Interactive machine learning

COMS 4721 Spring 2022

Introduction

Supervised (non-interactive) machine learning

Supervised (non-interactive) machine learning

Interactive vs. non-interactive machine learning

Non-interactive machine learning (e.g., “supervised learning”):

Interactive vs. non-interactive machine learning

Non-interactive machine learning (e.g., “supervised learning”):

1. Raw (unlabeled) data is collected.

Interactive vs. non-interactive machine learning

Non-interactive machine learning (e.g., “supervised learning”):

1. Raw (unlabeled) data is collected.
2. Human looks at data, labels each one (e.g., “spam” vs. “ham”), then goes away.

Interactive vs. non-interactive machine learning

Non-interactive machine learning (e.g., "supervised learning"):

1. Raw (unlabeled) data is collected.
2. Human looks at data, labels each one (e.g., "spam" vs. "ham"), then goes away.
3. Machine learns a classifier from the labeled data.

Interactive vs. non-interactive machine learning

Non-interactive machine learning (e.g., "supervised learning"):

1. Raw (unlabeled) data is collected.
2. Human looks at data, labels each one (e.g., "spam" vs. "ham"), then goes away.
3. Machine learns a classifier from the labeled data.

Interactive machine learning:

Machine learns from many rounds of interaction with humans.

Why interaction?

Why interaction?

1. Learn faster.

Why interaction?

1. Learn faster.

2. Data naturally come from interactions in the real world.

Why interaction?

1. Learn faster.
2. Data naturally come from interactions in the real world.
3. ...

Why interaction?

1. Learn faster.
   Active learning
2. Data naturally come from interactions in the real world.
   Online learning
3. ...

Active learning

Non-interactive machine learning:

1. Raw data is collected.
2. Human looks at data, labels each one (e.g., "spam" vs. "ham"), then goes away.
3. Machine learns a classifier from the labeled data.

Active learning

Non-interactive machine learning:

1. Raw data is collected.
2. Human looks at data, labels each one (e.g., "spam" vs. "ham"), then goes away.
3. Machine learns a classifier from the labeled data.

Active learning

Non-interactive machine learning:

1. Raw data is collected.
2. Human looks at data, labels each one (e.g., "spam" vs. "ham"), then goes away.
3. Machine learns a classifier from the labeled data.

Raw data is often cheap,
but labeling is expensive!

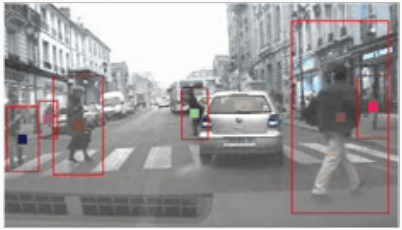
Active learning

Non-interactive machine learning:

1. Raw data is collected.
2. Human looks at data, labels each one (e.g., "spam" vs. "ham"), then goes away.
3. Machine learns a classifier from the labeled data.

Raw data is often cheap,
but labeling is expensive!

Pedestrian detection



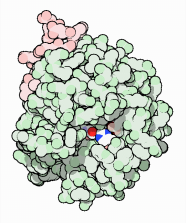
(Abramson & Freund, 2003)

Active learning

Non-interactive machine learning:

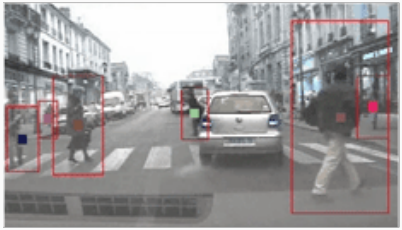
1. Raw data is collected.
2. Human looks at data, labels each one (e.g., "spam" vs. "ham"), then goes away.
3. Machine learns a classifier from the labeled data.

Drug design



(Warmuth *et al*, 2003)

Pedestrian detection



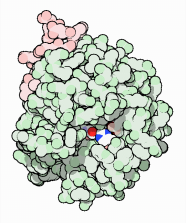
(Abramson & Freund, 2003)

Active learning

Non-interactive machine learning:

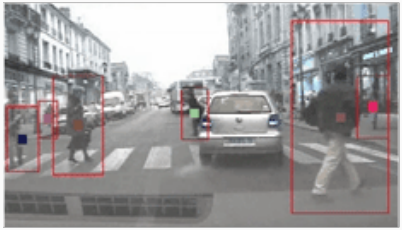
1. Raw data is collected.
2. Human looks at data, labels each one (e.g., "spam" vs. "ham"), then goes away.
3. Machine learns a classifier from the labeled data.

Drug design



(Warmuth *et al*, 2003)

Pedestrian detection



(Abramson & Freund, 2003)

Can we *learn* to be selective about which data to label?

Online learning

For each patient:

Online learning

For each patient:

1. Observe patient's symptoms, medical history, genetic sequence, test results, ...

Online learning

For each patient:

1. Observe patient's symptoms, medical history, genetic sequence, test results, ...
2. Prescribe treatment.

Online learning

For each patient:

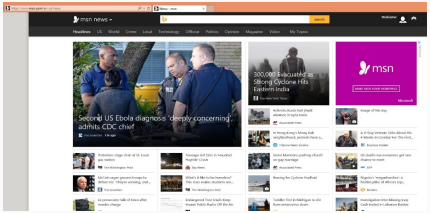
1. Observe patient's symptoms, medical history, genetic sequence, test results, ...
2. Prescribe treatment.
3. Observe impact on patient's health (e.g., improves, worsens).

Online learning

For each patient:

1. Observe patient's symptoms, medical history, genetic sequence, test results, ...
2. Prescribe treatment.
3. Observe impact on patient's health (e.g., improves, worsens).

News article recommendation



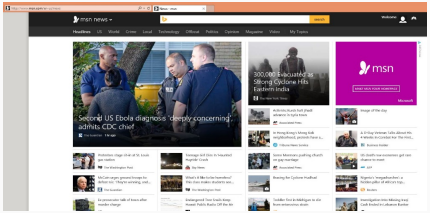
(Li *et al*, 2010)

Online learning

For each patient:

1. Observe patient's symptoms, medical history, genetic sequence, test results, ...
2. Prescribe treatment.
3. Observe impact on patient's health (e.g., improves, worsens).

News article recommendation



(Li *et al*, 2010)

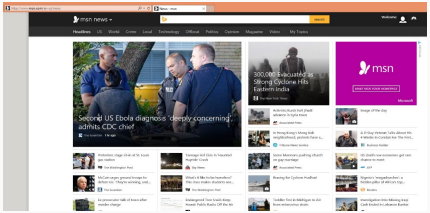
Choices determine what data are observed.

Online learning

For each patient:

1. Observe patient's symptoms, medical history, genetic sequence, test results, ...
2. Prescribe treatment.
3. Observe impact on patient's health (e.g., improves, worsens).

News article recommendation



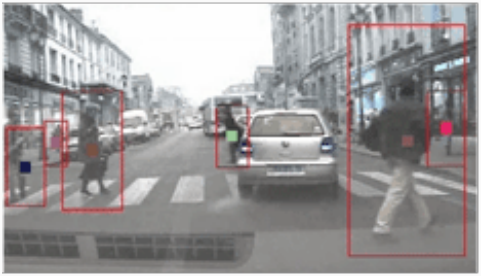
(Li *et al*, 2010)

Choices determine what data are observed.

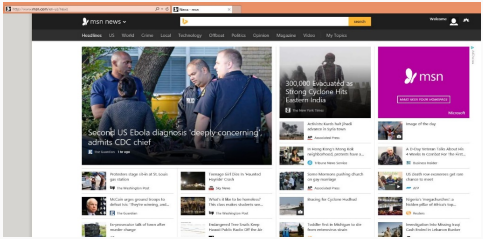
What would have been observed *if a different treatment was prescribed*?

Outline

1. Active learning



2. Online learning



Active learning

Toy example

Learning a threshold function

- Represent unlabeled e-mail by

$$x := \text{fraction of e-mail characters that is '\$'}.$$

Toy example

Learning a threshold function

- Represent unlabeled e-mail by

$$x := \text{fraction of e-mail characters that is '\$'}.$$

- **Simplifying assumption**: there is a threshold $t \in (0, 1)$ s.t.

an e-mail is spam if and only if $x > t$.

Toy example

Learning a threshold function

- Represent unlabeled e-mail by

$$x := \text{fraction of e-mail characters that is '\$'}.$$

- **Simplifying assumption**: there is a threshold $t \in (0, 1)$ s.t.
  an e-mail is spam if and only if $x > t$.

Given n unlabeled e-mails, **adaptively** choose which to label.

Toy example

Learning a threshold function

- Represent unlabeled e-mail by

$$x := \text{fraction of e-mail characters that is '\$'}.$$

- **Simplifying assumption**: there is a threshold $t \in (0, 1)$ s.t.
  an e-mail is spam if and only if $x > t$.

Given n unlabeled e-mails, **adaptively** choose which to label.

Need to label only $\log_2(n)$ adaptively chosen e-mails to find correct threshold.

Generalized binary search?

Does this generalize beyond simple threshold functions?

Generalized binary search?

Does this generalize beyond simple threshold functions?

- In some cases, yes!

  E.g., **linear classifiers**.
  - **Non-active ("supervised")**: label all n data.
  - **Active**: label just $\propto \log n$ adaptively chosen data (Freund, Seung, Shamir, & Tishby, 1997).

Generalized binary search?

Does this generalize beyond simple threshold functions?

- In some cases, yes!

  E.g., **linear classifiers**.
  - **Non-active ("supervised")**: label all n data.
  - **Active**: label just $\propto \log n$ adaptively chosen data (Freund, Seung, Shamir, & Tishby, 1997).

- But not in all cases.
  - **Sampling bias** can completely derail the learner.

Typical strategy: Uncertainty sampling

Start with IID sample of unlabeled examples U

Typical strategy: Uncertainty sampling

Start with IID sample of unlabeled examples U

1. Label some examples from U; fit model for $Y \mid X$ (e.g., logistic regression)

Typical strategy: Uncertainty sampling

Start with IID sample of unlabeled examples U

1. Label some examples from U; fit model for $Y \mid X$ (e.g., logistic regression)
2. Repeat:
   - Select unlabeled examples $x \in U$ for which current fitted model is most uncertain about $Y \mid X = x$
   - Label those examples, and re-fit model

Sampling bias

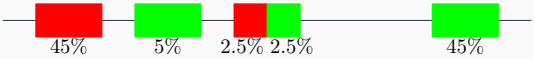
Labeled data from active learner is typically **biased**,
i.e., **not representative of overall data**.

Sampling bias

Labeled data from active learner is typically **biased**,
i.e., **not representative of overall data**.

Example: threshold functions, $x \in [0, 1]$, $y \in \{\text{ham}, \text{spam}\}$.

True (labeled) data distribution:

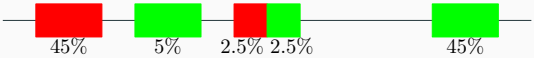


Sampling bias

Labeled data from active learner is typically **biased**,
i.e., **not representative of overall data**.

Example: threshold functions, $x \in [0, 1]$, $y \in \{\text{ham}, \text{spam}\}$.

True (labeled) data distribution:



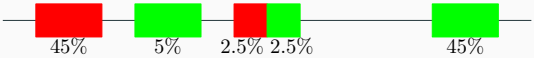
Labeled sample (based on "uncertainty sampling"):

Sampling bias

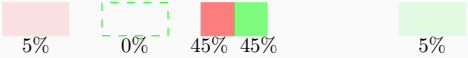
Labeled data from active learner is typically **biased**,
i.e., **not representative of overall data**.

Example: threshold functions, $x \in [0, 1]$, $y \in \{\text{ham}, \text{spam}\}$.

True (labeled) data distribution:



Labeled sample (based on "uncertainty sampling"):

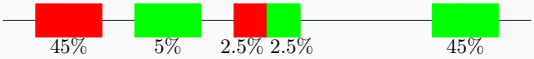


Sampling bias

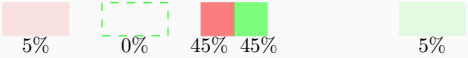
Labeled data from active learner is typically **biased**,
i.e., **not representative of overall data**.

Example: threshold functions, $x \in [0, 1]$, $y \in \{\text{ham}, \text{spam}\}$.

True (labeled) data distribution:



Labeled sample (based on "uncertainty sampling"):



Many active learners converge to threshold with 5% error rate, but best threshold has 2.5% error rate.

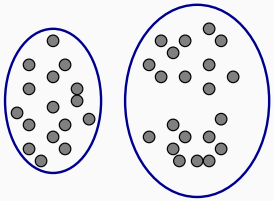
Tricks to manage sampling bias #1: sampling within clusters

Trick #1. **Optimistically cluster data using your favorite algorithm**, and hope that each cluster is “pure” in label.

Tricks to manage sampling bias #1: sampling within clusters

Trick #1. **Optimistically cluster data using your favorite algorithm**, and hope that each cluster is "pure" in label.

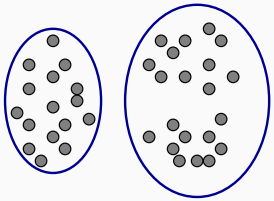
Then: label randomly chosen points from each cluster.



Tricks to manage sampling bias #1: sampling within clusters

Trick #1. **Optimistically cluster data using your favorite algorithm**, and hope that each cluster is "pure" in label.

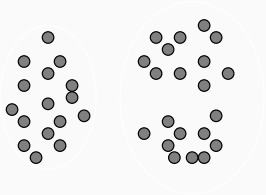
Then: label randomly chosen points from each cluster.



If clusters turn out to be "pure", then done!

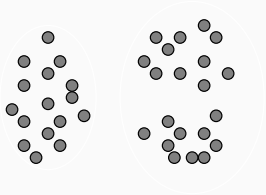
Adapting to cluster structure

1. Unlabeled data.

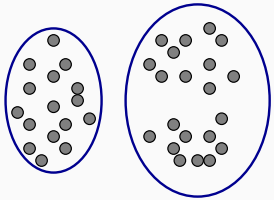


Adapting to cluster structure

1. Unlabeled data.

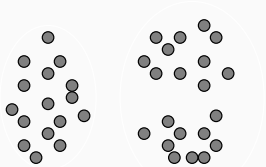


2. Cluster the data.

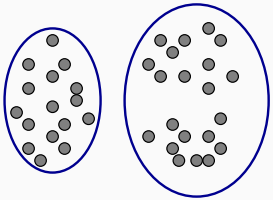


Adapting to cluster structure

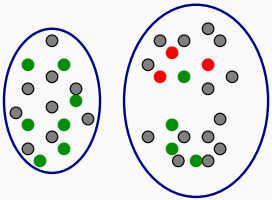
1. Unlabeled data.



2. Cluster the data.



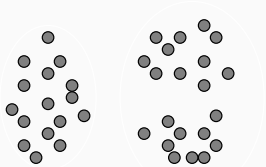
3. Query some labels.



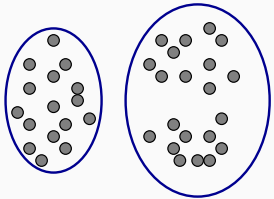
Now what?

Adapting to cluster structure

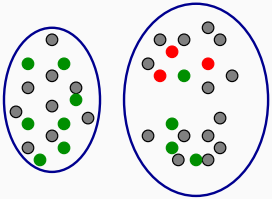
1. Unlabeled data.



2. Cluster the data.

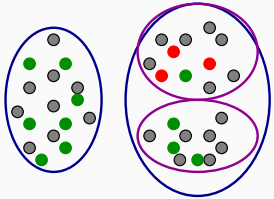


3. Query some labels.



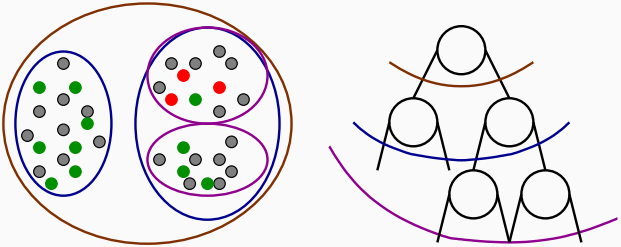
Now what?

4. Refine the clustering.

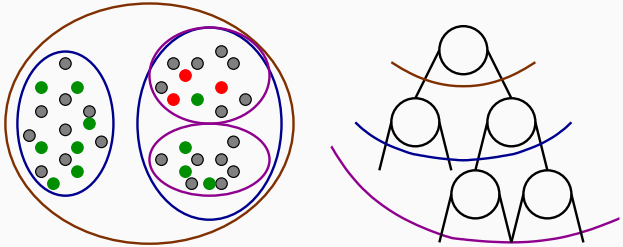


Recurse on the refinement.

Using a hierarchical clustering

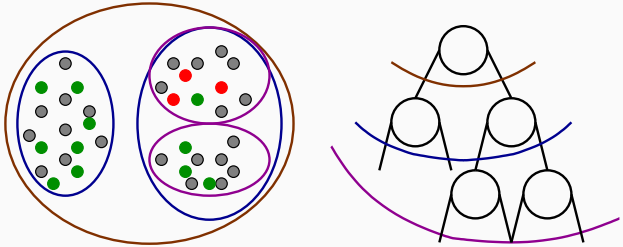


Using a hierarchical clustering



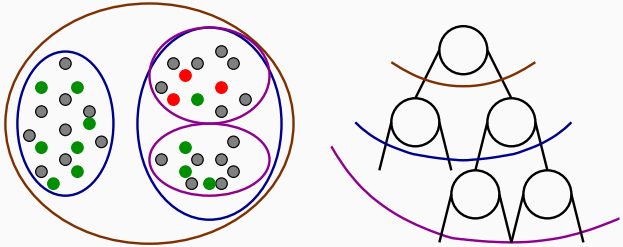
- Work with pruning of tree; induces flat clustering of data.

Using a hierarchical clustering



- Work with pruning of tree; induces flat clustering of data.
- **Repeat**: pick cluster; get label of a random point in it.

Using a hierarchical clustering



- Work with pruning of tree; induces flat clustering of data.
- **Repeat**: pick cluster; get label of a random point in it.
- For each tree node v, maintain:
  1. Majority label $L(v)$.
  2. Empirical label frequency $\widehat{p}_{v,\ell}$ for all possible labels ℓ.
  3. Confidence intervals for true label frequencies $p_{v,\ell}$ for all possible labels ℓ.

Tricks to manage sampling bias #2: bias correction

Trick #2. **Correct the bias using importance weights.**

Sampling bias: data S w/ labels is not iid sample from P, even if original unlabeled data was.

Tricks to manage sampling bias #2: bias correction

Trick #2. **Correct the bias using importance weights.**

Sampling bias: data S w/ labels is not iid sample from P, even if original unlabeled data was.

Difference between empirical error rate and true error rate can be very large, even in expectation.

$$\left| \mathbb{E}\left[\frac{1}{|S|} \sum_{(x,y)\in S} 1\{f(x) \neq y\} \right] - \Pr_{(x,y)\sim P}(f(x) \neq y) \right| \gg 0.$$

Tricks to manage sampling bias #2: bias correction

Trick #2. **Correct the bias using importance weights.**

Sampling bias: data S w/ labels is not iid sample from P, even if original unlabeled data was.

Difference between empirical error rate and true error rate can be very large, even in expectation.

$$\left| \mathbb{E}\left[\frac{1}{|S|} \sum_{(x,y)\in S} 1\{f(x) \neq y\} \right] - \Pr_{(x,y)\sim P}(f(x) \neq y) \right| \gg 0.$$

Solution: correct bias using randomization.

Randomized selective sampling template

Randomized selective sampling template

1. Initialize $W := \emptyset$.

Randomized selective sampling template

1. Initialize $W := \emptyset$.
2. For $i = 1, 2, \ldots, n$:
   - Get new point x_i.
   - Choose probability $p_i \in (0, 1)$.
   - With probability p_i: get label y_i, add $(x_i, y_i, 1/p_i)$ to W.

Randomized selective sampling template

1. Initialize $W := \emptyset$.
2. For $i = 1, 2, \ldots, n$:
   - Get new point x_i.
   - Choose probability $p_i \in (0, 1)$.
   - With probability p_i: get label y_i, add $(x_i, y_i, 1/p_i)$ to W.
3. Choose classifier $f \in \mathcal{F}$ to minimize weighted error rate:

$$\mathrm{err}_W(f) := \frac{1}{n} \sum_{(x,y,1/p) \in W} \frac{1}{p} \cdot 1\{f(x) \neq y\}.$$

Randomized selective sampling template

1. Initialize $W := \emptyset$.
2. For $i = 1, 2, \ldots, n$:
   - Get new point x_i.
   - Choose probability $p_i \in (0, 1)$.
   - With probability p_i: get label y_i, add $(x_i, y_i, 1/p_i)$ to W.
3. Choose classifier $f \in \mathcal{F}$ to minimize weighted error rate:

$$\mathrm{err}_W(f) := \frac{1}{n} \sum_{(x,y,1/p) \in W} \frac{1}{p} \cdot 1\{f(x) \neq y\}.$$

Let $Q_i := 1\{\text{got label } y_i\}$, so $\mathbb{E}[Q_i \mid p_i] = p_i$, and

$$\mathbb{E}[\mathrm{err}_W(f)] = \mathbb{E}\left[\frac{1}{n} \sum_{i=1}^{n} \frac{Q_i}{p_i} \cdot 1\{f(x_i) \neq y_i\}\right] = \frac{1}{n} \sum_{i=1}^{n} 1\{f(x_i) \neq y_i\}.$$

Active learning summary

1. Use interaction + adaptivity to be selective about which data to label.

Active learning summary

1. Use interaction + adaptivity to be selective about which data to label.
2. Must manage sampling bias.

Active learning summary

1. Use interaction + adaptivity to be selective about which data to label.
2. Must manage sampling bias.

   Two simple and general techniques:

   - Cluster-guided sampling.
   - Bias-correction.

Online learning

Contextual bandits (or, “learning with incomplete feedback”)

Online learning protocol for “contextual bandit” problem:

- For round $t = 1, 2, \ldots, T$:

Contextual bandits (or, “learning with incomplete feedback”)

Online learning protocol for “contextual bandit” problem:

- For round $t = 1, 2, \ldots, T$:
  1. Observe context x_t. (e.g., patient attributes)

Contextual bandits (or, “learning with incomplete feedback”)

Online learning protocol for “contextual bandit” problem:

- For round $t = 1, 2, \ldots, T$:
  1. Observe context x_t. (e.g., patient attributes)
  2. Choose action $a_t \in \{1, 2, \ldots, A\}$. (e.g., treatment option)

Contextual bandits (or, “learning with incomplete feedback”)

Online learning protocol for “contextual bandit” problem:

- For round $t = 1, 2, \ldots, T$:
  1. Observe context x_t. (e.g., patient attributes)
  2. Choose action $a_t \in \{1, 2, \ldots, A\}$. (e.g., treatment option)
  3. Observe reward $r_t(a_t) \in [0, 1]$. (e.g., 1 for full recovery, ...)

Contextual bandits (or, “learning with incomplete feedback”)

Online learning protocol for “contextual bandit” problem:

- For round $t = 1, 2, \ldots, T$:
  1. Observe context x_t. (e.g., patient attributes)
  2. Choose action $a_t \in \{1, 2, \ldots, A\}$. (e.g., treatment option)
  3. Observe reward $r_t(a_t) \in [0, 1]$. (e.g., 1 for full recovery, ...)

Goal: choose actions a_t to maximize cumulative reward.

Contextual bandits (or, “learning with incomplete feedback”)

Online learning protocol for “contextual bandit” problem:

- For round $t = 1, 2, \ldots, T$:
  1. Observe context x_t. (e.g., patient attributes)
  2. Choose action $a_t \in \{1, 2, \ldots, A\}$. (e.g., treatment option)
  3. Observe reward $r_t(a_t) \in [0, 1]$. (e.g., 1 for full recovery, ...)

Goal: choose actions a_t to maximize cumulative reward.

Context: x_t may be predictive of which action is best.

Contextual bandits (or, “learning with incomplete feedback”)

Online learning protocol for “contextual bandit” problem:

- For round $t = 1, 2, \ldots, T$:
  1. Observe context x_t. (e.g., patient attributes)
  2. Choose action $a_t \in \{1, 2, \ldots, A\}$. (e.g., treatment option)
  3. Observe reward $r_t(a_t) \in [0, 1]$. (e.g., 1 for full recovery, ...)

Goal: choose actions a_t to maximize cumulative reward.

Context: x_t may be predictive of which action is best.

Incomplete feedback: reward $r_t(a)$ for $a \neq a_t$ is not observed.

$\therefore$ data we observe is directly affected by actions we take.

Challenges

Challenges

1. **Exploration vs. exploitation**.
    - “Exploit”: use what you’ve learned.
    - “Explore”: learn about actions that could be good.

Challenges

1. **Exploration vs. exploitation**.
   - "Exploit": use what you've learned.
   - "Explore": learn about actions that could be good.

   Usually leads to **selection bias**.

Challenges

1. **Exploration vs. exploitation**.
   - "Exploit": use what you've learned.
   - "Explore": learn about actions that could be good.

   Usually leads to **selection bias**.
2. **Must use context**.
   - Want to do as well as the best decision rule ("policy")

     $$f\text{: context } x \mapsto \text{action } a$$

     from some family of decision rules $\mathcal{F}$ ("policy class").

Challenges

1. **Exploration vs. exploitation**.
   - "Exploit": use what you've learned.
   - "Explore": learn about actions that could be good.

   Usually leads to **selection bias**.
2. **Must use context**.
   - Want to do as well as the best decision rule ("policy")

     $$f\text{: context } x \ \mapsto \ \text{action } a$$

     from some family of decision rules $\mathcal{F}$ ("policy class").
   - Often, $\mathcal{F}$ is extremely large and expressive
     (e.g., $\mathcal{F} =$ all decision trees on d binary variables).

Hypothetical “full-information” setting

If we observed rewards for all actions ...

Hypothetical “full-information” setting

If we observed rewards for all actions ...

- Like **supervised learning**, have *labeled data* after t rounds:

$$\{(x_i, \vec{\rho}_i)\}_{i=1}^{t}, \quad \text{where } \vec{\rho}_i = (\rho_i(1), \ldots, \rho_i(A)).$$

Hypothetical “full-information” setting

If we observed rewards for all actions ...

- Like **supervised learning**, have *labeled data* after t rounds:

$$\{(x_i, \vec{\rho}_i)\}_{i=1}^{t}, \quad \text{where } \vec{\rho}_i = (\rho_i(1), \ldots, \rho_i(A)).$$

context	$\longrightarrow$	features
actions	$\longrightarrow$	classes
rewards	$\longrightarrow$	$-$costs
policy	$\longrightarrow$	classifier

Hypothetical “full-information” setting

If we observed rewards for all actions ...

- Like **supervised learning**, have *labeled data* after t rounds:

$$\{(x_i, \vec{\rho}_i)\}_{i=1}^t, \quad \text{where } \vec{\rho}_i = (\rho_i(1), \ldots, \rho_i(A)).$$

context	$\longrightarrow$	features
actions	$\longrightarrow$	classes
rewards	$\longrightarrow$	$-$costs
policy	$\longrightarrow$	classifier

- Can often exploit structure of $\mathcal{F}$ to get efficient algorithms.

  I.e., “supervised learning algorithm”:

$$\{(x_i, \vec{\rho}_i)\}_{i=1}^t \mapsto \text{“}f \in \mathcal{F} \text{ w/ max reward on } \{(x_i, \vec{\rho}_i)\}_{i=1}^t\text{”}.$$

Hypothetical “full-information” setting

If we observed rewards for all actions ...

- Like **supervised learning**, have *labeled data* after t rounds:

$$\{(x_i, \vec{\rho}_i)\}_{i=1}^{t}, \quad \text{where } \vec{\rho}_i = (\rho_i(1), \ldots, \rho_i(A)).$$

context	$\longrightarrow$	features
actions	$\longrightarrow$	classes
rewards	$\longrightarrow$	$-$costs
policy	$\longrightarrow$	classifier

- Can often exploit structure of $\mathcal{F}$ to get efficient algorithms.

  I.e., “supervised learning algorithm”:

$$\{(x_i, \vec{\rho}_i)\}_{i=1}^{t} \mapsto \text{“}f \in \mathcal{F} \text{ w/ max reward on } \{(x_i, \vec{\rho}_i)\}_{i=1}^{t}\text{”}.$$

But cannot directly use this with only partial feedback.

Counterfactual inference

Problem: don't observe reward $r_t(a)$ for $a \neq a_t$.

Counterfactual inference

Problem: don't observe reward $r_t(a)$ for $a \neq a_t$.

Trick: estimate reward about all actions, on average, using *randomness*.

Counterfactual inference

Problem: don't observe reward $r_t(a)$ for $a \neq a_t$.

Trick: estimate reward about all actions, on average, using *randomness*.

- Pick prob. dist. p_t over $\{1, 2, \ldots, A\}$, and draw $a_t \sim p_t$.

Counterfactual inference

Problem: don't observe reward $r_t(a)$ for $a \neq a_t$.

Trick: estimate reward about all actions, on average, using *randomness*.

- Pick prob. dist. p_t over $\{1, 2, \ldots, A\}$, and draw $a_t \sim p_t$.
- Observe $r_t(a_t)$.

Counterfactual inference

Problem: don't observe reward $r_t(a)$ for $a \neq a_t$.

Trick: estimate reward about all actions, on average, using *randomness*.

- Pick prob. dist. p_t over $\{1, 2, \ldots, A\}$, and draw $a_t \sim p_t$.
- Observe $r_t(a_t)$.
- Reward estimator: for *all* actions a,

$$\hat{r}_t(a) := \begin{cases} \dfrac{r_t(a_t)}{p_t(a_t)} & \text{if } a = a_t, \\ 0 & \text{if } a \neq a_t. \end{cases}$$

Counterfactual inference

Problem: don't observe reward $r_t(a)$ for $a \neq a_t$.

Trick: estimate reward about all actions, on average, using *randomness*.

- Pick prob. dist. p_t over $\{1, 2, \ldots, A\}$, and draw $a_t \sim p_t$.
- Observe $r_t(a_t)$.
- Reward estimator: for *all* actions a,

$$\hat{r}_t(a) := \begin{cases} \dfrac{r_t(a_t)}{p_t(a_t)} & \text{if } a = a_t, \\ 0 & \text{if } a \neq a_t. \end{cases}$$

  Estimator is unbiased for *all* actions a:

$$\mathop{\mathbb{E}}_{a_t \sim p_t}\left[\hat{r}_t(a)\right] = r_t(a) = \begin{pmatrix}\text{reward of } a \\ \text{in round } t\end{pmatrix}.$$

Online learning summary

1. Use randomness and counterfactual inference, can estimate cumulative reward for all actions.

Online learning summary

1. Use randomness and counterfactual inference, can estimate cumulative reward for all actions.
    - Hence, **can estimate cumulative reward for all policies**.

Online learning summary

1. Use randomness and counterfactual inference, can estimate cumulative reward for all actions.
   - Hence, **can estimate cumulative reward for all policies**.
   - This enables use of supervised learning algorithms!

Online learning summary

1. Use randomness and counterfactual inference, can estimate cumulative reward for all actions.
   - Hence, **can estimate cumulative reward for all policies**.
   - This enables use of supervised learning algorithms!
2. How to choose p_t?

Online learning summary

1. Use randomness and counterfactual inference, can estimate cumulative reward for all actions.
   - Hence, **can estimate cumulative reward for all policies**.
   - This enables use of supervised learning algorithms!
2. How to choose p_t?

$$\mathrm{Var}(\hat{r}_t(a)) = \left(\frac{1}{p_t(a)} - 1\right) r_t(a)^2.$$

Online learning summary

1. Use randomness and counterfactual inference, can estimate cumulative reward for all actions.
   - Hence, **can estimate cumulative reward for all policies**.
   - This enables use of supervised learning algorithms!
2. How to choose p_t?

$$\mathrm{Var}(\hat{r}_t(a)) = \left(\frac{1}{p_t(a)} - 1\right) r_t(a)^2.$$

$\therefore$ need to balance exploration and exploitation.

Online learning summary

1. Use randomness and counterfactual inference, can estimate cumulative reward for all actions.
   - Hence, **can estimate cumulative reward for all policies**.
   - This enables use of supervised learning algorithms!
2. How to choose p_t?

$$\mathrm{Var}(\hat{r}_t(a)) = \left(\frac{1}{p_t(a)} - 1\right) r_t(a)^2.$$

$\therefore$ need to balance exploration and exploitation.

Many ways to do this!

Conclusion

Recap and final points

Recap and final points

- **Interaction** is powerful resource for machine learning.
  - Reduce need for human annotation via **active learning**.

Recap and final points

- **Interaction** is powerful resource for machine learning.
  - Reduce need for human annotation via **active learning**.
- **Interaction** is intrinsic to many data-driven applications.
  - Can leverage supervised learning technology in new ways to solve these complex problems like **contextual bandits**.

Recap and final points

- **Interaction** is powerful resource for machine learning.
  - Reduce need for human annotation via **active learning**.
- **Interaction** is intrinsic to many data-driven applications.
  - Can leverage supervised learning technology in new ways to solve these complex problems like **contextual bandits**.

Questions?