Decision Making Under Uncertainty: Lecture 3—Personalized SAA

Lecture 3

Ryan Cory-Wright

Spring 2026

Outline of Lecture 3

Sample Average Approximation and Beyond

Improvement Strategy 1: Predictive to Prescriptive Analytics

Improvement Strategy 2: Smart "Predict Then Optimize"

Let's Look at Some Code on Prescriptive SAA For Next Part of Lecture

Warm-up: Let's Make a Deal

Imagine you are on a game-show, and you have the choice of three doors. Behind one door is a car, behind the other two doors are goats.

While goats make great pets, you prefer a car.

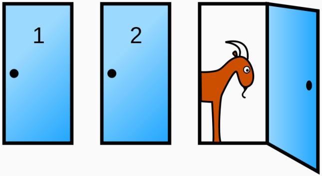
You pick a door, say door 1.

Warm-up: Let's Make a Deal

Imagine you are on a game-show, and you have the choice of three doors. Behind one door is a car, behind the other two doors are goats.

While goats make great pets, you prefer a car.

You pick a door, say door 1. The host, who knows what is behind each door and always opens a goat door, opens door 3, which has a goat.

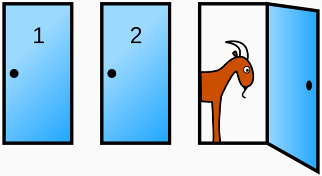


Warm-up: Let's Make a Deal

Imagine you are on a game-show, and you have the choice of three doors. Behind one door is a car, behind the other two doors are goats.

While goats make great pets, you prefer a car.

You pick a door, say door 1. The host, who knows what is behind each door and always opens a goat door, opens door 3, which has a goat.



She then asks you if you want to switch to door 2. **Should you switch?**

This Problem is About Conditional Expectations

- When you first picked a door, there was a 1/3 chance of winning a car if you picked door 1
- After door 3 was opened, the odds that a car was behind door 2 increased to 2/3. Why?

This Problem is About Conditional Expectations

- When you first picked a door, there was a 1/3 chance of winning a car if you picked door 1
- After door 3 was opened, the odds that a car was behind door 2 increased to 2/3. Why?
- 9 equally likely combinations of door: (car location) $\times$ (your initial choice).
- For 3/9 combinations, you win if you stay
- For 6/9 combinations, you win if you switch

This Problem is About Conditional Expectations

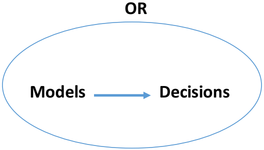
- When you first picked a door, there was a 1/3 chance of winning a car if you picked door 1
- After door 3 was opened, the odds that a car was behind door 2 increased to 2/3. Why?
- 9 equally likely combinations of door: (car location) $\times$ (your initial choice).
- For 3/9 combinations, you win if you stay
- For 6/9 combinations, you win if you switch
- Before opening door 3, we were indifferent between doors 1–2. After opening door 3, we prefer door 2.

This Problem is About Conditional Expectations

- When you first picked a door, there was a $1/3$ chance of winning a car if you picked door 1
- After door 3 was opened, the odds that a car was behind door 2 increased to $2/3$. Why?
- 9 equally likely combinations of door: (car location) $\times$ (your initial choice).
- For $3/9$ combinations, you win if you stay
- For $6/9$ combinations, you win if you switch
- Before opening door 3, we were indifferent between doors 1–2. After opening door 3, we prefer door 2. The *side information* we obtained by opening a door materially affected the best decision

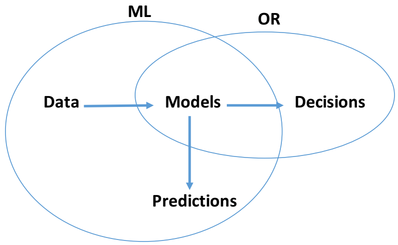
Sample Average Approximation and Beyond

Classical OR (Sample Average Approximation)



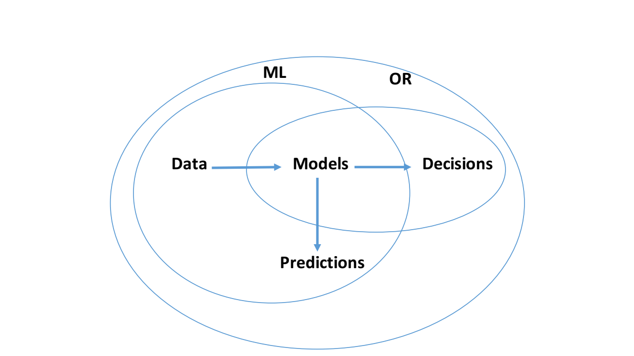
This is what you saw in your first optimization class

ML Today



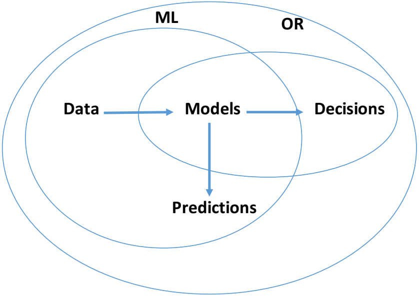
This is what you would see in an ML class

The future: Personalized Sample Average Approximation



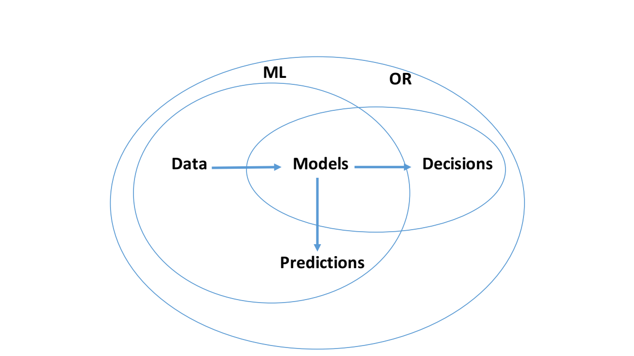
Optimization in the world *as it should be*, if not the world as it is.

The future: Personalized Sample Average Approximation



Optimization in the world *as it should be*, if not the world as it is. Because data is the objective reality we use to design models, models only exist *in our imagination*. And we should use data to improve decisions.

The future: Personalized Sample Average Approximation



Optimization in the world *as it should be*, if not the world as it is. Because data is the objective reality we use to design models, models only exist *in our imagination*. And we should use data to improve decisions. Let's concretize with an example.

Real Problem Setting: Big-Data Newsvendor

We run a hospital, and must decide how many nurses to schedule for tomorrow's shift. We have n observations of:

- The demand for the number of nurses in day $i \in [n]$, D_i
- The vector $\boldsymbol{z}_i \in \mathbb{R}^p$, which contains p different features (e.g., flu infection rates in the population, unemployment rate, current median rent, $\ldots$) predictive of demand D_i.

Real Problem Setting: Big-Data Newsvendor

We run a hospital, and must decide how many nurses to schedule for tomorrow's shift. We have n observations of:

- The demand for the number of nurses in day $i \in [n]$, D_i
- The vector $\boldsymbol{z}_i \in \mathbb{R}^p$, which contains p different features (e.g., flu infection rates in the population, unemployment rate, current median rent, $\ldots$) predictive of demand D_i.

Assume $(D_i, \boldsymbol{z}_i)$ are i.i.d. draws from the joint distribution of $(D, \boldsymbol{z})$, and we have access to $\boldsymbol{z}$, the vector of different features, for today's setting

Real Problem Setting: Big-Data Newsvendor

We run a hospital, and must decide how many nurses to schedule for tomorrow's shift. We have n observations of:

- The demand for the number of nurses in day $i \in [n]$, D_i
- The vector $\boldsymbol{z}_i \in \mathbb{R}^p$, which contains p different features (e.g., flu infection rates in the population, unemployment rate, current median rent, ...) predictive of demand D_i.

Assume $(D_i, \boldsymbol{z}_i)$ are i.i.d. draws from the joint distribution of $(D, \boldsymbol{z})$, and we have access to $\boldsymbol{z}$, the vector of different features, for today's setting

Discuss Among Yourselves: How should we set the number of nurses x, where each nurse needs to be paid c to work for the day, and we have revenue q per nurse actually utilized?

Real Problem Setting: Big-Data Newsvendor

We run a hospital, and must decide how many nurses to schedule for tomorrow's shift. We have n observations of:

- The demand for the number of nurses in day $i \in [n]$, D_i
- The vector $\boldsymbol{z}_i \in \mathbb{R}^p$, which contains p different features (e.g., flu infection rates in the population, unemployment rate, current median rent, ...) predictive of demand D_i.

Assume $(D_i, \boldsymbol{z}_i)$ are i.i.d. draws from the joint distribution of $(D, \boldsymbol{z})$, and we have access to $\boldsymbol{z}$, the vector of different features, for today's setting

Discuss Among Yourselves: How should we set the number of nurses x, where each nurse needs to be paid c to work for the day, and we have revenue q per nurse actually utilized?

Formally:

$$\max_{x \geq 0} \mathbb{E}_\omega[\min(D_\omega, x)q - cx | \boldsymbol{Z} = \boldsymbol{z}]$$

See Ban and Rudin (OR 2019) for a detailed study of problem setting

How do practitioners solve this problem?

Approach 1: Classical OR/SAA ("Adjust Your Expectations")

Ignore the side information $\boldsymbol{z}$, don't solve

$$\max_{x \geq 0} \mathbb{E}_\omega[\min(D_\omega, x)q - cx | \boldsymbol{Z} = \boldsymbol{z}]$$

Approach 1: Classical OR/SAA ("Adjust Your Expectations")

Ignore the side information $\boldsymbol{z}$, don't solve

$$\max_{x \geq 0} \mathbb{E}_\omega[\min(D_\omega, x)q - cx | \boldsymbol{Z} = \boldsymbol{z}]$$

Instead, solve

$$\max_{x \geq 0} \mathbb{E}_\omega[\min(D_\omega, x)q - cx]$$

Approach 1: Classical OR/SAA ("Adjust Your Expectations")

Ignore the side information $\boldsymbol{z}$, don't solve

$$\max_{x \geq 0} \mathbb{E}_\omega[\min(D_\omega, x)q - cx | \boldsymbol{Z} = \boldsymbol{z}]$$

Instead, solve

$$\max_{x \geq 0} \mathbb{E}_\omega[\min(D_\omega, x)q - cx]$$

via its sample-average approximation

$$\max_{x \geq 0} \frac{1}{n} \sum_{i=1}^{n} \min(D_i, x)q - cx$$

Like we talked about last week

Approach 1: Classical OR/SAA ("Adjust Your Expectations")

Ignore the side information $\boldsymbol{z}$, don't solve

$$\max_{x \geq 0} \mathbb{E}_\omega[\min(D_\omega, x)q - cx | \boldsymbol{Z} = \boldsymbol{z}]$$

Instead, solve

$$\max_{x \geq 0} \mathbb{E}_\omega[\min(D_\omega, x)q - cx]$$

via its sample-average approximation

$$\max_{x \geq 0} \frac{1}{n} \sum_{i=1}^{n} \min(D_i, x)q - cx$$

Like we talked about last week

- Pros: SAA converges almost surely to an optimal solution where we don't have any side information

Approach 1: Classical OR/SAA ("Adjust Your Expectations")

Ignore the side information $\boldsymbol{z}$, don't solve

$$\max_{x \geq 0} \mathbb{E}_\omega[\min(D_\omega, x)q - cx | \boldsymbol{Z} = \boldsymbol{z}]$$

Instead, solve

$$\max_{x \geq 0} \mathbb{E}_\omega[\min(D_\omega, x)q - cx]$$

via its sample-average approximation

$$\max_{x \geq 0} \frac{1}{n} \sum_{i=1}^{n} \min(D_i, x)q - cx$$

Like we talked about last week

- Pros: SAA converges almost surely to an optimal solution where we don't have any side information
- Cons: even when we have infinite data and know the marginal distribution of D, we leave something on the table by ignoring $\boldsymbol{z}$ (e.g., what if $\boldsymbol{z}$ perfectly predicts D?)

Approach 2: (Naive) Predict-then-optimize

Take a two-step approach:

1. **Predict:** Use historical observations $(\boldsymbol{z}_i, \boldsymbol{D}_i)_{i \in [n]}$ to create a model for how $\boldsymbol{D}$ depends on $\boldsymbol{z}$, say $\hat{D} = f(\boldsymbol{z})$, where f is our trained model and $\hat{D}$ our prediction

Approach 2: (Naive) Predict-then-optimize

Take a two-step approach:

1. **Predict:** Use historical observations $(\boldsymbol{z}_i, \boldsymbol{D}_i)_{i \in [n]}$ to create a model for how $\boldsymbol{D}$ depends on $\boldsymbol{z}$, say $\hat{D} = f(\boldsymbol{z})$, where f is our trained model and $\hat{D}$ our prediction
2. **Optimize:** Solve the optimization problem assuming $\hat{D} = f(\boldsymbol{z})$, output the solution $x = \hat{D}$

Approach 2: (Naive) Predict-then-optimize

Take a two-step approach:

1. **Predict:** Use historical observations $(\boldsymbol{z}_i, \boldsymbol{D}_i)_{i \in [n]}$ to create a model for how $\boldsymbol{D}$ depends on $\boldsymbol{z}$, say $\hat{D} = f(\boldsymbol{z})$, where f is our trained model and $\hat{D}$ our prediction
2. **Optimize:** Solve the optimization problem assuming $\hat{D} = f(\boldsymbol{z})$, output the solution $x = \hat{D}$
3. But this is *obviously* suboptimal! (Recall the critical fractile result)

Approach 2: (Naive) Predict-then-optimize

Take a two-step approach:

1. **Predict:** Use historical observations $(\boldsymbol{z}_i, \boldsymbol{D}_i)_{i \in [n]}$ to create a model for how $\boldsymbol{D}$ depends on $\boldsymbol{z}$, say $\hat{D} = f(\boldsymbol{z})$, where f is our trained model and $\hat{D}$ our prediction
2. **Optimize:** Solve the optimization problem assuming $\hat{D} = f(\boldsymbol{z})$, output the solution $x = \hat{D}$
3. But this is *obviously* suboptimal! (Recall the critical fractile result)
    - Reminder: for the newsvendor problem, the mean demand is not an optimal solution when overage cost does not equal the underage cost

Where did we go wrong?

Approach 2: (Naive) Predict-then-optimize

Take a two-step approach:

1. **Predict:** Use historical observations $(\boldsymbol{z}_i, \boldsymbol{D}_i)_{i \in [n]}$ to create a model for how $\boldsymbol{D}$ depends on $\boldsymbol{z}$, say $\hat{D} = f(\boldsymbol{z})$, where f is our trained model and $\hat{D}$ our prediction
2. **Optimize:** Solve the optimization problem assuming $\hat{D} = f(\boldsymbol{z})$, output the solution $x = \hat{D}$
3. But this is *obviously* suboptimal! (Recall the critical fractile result)
    - Reminder: for the newsvendor problem, the mean demand is not an optimal solution when overage cost does not equal the underage cost

Where did we go wrong? *The best prediction is not the best decision*

Approach 2: (Naive) Predict-then-optimize

Take a two-step approach:

1. **Predict:** Use historical observations $(\boldsymbol{z}_i, \boldsymbol{D}_i)_{i\in[n]}$ to create a model for how $\boldsymbol{D}$ depends on $\boldsymbol{z}$, say $\hat{D} = f(\boldsymbol{z})$, where f is our trained model and $\hat{D}$ our prediction
2. **Optimize:** Solve the optimization problem assuming $\hat{D} = f(\boldsymbol{z})$, output the solution $x = \hat{D}$
3. But this is *obviously* suboptimal! (Recall the critical fractile result)
    - Reminder: for the newsvendor problem, the mean demand is not an optimal solution when overage cost does not equal the underage cost

Where did we go wrong? *The best prediction is not the best decision*
Accounted for side information, but forgot to account for uncertainty.

Approach 2: (Naive) Predict-then-optimize

Take a two-step approach:

1. **Predict:** Use historical observations $(\boldsymbol{z}_i, \boldsymbol{D}_i)_{i \in [n]}$ to create a model for how $\boldsymbol{D}$ depends on $\boldsymbol{z}$, say $\hat{D} = f(\boldsymbol{z})$, where f is our trained model and $\hat{D}$ our prediction
2. **Optimize:** Solve the optimization problem assuming $\hat{D} = f(\boldsymbol{z})$, output the solution $x = \hat{D}$
3. But this is *obviously* suboptimal! (Recall the critical fractile result)
    - Reminder: for the newsvendor problem, the mean demand is not an optimal solution when overage cost does not equal the underage cost

Where did we go wrong? *The best prediction is not the best decision*
Accounted for side information, but forgot to account for uncertainty.

What we should do: leverage the data $\boldsymbol{z}$ to make the best decision possible. One approach for this: construct model of conditional distribution $D|\boldsymbol{z}$ from historical data, minimize sample-average approximation over conditional distribution.

Approach 2: (Naive) Predict-then-optimize

Take a two-step approach:

1. **Predict:** Use historical observations $(\boldsymbol{z}_i, \boldsymbol{D}_i)_{i\in[n]}$ to create a model for how $\boldsymbol{D}$ depends on $\boldsymbol{z}$, say $\hat{D} = f(\boldsymbol{z})$, where f is our trained model and $\hat{D}$ our prediction
2. **Optimize:** Solve the optimization problem assuming $\hat{D} = f(\boldsymbol{z})$, output the solution $x = \hat{D}$
3. But this is *obviously* suboptimal! (Recall the critical fractile result)
    - Reminder: for the newsvendor problem, the mean demand is not an optimal solution when overage cost does not equal the underage cost

Where did we go wrong? *The best prediction is not the best decision*
Accounted for side information, but forgot to account for uncertainty.

What we should do: leverage the data $\boldsymbol{z}$ to make the best decision possible. One approach for this: construct model of conditional distribution $D|\boldsymbol{z}$ from historical data, minimize sample-average approximation over conditional distribution.
Called *personalized SAA/contextual optimization*

Aside: How Ban and Rudin Solved This for Newsvendors

Approach 3: leverage knowledge of critical fractile result, train ML model to predict an optimal solution directly from context $\boldsymbol{z}$ using a linear decision rule

Aside: How Ban and Rudin Solved This for Newsvendors

Approach 3: leverage knowledge of critical fractile result, train ML model to predict an optimal solution directly from context $\boldsymbol{z}$ using a linear decision rule

Pros: optimal in large-sample settings, very efficient, nice guarantees. Solves the Newsvendor problem

Cons: unclear how to generalize to settings with constraints

Plan for Rest of Lecture

The "best" way of performing personalized SAA is (in my view) not fully resolved. Therefore, we discuss several approaches from the literature, and their pros/cons. Note that not all aspects of what we discuss today will be as satisfying as last week, since this isn't a solved problem.

Plan for Rest of Lecture

The "best" way of performing personalized SAA is (in my view) not fully resolved. Therefore, we discuss several approaches from the literature, and their pros/cons. Note that not all aspects of what we discuss today will be as satisfying as last week, since this isn't a solved problem.

Nonetheless, I think showing you things we don't know how to do yet is as important as things we do know how to do

Contextual Optimization: Full Problem Setting

- We have data $(\boldsymbol{D}^i, \boldsymbol{z}^i)_{i \in [n]}$ from observations of a stochastic process, where $\boldsymbol{D}$ is a random variable that appears in our optimization problem, and $\boldsymbol{z}$ is broadly predictive of $\boldsymbol{D}$

Contextual Optimization: Full Problem Setting

- We have data $(\boldsymbol{D}^i, \boldsymbol{z}^i)_{i \in [n]}$ from observations of a stochastic process, where $\boldsymbol{D}$ is a random variable that appears in our optimization problem, and $\boldsymbol{z}$ is broadly predictive of $\boldsymbol{D}$
- Given this data, and side information $\boldsymbol{z}$, we want to solve for

$$\boldsymbol{x}(\boldsymbol{z}) \in \arg\min_{\boldsymbol{x} \in \mathcal{X}} \quad \mathbb{E}[f(\boldsymbol{x}, \boldsymbol{D}) | \boldsymbol{Z} = \boldsymbol{z}],$$

  where $\mathcal{X}$ is our feasible region, f is our objective function

Contextual Optimization: Full Problem Setting

- We have data $(\boldsymbol{D}^i, \boldsymbol{z}^i)_{i \in [n]}$ from observations of a stochastic process, where $\boldsymbol{D}$ is a random variable that appears in our optimization problem, and $\boldsymbol{z}$ is broadly predictive of $\boldsymbol{D}$
- Given this data, and side information $\boldsymbol{z}$, we want to solve for

$$\boldsymbol{x}(\boldsymbol{z}) \in \arg\min_{\boldsymbol{x} \in \mathcal{X}} \quad \mathbb{E}[f(\boldsymbol{x}, \boldsymbol{D}) | \boldsymbol{Z} = \boldsymbol{z}],$$

  where $\mathcal{X}$ is our feasible region, f is our objective function
- In general, $\boldsymbol{x}(\boldsymbol{z})$ might need to be a function of $\boldsymbol{z}$, which makes optimizing over the space of policies $\boldsymbol{x}(\boldsymbol{z})$ hard

Before looking at methods, let's verify the importance of the problem setting by looking at more examples

Contextual Optimization: Variance-Based Portfolio Selection

Problem setting:

- Universe of p assets with random future returns r_i
- We want to pick $\boldsymbol{x} \in \mathbb{R}^{\boldsymbol{p}}_{+} : \boldsymbol{e}^\top \boldsymbol{x} = 1$ to minimize a weighted sum of variance minus expected return, given the context $\boldsymbol{z}$, which captures relevant side information (e.g., interest rates, oil prices)
- Formally:

$$\min_{\boldsymbol{x} \in \mathbb{R}^p_+ :\ \boldsymbol{e}^\top \boldsymbol{x} = 1, \gamma \in \mathbb{R}} \quad \mathbb{E}\left[\left(\sum_{i=1}^{p} x_i r_i - \gamma \right)^2 - \lambda \boldsymbol{r}^\top \boldsymbol{x} \middle| \boldsymbol{Z} = \boldsymbol{z} \right],$$

where λ balances the importance of risk/return, and γ is, at optimality, the conditional mean of $\boldsymbol{r}^\top \boldsymbol{x}$.

Contextual Optimization: Variance-Based Portfolio Selection

Problem setting:

- Universe of p assets with random future returns r_i
- We want to pick $\boldsymbol{x} \in \mathbb{R}_+^p : \boldsymbol{e}^\top \boldsymbol{x} = 1$ to minimize a weighted sum of variance minus expected return, given the context $\boldsymbol{z}$, which captures relevant side information (e.g., interest rates, oil prices)
- Formally:

$$\min_{\boldsymbol{x} \in \mathbb{R}_+^p:\ \boldsymbol{e}^\top \boldsymbol{x} = 1, \gamma \in \mathbb{R}} \quad \mathbb{E}\left[\left(\sum_{i=1}^p x_i r_i - \gamma\right)^2 - \lambda \boldsymbol{r}^\top \boldsymbol{x} \,\middle|\, \boldsymbol{Z} = \boldsymbol{z}\right],$$

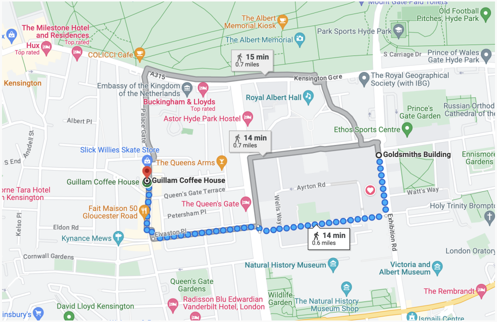
where λ balances the importance of risk/return, and γ is, at optimality, the conditional mean of $\boldsymbol{r}^\top \boldsymbol{x}$.

Quiz: who can tell me why first term is valid formulation of variance

$$\mathbb{V}[\boldsymbol{r}^\top \boldsymbol{x}] = \mathbb{E}[(\boldsymbol{r}^\top \boldsymbol{x} - \mathbb{E}[\boldsymbol{r}^\top \boldsymbol{x}])^2]$$

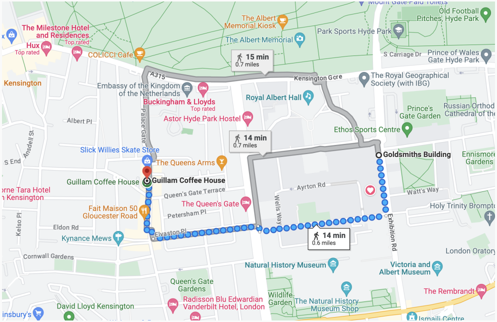
Answering the Real Questions: Getting Coffee Before Work

- Ryan is deciding whether he has time to get a coffee before work ☕
- He believes it will make him 2x as productive for the next 30 minutes



Answering the Real Questions: Getting Coffee Before Work

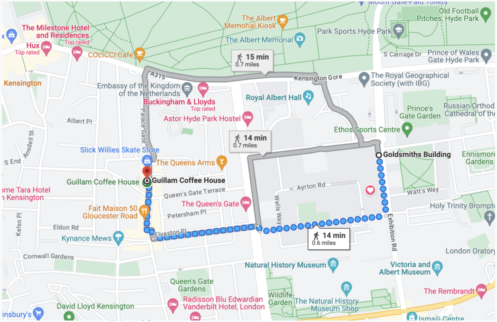
- Ryan is deciding whether he has time to get a coffee before work ☕
- He believes it will make him 2x as productive for the next 30 minutes



- Travel time is uncertain: if a Santander bike is available, it will take 20 minutes. Otherwise, he's walking, and it will take 40 mins.
- Assume a Santander bike is available w.p. 0.5: indifferent to coffee

Answering the Real Questions: Getting Coffee Before Work

- Ryan is deciding whether he has time to get a coffee before work ☕
- He believes it will make him 2x as productive for the next 30 minutes



- Travel time is uncertain: if a Santander bike is available, it will take 20 minutes. Otherwise, he's walking, and it will take 40 mins.
- Assume a Santander bike is available w.p. 0.5: indifferent to coffee
- Context: if Ryan's phone says there is currently a bike, the odds that one will be available in 5 mins time are much higher. So a valid decision rule is: if phone says bike available, get coffee

Improvement Strategy 1: Predictive to Prescriptive Analytics

Predictive to Prescriptive Analytics

Proposed by Bertsimas and Kallus (Management Science, 2020).
Two-step approach:

1. Use supervised learning to pick non-negative weights $w^i(\boldsymbol{z})$ to assign to each data point i such that $\sum_{i=1}^n w^i(\boldsymbol{z}) = 1 \forall\ \boldsymbol{z}$. Ideally, the weights $w^i(\boldsymbol{z})$ and the data points $\boldsymbol{D}_i$ comprise a good approximation to the conditional distribution $\boldsymbol{D}|\boldsymbol{Z} = \boldsymbol{z}$.
2. Optimize a sample-average approximation under this conditional distribution, i.e., solve

$$\boldsymbol{x}^\star(\boldsymbol{z}) \in \arg\min_{\boldsymbol{x} \in \mathcal{X}} \sum_{i=1}^n w^i(\boldsymbol{z}) f(\boldsymbol{x}, \boldsymbol{D}^i) \approx \arg\min_{\boldsymbol{x} \in \mathcal{X}} \mathbb{E}[f(\boldsymbol{x}, \boldsymbol{D})|\boldsymbol{Z} = \boldsymbol{z}]$$

Predictive to Prescriptive Analytics

Proposed by Bertsimas and Kallus (Management Science, 2020).
Two-step approach:

1. Use supervised learning to pick non-negative weights $w^i(\boldsymbol{z})$ to assign to each data point i such that $\sum_{i=1}^n w^i(\boldsymbol{z}) = 1 \forall\ \boldsymbol{z}$. Ideally, the weights $w^i(\boldsymbol{z})$ and the data points $\boldsymbol{D}_i$ comprise a good approximation to the conditional distribution $\boldsymbol{D}|\boldsymbol{Z} = \boldsymbol{z}$.
2. Optimize a sample-average approximation under this conditional distribution, i.e., solve

$$\boldsymbol{x}^\star(\boldsymbol{z}) \in \arg\min_{\boldsymbol{x}\in\mathcal{X}} \sum_{i=1}^n w^i(\boldsymbol{z}) f(\boldsymbol{x}, \boldsymbol{D}^i) \approx \arg\min_{\boldsymbol{x}\in\mathcal{X}} \mathbb{E}[f(\boldsymbol{x}, \boldsymbol{D})|\boldsymbol{Z} = \boldsymbol{z}]$$

Theorem: if $f(\boldsymbol{x}, \boldsymbol{D}^i)$ convex and $\mathcal{X}$ convex, can compute $\boldsymbol{x}^\star(\boldsymbol{z})$ in polynomial time.

The Approach in More Detail: Picking w

How do we pick w?

The Approach in More Detail: Picking w

How do we pick w?
According to Bertsimas and Kallus (2020):

- Keep the values $\boldsymbol{D}_i$ we observed from data
- Change the weights assigned to each point $\boldsymbol{D}_i$ depending on $\boldsymbol{z}$ (in SAA, $w_i = 1/n \; \forall i$), say $w^i(\boldsymbol{z})$

The Approach in More Detail: Picking w

How do we pick w?
According to Bertsimas and Kallus (2020):

- Keep the values $\boldsymbol{D}_i$ we observed from data
- Change the weights assigned to each point $\boldsymbol{D}_i$ depending on $\boldsymbol{z}$ (in SAA, $w_i = 1/n \; \forall i$), say $w^i(\boldsymbol{z})$
- How to assign weights? kNN, decision trees, random forests, ...

The Approach in More Detail: Picking w

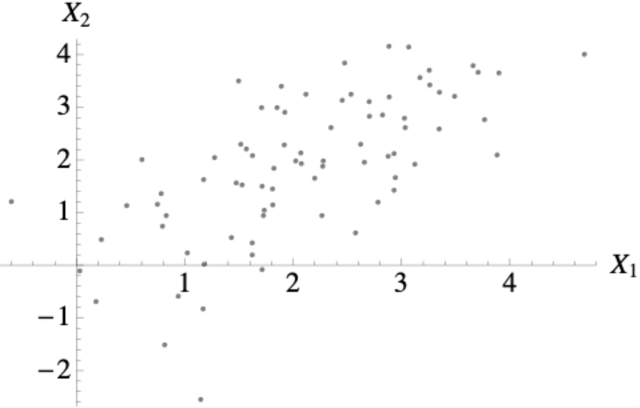
How do we pick w?
According to Bertsimas and Kallus (2020):

- Keep the values $\boldsymbol{D}_i$ we observed from data
- Change the weights assigned to each point $\boldsymbol{D}_i$ depending on $\boldsymbol{z}$ (in SAA, $w_i = 1/n \ \forall i$), say $w^i(\boldsymbol{z})$
- How to assign weights? kNN, decision trees, random forests, ...

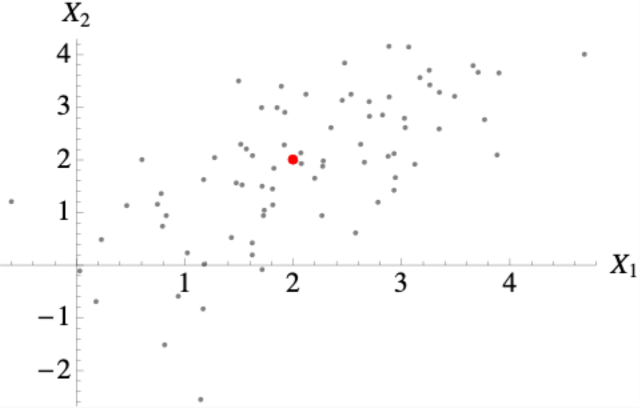
kNN case:

$$\min_{\boldsymbol{x} \in \mathcal{X}} \quad \sum_{i \in [n] : \boldsymbol{z}^i \text{ is a kNN of } \boldsymbol{z}} \frac{1}{k} f(\boldsymbol{x}, \boldsymbol{D}^i)$$

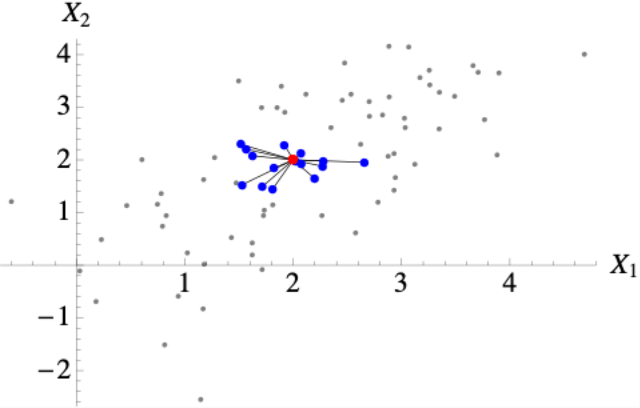
Fitting k nearest neighbors, visualized



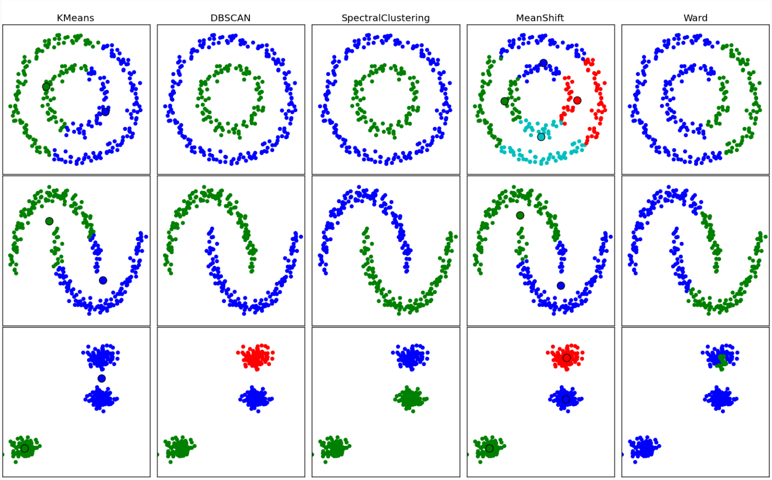
Fitting k nearest neighbors, visualized



Fitting the k nearest neighbors. visualized



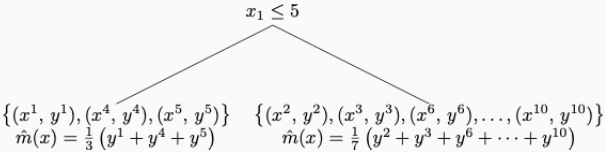
Many Clustering Strategies are Possible, But Use Caution



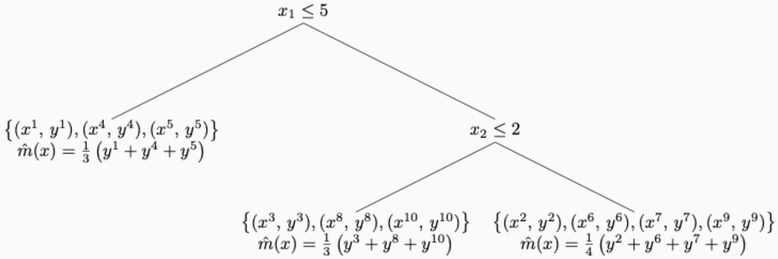
CART Approach

$$\{(x^1,\, y^1), (x^2,\, y^2), (x^2,\, y^2), (x^3,\, y^3), (x^4,\, y^4), (x^5,\, y^5), (x^6,\, y^6), (x^7,\, y^7), (x^8,\, y^8), (x^9,\, y^9), (x^{10},\, y^{10})\}$$
$$\hat{m}(x) = \frac{1}{10}\left(y^1 + y^2 + y^3 + y^4 + y^5 + y^6 + y^7 + y^8 + y^9 + y^{10}\right)$$

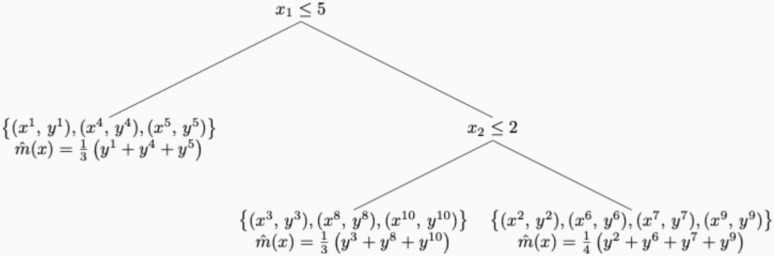
CART Approach



CART Approach



CART Approach



Implied binning rule: divide the region of feasible side information inputs, and use different policies depending on which region the side information falls into.

Random Forest Approach

Average over decision trees in forest, to “smooth out” dividing lines between feasible regions.

Aside: have you met random forests/CART/XGBoost etc. before?

Discussion: Advantages and Disadvantages of the Framework

- Pros: Conceptually simple—use ML to update the weights on the sample-average approximation, then apply SAA. Tractable. Materially improves on SAA in practice. Converges (a.s.) to an optimal policy of the unconditioned problem as n increases when the ML model is appropriate.

Discussion: Advantages and Disadvantages of the Framework

- Pros: Conceptually simple—use ML to update the weights on the sample-average approximation, then apply SAA. Tractable. Materially improves on SAA in practice. Converges (a.s.) to an optimal policy of the unconditioned problem as n increases when the ML model is appropriate.
- Cons: Fixing the data $\boldsymbol{D}_i$ and modifying the weights might leave something on the table: And not clear that a two-step approach is optimal vs. jointly optimizing the ML predictor and the optimization

Let’s take a break here.

Improvement Strategy 2: Smart "Predict Then Optimize"

Smart "Predict-then-Optimize"

Elmachtoub and Grigas (2022) study the following problem:

Given context $\boldsymbol{z}$, solve

$$\min_{\boldsymbol{x} \in \mathcal{X}} \mathbb{E}_{\boldsymbol{D} \sim \mathcal{D}_{\boldsymbol{z}}}[\boldsymbol{D}^\top \boldsymbol{x} | \boldsymbol{Z} = \boldsymbol{z}] = \mathbb{E}_{\boldsymbol{D} \sim \mathcal{D}_{\boldsymbol{z}}}[\boldsymbol{D} | \boldsymbol{Z} = \boldsymbol{z}]^\top \boldsymbol{x} \quad \text{(linearity of expectation)}$$

with goal of minimizing decision error on $\boldsymbol{D}^\top \boldsymbol{x}$, not prediction error on $\boldsymbol{x}$

Smart "Predict-then-Optimize"

Elmachtoub and Grigas (2022) study the following problem:

Given context $\boldsymbol{z}$, solve

$$\min_{\boldsymbol{x}\in\mathcal{X}} \mathbb{E}_{\boldsymbol{D}\sim\mathcal{D}_{\boldsymbol{z}}}[\boldsymbol{D}^\top \boldsymbol{x} | \boldsymbol{Z} = \boldsymbol{z}] = \mathbb{E}_{\boldsymbol{D}\sim\mathcal{D}_{\boldsymbol{z}}}[\boldsymbol{D} | \boldsymbol{Z} = \boldsymbol{z}]^\top \boldsymbol{x} \quad \text{(linearity of expectation)}$$

with goal of minimizing decision error on $\boldsymbol{D}^\top \boldsymbol{x}$, not prediction error on $\boldsymbol{x}$

Smart "Predict-then-Optimize"

To address problem, Elmachtoub and Grigas (2022) propose regret minimization. i.e., ensure good worst-case performance by minimizing quantities related to

$$c(\hat{\boldsymbol{D}}, \boldsymbol{D}) := \underbrace{\boldsymbol{D}^\top \boldsymbol{x}^\star(\hat{\boldsymbol{D}})}_{\text{cost using prediction}} - \underbrace{\boldsymbol{D}^\top \boldsymbol{x}^\star(\boldsymbol{D})}_{\text{cost if we predicted perfectly}},$$

where $\boldsymbol{x}^\star(\boldsymbol{D})$ is an optimal choice of $\boldsymbol{x}$ under realization $\boldsymbol{D}$ (take to be unique for convenience), $\hat{\boldsymbol{D}}$ is our predicted realization

Smart "Predict-then-Optimize"

To address problem, Elmachtoub and Grigas (2022) propose regret minimization. i.e., ensure good worst-case performance by minimizing quantities related to

$$c(\hat{\boldsymbol{D}}, \boldsymbol{D}) := \underbrace{\boldsymbol{D}^\top \boldsymbol{x}^\star(\hat{\boldsymbol{D}})}_{\text{cost using prediction}} - \underbrace{\boldsymbol{D}^\top \boldsymbol{x}^\star(\boldsymbol{D})}_{\text{cost if we predicted perfectly}},$$

where $\boldsymbol{x}^\star(\boldsymbol{D})$ is an optimal choice of $\boldsymbol{x}$ under realization $\boldsymbol{D}$ (take to be unique for convenience), $\hat{\boldsymbol{D}}$ is our predicted realization

Concretely, using the SAA/ERM principle, we ideally want to minimize

$$\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} c(f(\boldsymbol{z}_i), \boldsymbol{D}_i),$$

where f is predictor of $\hat{D}$, $\mathcal{H}$ is class of ML models we select f from

Smart "Predict-then-Optimize"

To address problem, Elmachtoub and Grigas (2022) propose regret minimization. i.e., ensure good worst-case performance by minimizing quantities related to

$$c(\hat{\boldsymbol{D}}, \boldsymbol{D}) := \underbrace{\boldsymbol{D}^\top \boldsymbol{x}^\star(\hat{\boldsymbol{D}})}_{\text{cost using prediction}} - \underbrace{\boldsymbol{D}^\top \boldsymbol{x}^\star(\boldsymbol{D})}_{\text{cost if we predicted perfectly}},$$

where $\boldsymbol{x}^\star(\boldsymbol{D})$ is an optimal choice of $\boldsymbol{x}$ under realization $\boldsymbol{D}$ (take to be unique for convenience), $\hat{\boldsymbol{D}}$ is our predicted realization

Concretely, using the SAA/ERM principle, we ideally want to minimize

$$\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} c(f(\boldsymbol{z}_i), \boldsymbol{D}_i),$$

where f is predictor of $\hat{D}$, $\mathcal{H}$ is class of ML models we select f from

Objective non-convex, usually intractable (could be discontinuous)

Smart “Predict-then-Optimize”

To address intractability, convexify the loss function c (details of precisely how this is a convexification are unimportant; see their paper)

$$\hat{c}(\hat{\boldsymbol{D}}, \boldsymbol{D}) = \max_{\boldsymbol{x} \in \mathcal{X}} \{(\boldsymbol{D} - 2\hat{\boldsymbol{D}})^\top \boldsymbol{x}\} + 2\boldsymbol{D}^\top \boldsymbol{x}^\star(\hat{\boldsymbol{D}}) - \boldsymbol{D}^\top \boldsymbol{x}^\star(\boldsymbol{D})$$

Smart “Predict-then-Optimize”

To address intractability, convexify the loss function c (details of precisely how this is a convexification are unimportant; see their paper)

$$\hat{c}(\hat{\boldsymbol{D}}, \boldsymbol{D}) = \max_{\boldsymbol{x} \in \mathcal{X}} \{(\boldsymbol{D} - 2\hat{\boldsymbol{D}})^\top \boldsymbol{x}\} + 2\boldsymbol{D}^\top \boldsymbol{x}^\star(\hat{\boldsymbol{D}}) - \boldsymbol{D}^\top \boldsymbol{x}^\star(\boldsymbol{D})$$

One can show that this loss is a differentiable convex surrogate of c

Smart "Predict-then-Optimize"

To address intractability, convexify the loss function c (details of precisely how this is a convexification are unimportant; see their paper)

$$\hat{c}(\hat{\boldsymbol{D}}, \boldsymbol{D}) = \max_{\boldsymbol{x} \in \mathcal{X}} \{(\boldsymbol{D} - 2\hat{\boldsymbol{D}})^\top \boldsymbol{x}\} + 2\boldsymbol{D}^\top \boldsymbol{x}^\star(\hat{\boldsymbol{D}}) - \boldsymbol{D}^\top \boldsymbol{x}^\star(\boldsymbol{D})$$

One can show that this loss is a differentiable convex surrogate of c

Therefore, solve

$$\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \hat{c}(f(\boldsymbol{z}_i), \boldsymbol{D}_i),$$

by, e.g., leveraging duality to reformulate it as a single optimization problem, or using gradient descent.

Let's Watch a 20-min Summary of SPO+ Paper, Then Discuss

20 minute summary video of their paper available [here]

Discussion: What do we think of SPO?

- Pros: achieves regret minimization under some conditions in the linear objective setting; can show asymptotic optimality guarantees

Discussion: What do we think of SPO?

- Pros: achieves regret minimization under some conditions in the linear objective setting; can show asymptotic optimality guarantees
- Cons: unclear what to do in the non-linear setting, since we have Jensen's inequality rather than linearity of expectation in that setting, other parts of the approach heavily leverage linearity

Discussion: What do we think of SPO?

- Pros: achieves regret minimization under some conditions in the linear objective setting; can show asymptotic optimality guarantees
- Cons: unclear what to do in the non-linear setting, since we have Jensen's inequality rather than linearity of expectation in that setting, other parts of the approach heavily leverage linearity
- See Ho-Nguyen and Kilinc-Karzan (MS, 2022) for a discussion of some positive and negative aspects of Elmachtoub and Grigas (2022)

Summary

- We saw a new and quite important problem setting today: contextual optimization
- We saw two proposals for obtaining good solutions to this problem, and discussed when they are applicable
- This is quite an active research area, so it's potentially a good one to work on a project for

Let’s take a break here.

Let's Look at Some Code on Prescriptive SAA For Next Part of Lecture

Further Reading

- The Big Data Newsvendor: Practical Insights from Machine Learning, Ban and Rudin (Operations Research, 2019)
- From Predictive to Prescriptive Analytics, Bertsimas and Kallus (Management Science, 2020)
- Smart "Predict Then Optimize", Elmachtoub and Grigas (Management Science, 2022)
- End-to-end Prediction and Optimization, Ho-Nguyen and Kilinc-Karzan (Management Science, 2022)