ML algorithms with guarantees:
Reproducibility and Speed
Jean Honorio
Example 1: cocaine addiction

- **Interpretation**
  - cocaine addiction affects brain function in the long term

- **Prediction**
  - predict if a person is cocaine addicted or control: 84.6%
Example 2: congressional voting

By using *game theory*, we found out which senators are the *most influential*
Many other examples

- *Learning graphs from data*, e.g., graphical models, game theory

- *Regression and classification*, e.g., linear, nonparametric

- *Matrix factorization*, e.g., Netflix

\[ \begin{array}{c}
X \rightarrow y \\
\text{genetics data} \quad +1: \text{person has cancer} \\
\text{\quad \quad -1: person is healthy}
\end{array} \]
Many other examples

- **Clustering**, e.g., $k$-means, nonparametric

- **Structured prediction**, e.g., natural language processing

\[ \mathbf{x} = \text{“the old cat is white”} \quad \Rightarrow \quad y = \]
Reproducibility

Train to perform well on *training* data

We want to perform well on *testing* data

Training Data 80%

Testing Data 75%
Train to perform well on *training* data

- Training Data 1: 80%
- Training Data 2: 70%
- Training Data 3: 90%

We want to perform well on *testing* data

- Testing Data 1: 75%
- Testing Data 2: 95%
- Testing Data 3: 30%

— **Big data**: more features, more samples needed?
Train to perform well on \textit{training} data

We want to perform well on \textit{testing} data

\begin{itemize}
  \item \textbf{Training Data 1}
  \item \textbf{Training Data 2}
  \item \textbf{Training Data 3}
  \item \textbf{Testing Data 1}
  \item \textbf{Testing Data 2}
  \item \textbf{Testing Data 3}
\end{itemize}

\textbf{Big data}: more features, more samples needed?
Speed

• Classical convergence theory
  • starting close to optimal
  • asymptotic convergence

– Start arbitrarily far from the optimal?
– After a finite number iterations?
– Complex objectives
  • Noisy
  • Nonsmooth
  • Nonconvex
  • NP-hard
• Areas:
  – Statistical learning theory
  – Optimization
  – Information theory

• We will:
  – prove theorems
  – write code
  – execute experiments