specimen data. This relates to the unstructured-DataSifter, synthetic text generation, text-obfuscation, and text-mining/inference.

Student Skills

- EECS, Stats/Biostats/Math, Computer Science (CSE/CS-LSA) and School of Information (SI)
- UI/UX design, HTML5, JavaScript

Project Goals

- Go through the Training Modules, practice HTML/JS/Angular/Node programming
- Get your GitHub domain going
- Choose 1-2 deliverables, go over current design, start expansion, include unit tests, pilot development
- Coordinate with team

Deliverables

- Learn the current state of the project (Simeone)
- Examine the prior datasets (NHANES, MIMIC, etc.)
- Choose categorical, discrete, continuous, imaging, text and biomedical specimen data and illustrate examples
- Experiment and demo 1,000 VH/SP cases
- Validate VH/SP using machine learning classification, prediction, clustering and analytics modules.

Team Activities

- Weekly team meetings
- Code review (pull/push Github requests)
- Join/present the SOCR All-Hands Calls (twice a month, SOCR BlueJeans channel)

References

- **synthpop** package provides synthetic –data generation: https://www.r-bloggers.com/generating-synthetic-data-sets-with-synthpop-in-r/ and saving the synthText in diff formats. See this paper.
- **stringdist** package allows us to compare strings/text: https://cran.r-project.org/web/packages/stringdist
- GAN (generative Adversarial Network) models for synthetic image generation: https://www.r-bloggers.com/conditional-generative-adversarial-network-with-mxnet-r-package/
    - Also see: https://becominghuman.ai/generative-adversarial-networks-for-text-generation-part-1-2b886c8cab10
- See this medical text-GAN mtGAN (Python) paper: https://arxiv.org/pdf/1812.02793.pdf
- Autoencoder approach: https://github.com/stas-semeniuta/textvae and https://www.aclweb.org/anthology/D17-1066.pdf

SOCR 2020 MDP Project: BlueML, DVT

SOCR Project Leaders: Syed Husain, Ivo Dinov
Website: http://socr.umich.edu/HTML5/Dashboard/
GitHub: https://github.com/SOCR
Training Modules: https://github.com/SOCR/socr-tutorials
GDrive:

Description

The SOCR BlueML project is purely build using HTML5 standards and JavaScript and includes a core library for applying machine learning to high sampling-rate longitudinal data like waveform EEG and EKG. For example, students will dive deep into TensorFlow.JS (https://js.tensorflow.org, https://js.tensorflow.org/api/latest/, https://codepen.io/pen?&editors=1011) and TensorBoard.JS (https://github.com/tensorflow/tensorboard, https://www.tensorflow.org/guide/summaries_and_tensorboard). Another example is the Dynamic Visualization Toolkit (https://github.com/SOCR/DVT).

Student Skills

- EECS, Computer Science (CSE/CS-LSA) and School of Information (SI)
- UI/UX design, HTML5, JavaScript

Project Goals

- Go through the Training Modules, practice HTML/JS/Angular/Node programming
- Get your GitHub domain going and pull current BlueML branch
- Choose 1-2 deliverables, go over current design, start expansion, include unit tests, pilot development
- Coordinate with team

Deliverables

- …
- ...

Team Activities

- Weekly team BlueJeans meetings
- Code review (pull/push Github requests)
- Join/present the SOCR All-Hands Calls (twice a month, SOCR BlueJeans channel)

References

- Review the websites