GREENBERG QUINLAN ROSNER RESEARCH

May 21, 2009

Stata for micro targeting using C++ and ODBC

Masahiko Aida

The Aim of the Research

- Outline of presentation
  - What is micro-targeting?
  - Work flows

- Utilizing best part of three computing platform
  - Stata for its flexibility.
  - SQL server for handling large data.
  - C++ for fast complex calculation.

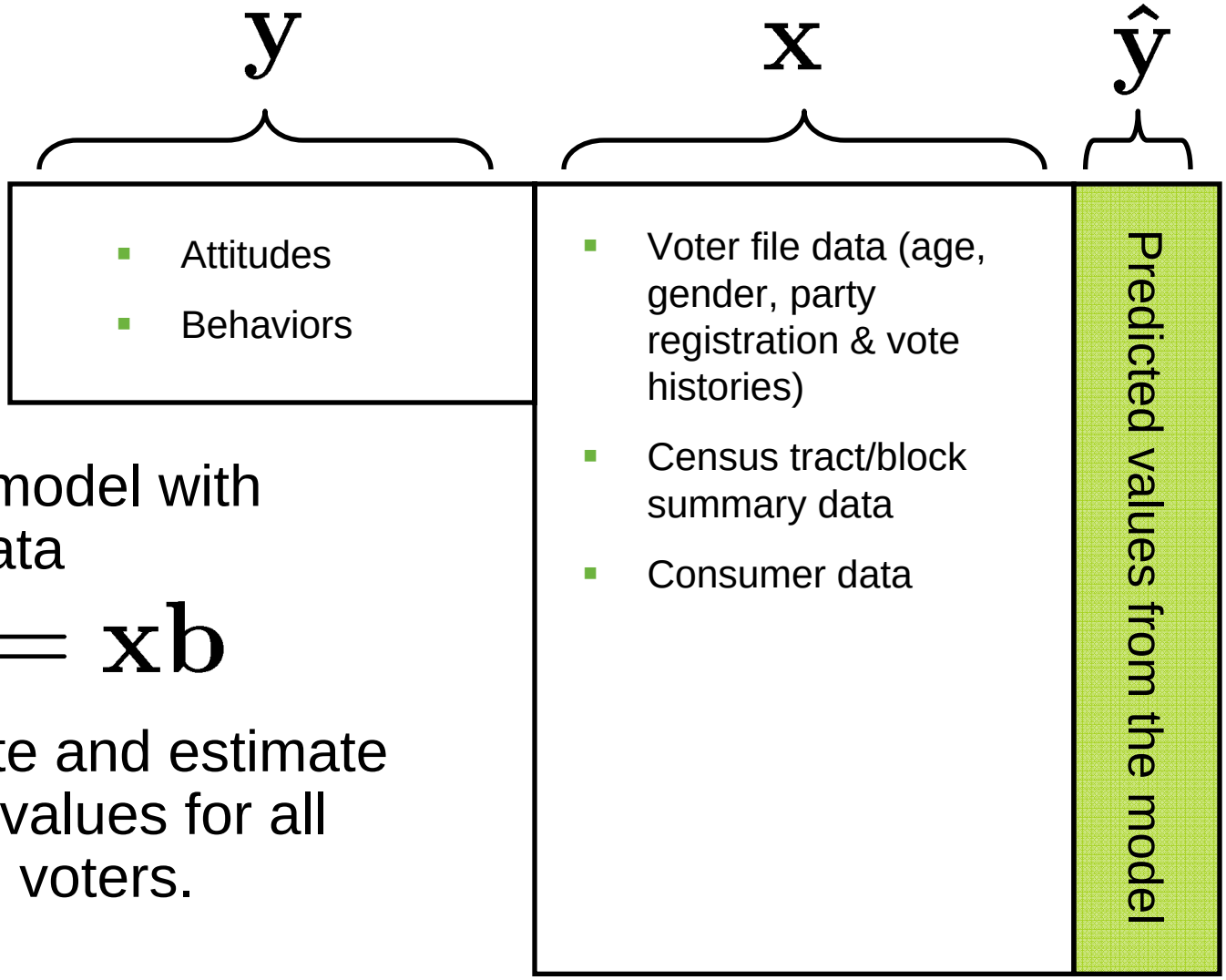
What is targeting?

- Political campaigns targeted voters to convey candidate's views or to encourage voting (GOTV).

- Targeting using macro level data
  - Buy ad spots by media market (defined by county).
  - Send canvassers to particular precincts.

- Nature of macro level targeting
  - Pros: Candidate-level breakdown of votes are only reported by precinct level.
  - Cons: Not very efficient.

How does Micro-targeting Work?



1. Create a model with training data

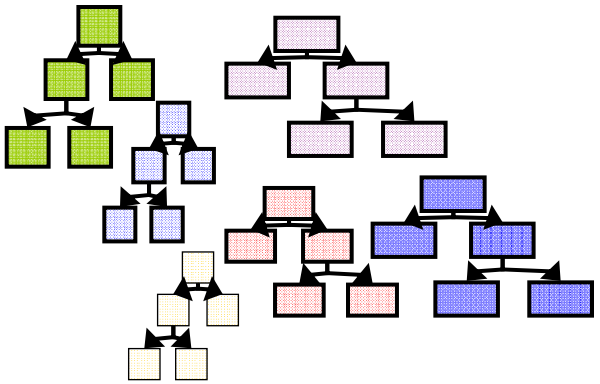
$$\mathbf{y} = \mathbf{xb}$$

2. Extrapolate and estimate predicted values for all registered voters.

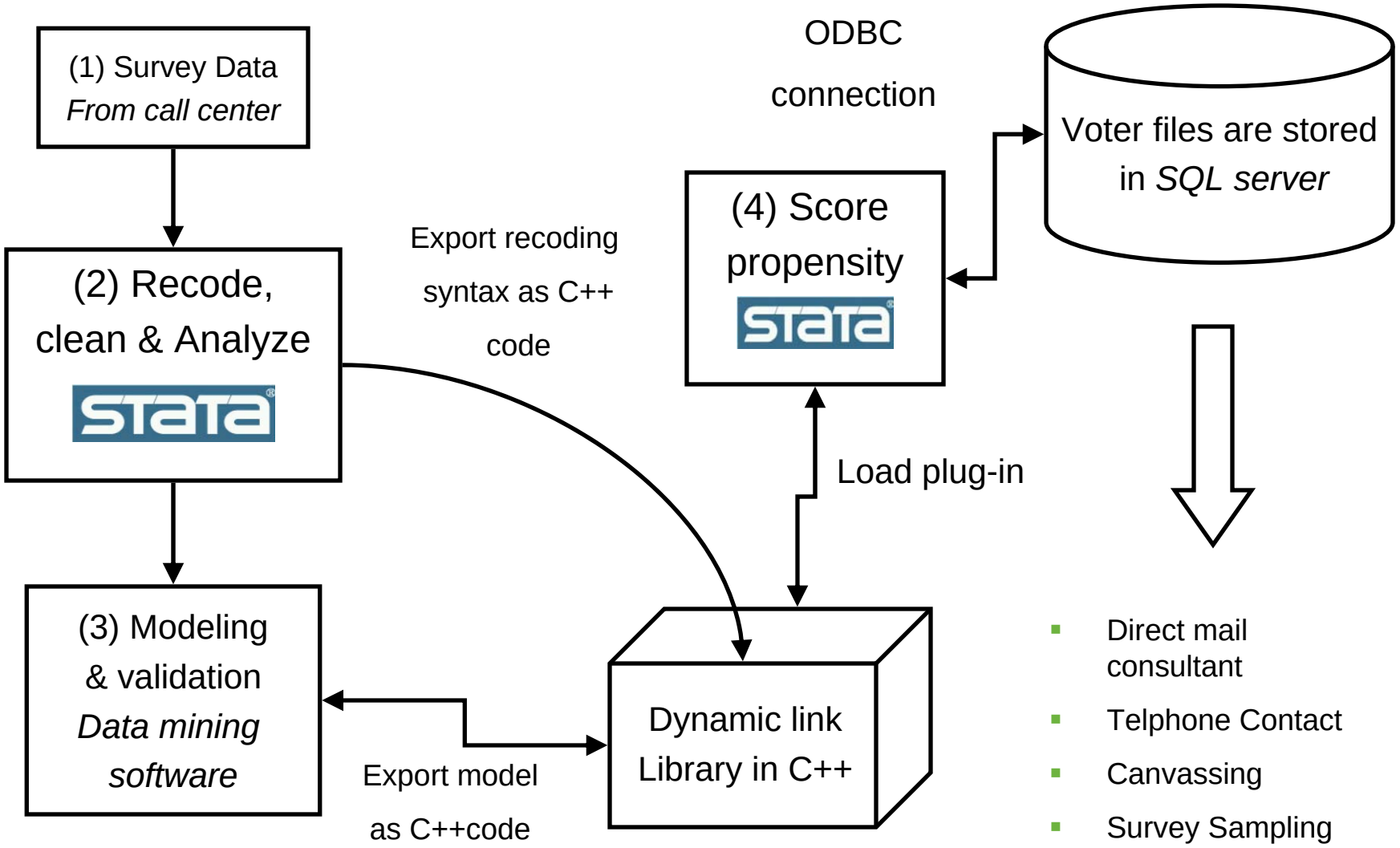
Statistical Models for Micro-Targeting

- Required properties for targeting models
  - Large number of covariates.
  - Incorporating complex interaction terms.
  - Robustness.
  - Need to avid over fitting.

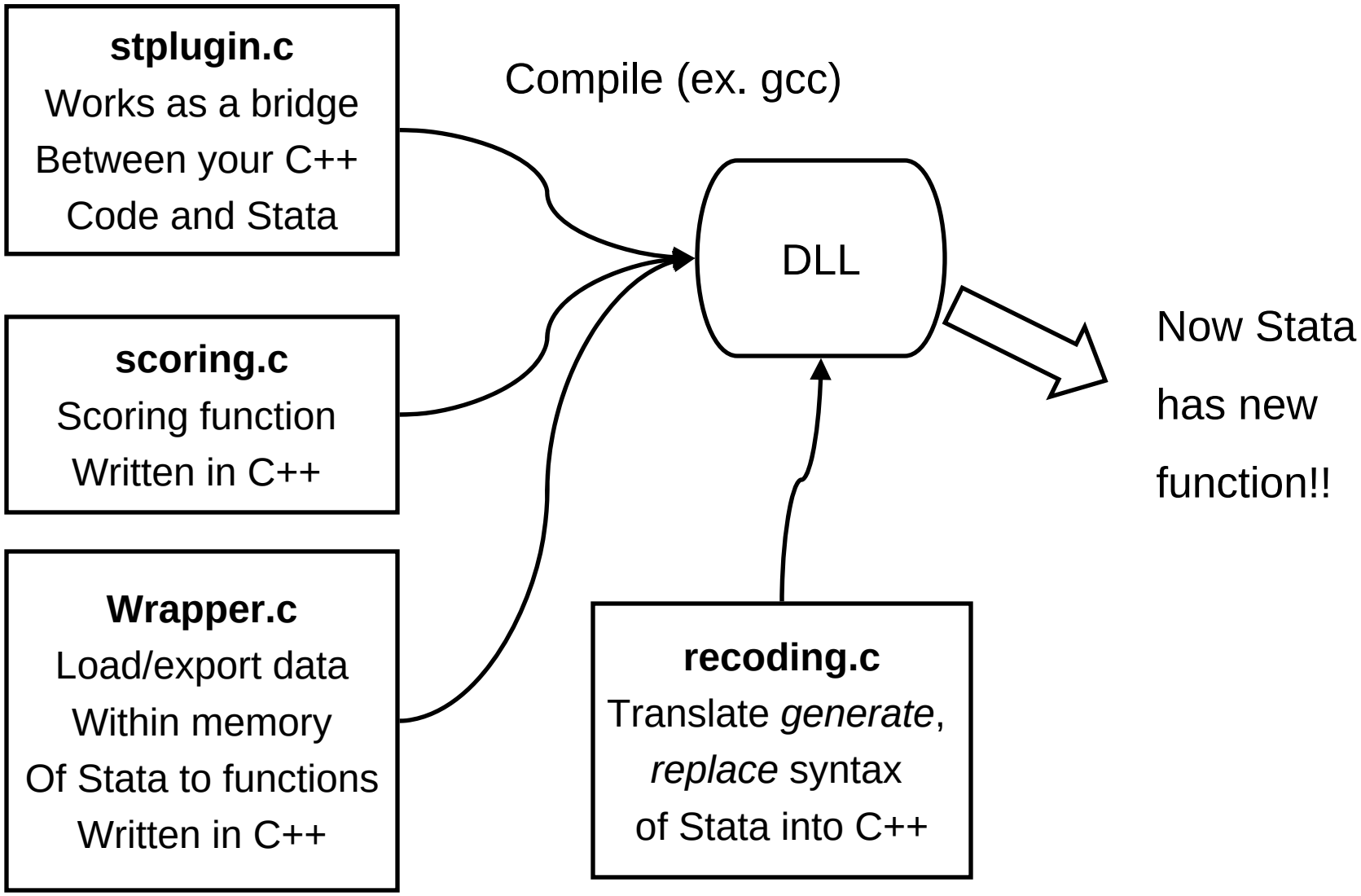
- Working solutions for above requirement (2009 version)
  - Decision tree models.
  - Model averaging (bagging or boosting).
  - Cross validation.
  - Use (political) common sense.



Workflow of Micro-Targeting Analysis & Scoring



How to Make Plug-ins



Example

- Example of scoring 4th random subset (2nd argument) of Michigan voter file (1st argument).

```
capture program drop Scoring
program define Scoring, rclass

   odbc load, dialog(complete) dsn(“xxx")  clear
   sqlshow exec("SELECT VAR1, VAR2 … FROM Table_`1’
   WHERE Rand = `2’)

   do recoding.do
   gen Predicted = .
   plugin call Scoring VAR1 VAR2 Predicted
   save w:\Scores\Score_`1’_`2’, replace
end

clear
Scoring MI 4
```

Is Plug-ins Worth Investing?

- It is not feasible to write scoring syntax for boosted/bagged model using SQL command or Stata do file.
- Time needed for coding in C++ becomes insignificant for large applications.

	Expected time to score AR voter file	Expected time to score CA voter file
Singe Decision Tree (100 nodes)	241 sec	39 min
Boosted Tree 30 trees (100 nodes)	316 sec	51 min
Boosted Tree 300 trees (100 nodes)	914 sec	147 min
Bagged Tree 300 trees (248 nodes)	62 min	596 min

- 21 predictors
- N= 67,177

Putting It in Perspective

- Loading Data via ODBC: 49% of total process
  - Under windows 2003 server platform, ODBC seems fast enough.
  - DB should be indexed by the query variable.
  - SQL server can act faster if more memory/CPUs are dedicated.
- Scoring Data (Bagging): 48% of total process
  - This of course depends on complexity of the model.
- Recoding: 2% of total process
  - Recoding data that are already in Stata's memory is fast.
- Saving/exporting data: 1% of total time
  - Saving Stata file in fast disk array (RAID 10) requires negligible time.

Summary

- Stata's flexibility and various functions make it ideal platform for data analysis and data management.
- However, relational database can handle large data with much ease.
- Specialized software (ex. data mining) can run certain analysis more efficiently (decision tree & boosting), and many of them can spit C++ codes.
- Stata can speak both ODBC and C++.