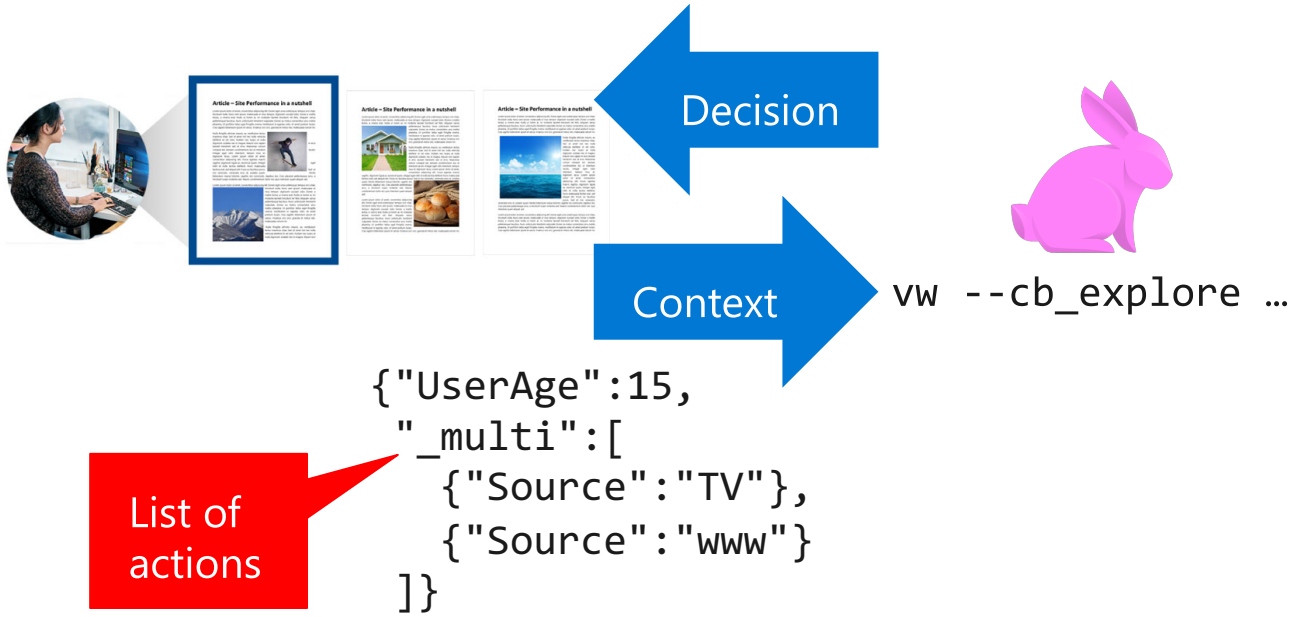
VW Workshop
Slates

Adith Swaminathan & Jack Gerrits
Microsoft Research

cb_explore: Contextual Decisions with Exploration

- Use contextual bandit algorithms to pick actions that maximize rewards
  - Balances exploration to discover rewarding actions against exploitation
  - Tunable parameters: exploration rate, learning rate, base learner type, etc.

- Action set represented as a list
  - Either a fixed set across contexts, or,
  - Featurized action lists per context



Slates: Motivating Applications

Network configuration

Page layout optimization

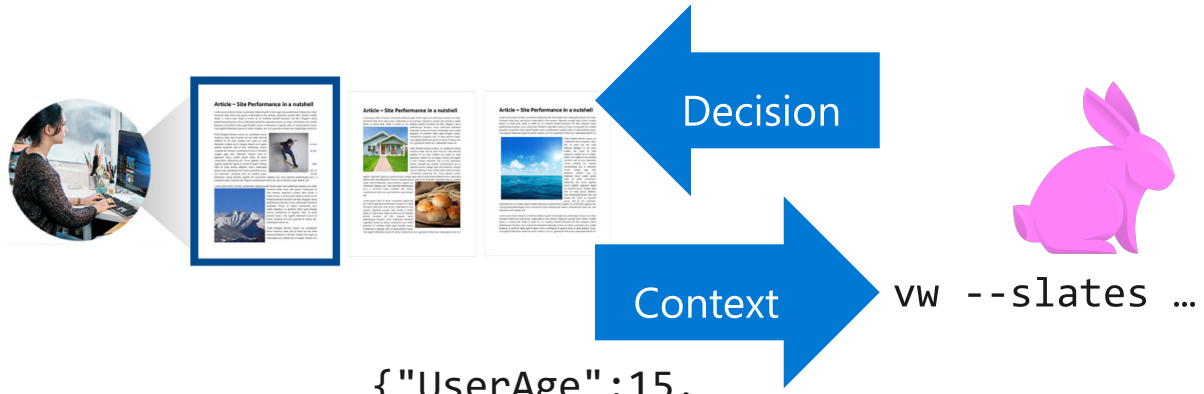
Slate recommendations

- "Composite" actions that are composed of several choices
  - Each choice populates a "slot"
  - Reward is observed for the composite action
  - Combinatorially many actions if encoded as a list

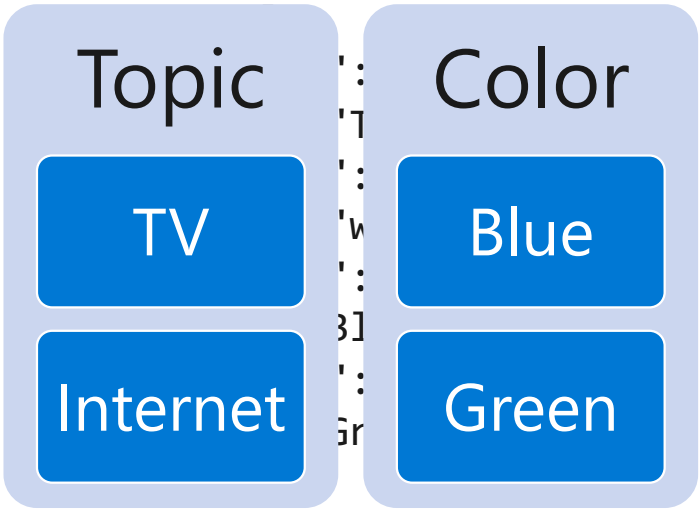
  - VW automagically handles feature interactions across namespaces…
  - Can it handle these action-slot-interactions also? **Yes! --slates**

Slates: Problem Definition

- Context
  - Shared features + Number of slots
  - Slot-specific features
  - Action-set for each slot
    - **Disjoint** action-sets across slots
- Decision
  - List (len=#slots) of action ids and selection probabilities
    - Semantics: probability of slate = product of selection probabilities
  - Action ids index into respective action set
- Reward
  - A single scalar representing global slate-level reward




```
{"UserAge":15,
```




```
"_slots":[
   {"MobileFriendly":1},
   {"ColorblindFriendly":1}]}
```

Contract

VW will run cb explore-exploit with actions as the cartesian product of slot-specific action sets

Under The Hood

VW --slates repeatedly calls cb_explore with book-keeping

- One cb_explore instance for each slot
- Context automatically computes correct shared/slot/action feature interactions
- Action constructed by invoking each cb_explore instance to populate slots independently
- Reward at slate-level is automatically decomposed into slot-specific rewards for each learner

Tunable hyper-parameters: Same as for base learner (cb_explore).

- No additional slate-specific tuning parameters

slates advantages over cb_explore:

1. Efficient encoding
2. Faster learning
3. Correct feature interactions

When to use slates vs. cb_explore vs. ccb?

cb_explore: General purpose base-learner. Works for small #actions.

ccb: Extends cb_explore to slate recommendation scenarios.

- Expects slot-specific rewards. (No slate->slot credit assignment)
- Works with dependent slot-specific action sets. (Supports rankings)

slates: Extends cb_explore to slate recommendation scenarios.

- Works with only slate-level reward. (Performs slate->slot credit assignment)
- Currently supports only disjoint slot-specific action sets. (E.g., no rankings support yet)

https://arxiv.org/abs/1605.04812

A Concrete Example & Demo

Outfit Optimization for Job Interview

https://www.youtube.com/watch?v=G-omu_ki7YM