The Architecture of a Churn Prediction System based on Stream Mining

Borja Balle (UPC), Bernardino Casas (UPC), Alex Catarineu (UPC),
Ricard Gavaldà (UPC), David Manzano (Ericsson Spain)

CCIA 2013, Vic, oct. 24th

Outline

1. Churn Prediction and Stream Mining
2. Goals of proof-of-concept
3. The system
4. Conclusions

On Data Stream Mining

- Data arrive as a sequence of items, at high speed
- Can't store all of it, not even in secondary memory
- An item is processed and discarded
- Needs to provide accurate answers at all times
- Data distribution evolves over time
- Mined patterns must be created, revised, possibly dropped

On Churning

Churning

Churn is the action of customers who discontinue a service or who leave a company

- Focus on Telcos
- *Getting a new customer is more expensive than keeping it*
  - Predict subscribers (customers) likely to leave
  - With enough time to take retention actions

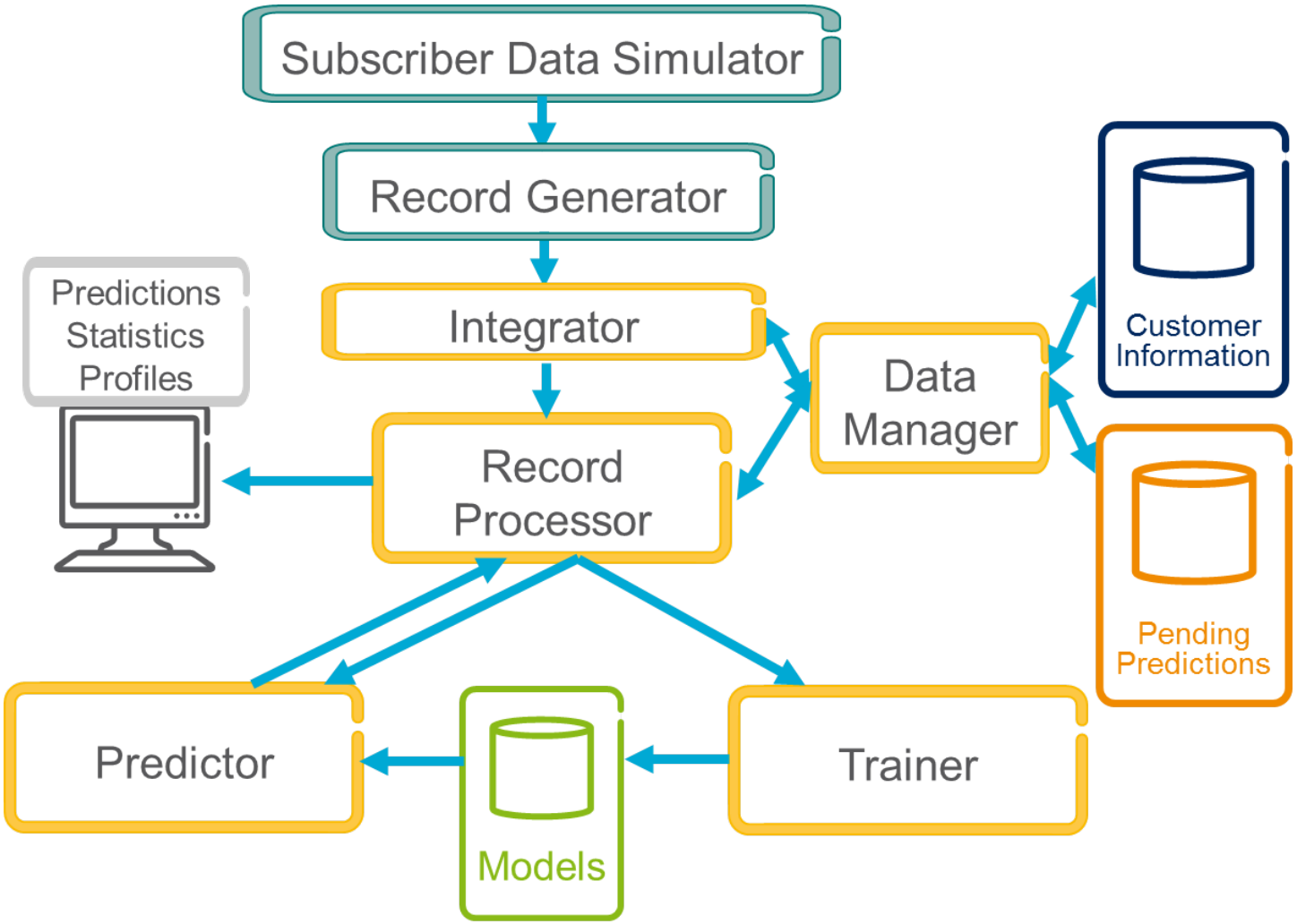
Data Mining and Churn Prediction

- Approach via data mining:
  - Use data from the past to predict future behaviors
  - I.e., find patterns and subscriber profiles that precede churning
- Why stream mining?
  - Subscriber activity = high-speed stream
  - Highly volatile customer patterns: our and competitor moves, ad campaigns, external events, market changes, social trends, ...
  - Need to detect emerging patterns ASAP
  - Need to keep prediction rules updated at all times

Goals

- Show potential of stream mining techniques in churn prediction scenarios:
  - Able to keep prediction rules updated at all times
  - Business value of fast reaction to changes
  - Show current churn patterns and profiles for detailed analysis
  - Ability to suggest optimal retention actions
- Not intended: final, tuned, scalable product
- Learned lessons towards a real product, particularly scalability
- Production system should be developed with churn experts and on real data

Model of the system



Details of the modules I

- Generator Generates stream of Events
  - Currently: "joins", "calls", "gets-bill", "complains", "churns"

  Markov chain with states "happy", "neutral", "angry", "churn", and user-specific values for "communicative" and "impulsive"

- Integrator: Reads the different incoming streams and creates a single stream of Events

Details of the modules II

- Record generator: Generates records:

  `(customer, attr1, attr2, ..., attrN, churn?)`

  with features useful for prediction, computed from past events and static info

- Proactively injects records for inactive subscribers

Details of the modules III

- Prediction Manager:
  - Based on the MOA (Massive Online Analysis) stream mining software
  - "Adaptive Hoeffding Tree" prediction method
  - Processes records from active customers (Events) and injects occasional records for inactive customers
  - Keeps queue of predictions pending to be verified
  - Current set of features, consistent with literature:
    - Age & sex
    - Income range
    - Average call duration
    - # calls last week
    - # calls last month
    - $\triangle$ # calls in last 2 weeks
    - $\triangle$ # calls in last 2 months
    - % in-company calls
    - # resolved complaints
    - # unresolved complaints
    - Average bill value
    - $\triangle$ between last 2 bills

Databases

- General info: Current rates, zip codes, special numbers (complaints, customer service, ...)
- Subscriber information, both static and dynamic
- Pending predictions. Large and *the* bottleneck
  - In RAM in latest stable version
  - Candidate for (key,value) NoSQL database
  - Preliminary experience with Cassandra
- About 10,000 records/second, 4Mb/1,000 subscribers

Threads

We implement this system with threads.

- GUI App (client)
  - Thread 1: Graphic user interface
- Server App
  - Thread 2: Synthetic data Generator
  - Thread 3: Integrator, Record generator
  - Thread 4: Proactive
  - Thread 5: Prediction Manager (MOA)

Conclusions

- Prototype to illustrate potential advantages of stream mining
- Indicates (but only partially implements) scalability
- Some of the many lines for future work:
  - increase throughput
  - better subscriber profiling
  - deal with geographical distribution (centralization?)
  - use friends net (contact graph, peer-pressure)
  - use twitter, facebook information
- But future work requires access to domain knowledge and real data

Thank you!

The Architecture of a Churn Prediction System based on Stream Mining

Borja Balle, Bernardino Casas, Alex Catarineu,
Ricard Gavaldà, David Manzano

Contact: gavalda@lsi.upc.edu

I'm around for demo till tomorrow lunchtime