How to Play Strategically in Fantasy Sports (and Win)

Martin Haugh
Imperial College Business School

Columbia-Bloomberg Machine Learning in Finance Workshop
17^{st} May 2019

Based on joint work with Raghav Singal
(IE&OR Columbia University)

Motivation

- Daily fantasy sports (DFS) a multi-billion dollar industry
- Millions of annual users
- Approx $\$3.3$ billion in entry fees in 2017 in U.S.
- DraftKings and FanDuel represent approx 97% of U.S. market

Motivation

- Daily fantasy sports (DFS) a multi-billion dollar industry
- Millions of annual users
- Approx $\$3.3$ billion in entry fees in 2017 in U.S.
- DraftKings and FanDuel represent approx 97% of U.S. market

Our Problem: How to construct a portfolio of teams for a DFS contest.

Preliminaries

- **Athletes**
  - P real-world athletes ($P \approx 100$ to 500 in a given DFS contest).
  - Athletes performance denoted by $\boldsymbol{\delta} \in \mathbb{R}^P$. (Uncertainty $\#1$)

Preliminaries

- **Athletes**
  - P real-world athletes ($P \approx 100$ to 500 in a given DFS contest).
  - Athletes performance denoted by $\boldsymbol{\delta} \in \mathbb{R}^P$. (Uncertainty #1)

- **Decision**
  - Choose a team $\mathbf{w} \in \{0, 1\}^P$ of athletes.
  - $\mathbf{w} \in \mathbb{W}$ must satisfy budget, diversity, position constraints etc.
  - **Our points total**: $F := \mathbf{w}^\top \boldsymbol{\delta}$.
  - Can submit up to N teams.

Preliminaries

- **Athletes**
  - P real-world athletes ($P \approx 100$ to 500 in a given DFS contest).
  - Athletes performance denoted by $\boldsymbol{\delta} \in \mathbb{R}^P$. (Uncertainty #1)

- **Decision**
  - Choose a team $\mathbf{w} \in \{0,1\}^P$ of athletes.
  - $\mathbf{w} \in \mathbb{W}$ must satisfy budget, diversity, position constraints etc.
  - **Our points total**: $F := \mathbf{w}^\top \boldsymbol{\delta}$.
  - Can submit up to N teams.

- **Opponents**
  - O DFS opponents ($O \approx 1$ to $500,000$).
  - Opponents' entries: $\mathbf{W}_{\text{op}} := \{\mathbf{w}_o\}_{o=1}^{O}$. (Uncertainty #2)
  - **Opponents' points total**: $G_o := \mathbf{w}_o^\top \boldsymbol{\delta}$.

Reward Structures

Double-Up

- Top $r\%$ of teams each earn a payoff of R dollars.
- All other teams receive 0.

Reward Structures

Double-Up

- Top $r\%$ of teams each earn a payoff of R dollars.
- All other teams receive 0.

Top-Heavy

- Top few ranks win R_1, next few ranks win $R_2 < R_1$, and so on.
- R_1 could be as high as $1m.

Problem Formulations When $N = 1$

Denote by $G^{(r)}$ the r^{th} percentile of $\{G_o\}_{o=1}^{O}$.

- $G^{(r)}$ is the stochastic benchmark we need to beat in double-up contest.
- Depends on both $\boldsymbol{\delta}$ and $\mathbf{W}_{\text{op}}$.

Problem Formulations When $N = 1$

Denote by $G^{(r)}$ the r^{th} percentile of $\{G_o\}_{o=1}^{O}$.

- $G^{(r)}$ is the stochastic benchmark we need to beat in double-up contest.
- Depends on both $\boldsymbol{\delta}$ and $\mathbf{W}_{\text{op}}$.

Double-Up Formulation:

$$\max_{\mathbf{w} \in \mathbb{W}} \mathbb{P}\left\{ \underbrace{\mathbf{w}^\top \boldsymbol{\delta}}_{\text{our fantasy points}} > \underbrace{G^{(r)}(\mathbf{W}_{\text{op}}, \boldsymbol{\delta})}_{\text{stochastic benchmark}} \right\}$$

Problem Formulations When $N = 1$

Denote by $G^{(r)}$ the r^{th} percentile of $\{G_o\}_{o=1}^{O}$.

- $G^{(r)}$ is the stochastic benchmark we need to beat in double-up contest.
- Depends on both $\boldsymbol{\delta}$ and $\mathbf{W}_{\text{op}}$.

Double-Up Formulation:

$$\max_{\mathbf{w} \in \mathbb{W}} \mathbb{P} \left\{ \underbrace{\mathbf{w}^\top \boldsymbol{\delta}}_{\text{our fantasy points}} > \underbrace{G^{(r)}(\mathbf{W}_{\text{op}}, \boldsymbol{\delta})}_{\text{stochastic benchmark}} \right\}$$

Top-Heavy Formulation:

$$\max_{\mathbf{w} \in \mathbb{W}} \sum_{d=1}^{D} R_d \, \mathbb{P} \left\{ \mathbf{w}^\top \boldsymbol{\delta} > G^{(r'_d)}(\mathbf{W}_{\text{op}}, \boldsymbol{\delta}) \right\}$$

where the R_d's are decreasing in d.

Motivation

Problem Formulation

Related Work & Contributions

Modeling Opponents

Constructing Double-Up Portfolios

Constructing Top-Heavy Portfolios

Numerical Results

The Value of Insider Trading and Collusion

Conclusions and Further Research

Related Work & Contributions

Hunter, Vielma, Zaman (2016)

- Only consider winner-takes-all payoff structure.
- Propose a greedy MIP formulation to construct portfolio of teams
    - Each team targeted to have a high mean and variance
    - Teams designed to have low correlation
- Do not consider opponents behavior.

Related Work & Contributions

Hunter, Vielma, Zaman (2016)

- Only consider winner-takes-all payoff structure.
- Propose a greedy MIP formulation to construct portfolio of teams
  - Each team targeted to have a high mean and variance
  - Teams designed to have low correlation
- Do not consider opponents behavior.

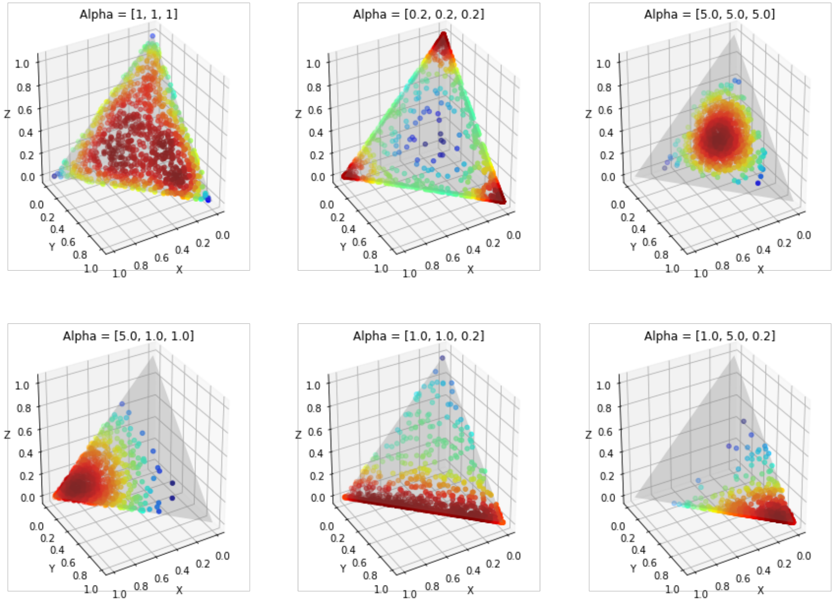
Our contributions

- Model exact payoff structure of contest.
- Model DFS opponents behavior leading to Dirichlet regressions.
- Connect to mean-variance theory on outperforming stochastic benchmarks.
- Optimal mean / variance trade-off determined via sequence of binary quadratic programs.
- Portfolio constructed via greedy algorithm motivated by parimutuel betting.
- Estimate value of insider trading and collusion.

The Dirichlet Distribution

- A Dirichlet distribution $\text{Dir}(\alpha_1, \ldots, \alpha_K)$ is a distribution on the $(K-1)$-dimensional simplex in $\mathbb{R}^K$.

- So a draw from $\text{Dir}(\alpha_1, \ldots, \alpha_K)$ yields a probability vector in $\mathbb{R}^K$.



Six Dirichlet distributions on the 2-dimensional simplex.
Source: `towardsdatascience.com`

Positional Marginals & Dirichlet Regression

Consider QB selection for DFS opponent's team:

- QB k selected with unknown probability p^k_{QB} for all k.

Brady: p^1_{QB}

Rodgers: p^2_{QB}

Stafford: p^3_{QB}

Wentz: p^4_{QB}

Positional Marginals & Dirichlet Regression

Consider QB selection for DFS opponent's team:

- QB k selected with unknown probability p_{QB}^{k} for all k.


Brady: p_{QB}^{1}


Rodgers: p_{QB}^{2}


Stafford: p_{QB}^{3}


Wentz: p_{QB}^{4}

- **QB positional marginal**: $\mathbf{p}_{\text{QB}} \sim \text{Dir}\left(\boldsymbol{\alpha}_{\text{QB}}\right)$.

Positional Marginals & Dirichlet Regression

Consider QB selection for DFS opponent's team:

- QB k selected with unknown probability p^k_{QB} for all k.

Brady: p^1_{QB}

Rodgers: p^2_{QB}

Stafford: p^3_{QB}

Wentz: p^4_{QB}

- **QB positional marginal**: $\mathbf{p}_{\text{QB}} \sim \text{Dir}\left(\boldsymbol{\alpha}_{\text{QB}}\right)$.
- Assume $\boldsymbol{\alpha}_{\text{QB}} = \exp\left(\mathbf{X}_{\text{QB}}\boldsymbol{\beta}_{\text{QB}}\right)$ where $\mathbf{X}_{\text{QB}}$ a feature matrix
  - $\boldsymbol{\beta}_{\text{QB}}$ estimated via Dirichlet regression.

Positional Marginals & Dirichlet Regression

Consider QB selection for DFS opponent's team:

- QB k selected with unknown probability p_{QB}^{k} for all k.

Brady: p_{QB}^{1}

Rodgers: p_{QB}^{2}

Stafford: p_{QB}^{3}

Wentz: p_{QB}^{4}

- **QB positional marginal**: $\mathbf{p}_{\text{QB}} \sim \text{Dir}\left(\boldsymbol{\alpha}_{\text{QB}}\right)$.
- Assume $\boldsymbol{\alpha}_{\text{QB}} = \exp\left(\mathbf{X}_{\text{QB}}\boldsymbol{\beta}_{\text{QB}}\right)$ where $\mathbf{X}_{\text{QB}}$ a feature matrix
  - $\boldsymbol{\beta}_{\text{QB}}$ estimated via Dirichlet regression.
- Now easy to generate QBs for $\mathbf{W}_{\text{op}}$ as Dirichlet-multinomial.

Positional Marginals & Dirichlet Regression

Consider QB selection for DFS opponent's team:

- QB k selected with unknown probability p^k_{QB} for all k.

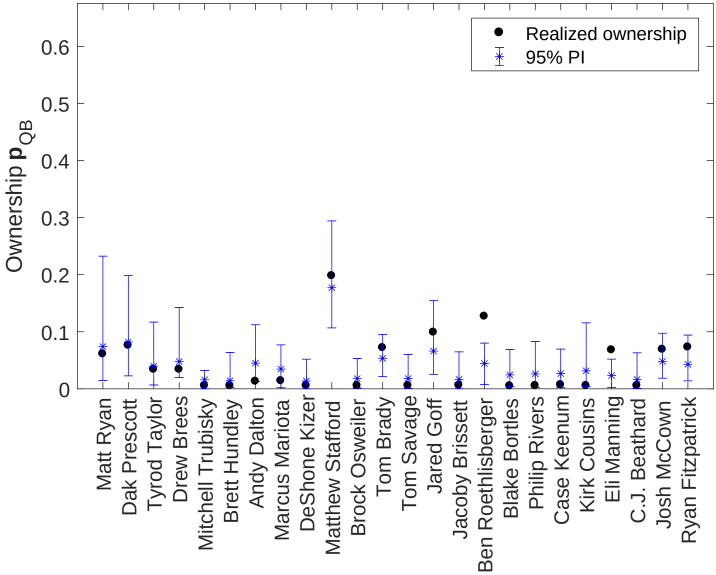



Brady: p^1_{QB} Rodgers: p^2_{QB} Stafford: p^3_{QB} Wentz: p^4_{QB}

- **QB positional marginal**: $\mathbf{p}_{\text{QB}} \sim \text{Dir}\left(\boldsymbol{\alpha}_{\text{QB}}\right)$.
- Assume $\boldsymbol{\alpha}_{\text{QB}} = \exp\left(\mathbf{X}_{\text{QB}}\boldsymbol{\beta}_{\text{QB}}\right)$ where $\mathbf{X}_{\text{QB}}$ a feature matrix
  - $\boldsymbol{\beta}_{\text{QB}}$ estimated via Dirichlet regression.
- Now easy to generate QBs for $\mathbf{W}_{\text{op}}$ as Dirichlet-multinomial.

Other positional marginals obtained similarly so easy to simulate $\mathbf{W}_{\text{op}}$ once some copula chosen.

Dirichlet Regression Results



Constructing Double-Up Portfolios

- Barring pathological cases, intuitively clear that optimal portfolio of N teams is to solve problem with a single entry and replicate N times.

Constructing Double-Up Portfolios

- Barring pathological cases, intuitively clear that optimal portfolio of N teams is to solve problem with a single entry and replicate N times.
- When $N = 1$ the double-up problem

$$\max_{\mathbf{w} \in \mathbb{W}} \mathbb{P} \left\{ \underbrace{\mathbf{w}^\top \boldsymbol{\delta}}_{\text{our fantasy points}} > \underbrace{G^{(r)}(\mathbf{W}_{\text{op}}, \boldsymbol{\delta})}_{\text{stochastic benchmark}} \right\}$$

can be restated as

$$\max_{\mathbf{w} \in \mathbb{W}} \mathbb{P} \left\{ Y_{\mathbf{w}} > 0 \right\}$$

where $Y_{\mathbf{w}} := \mathbf{w}^\top \boldsymbol{\delta} - G^{(r)}$.

Constructing Double-Up Portfolios

- Barring pathological cases, intuitively clear that optimal portfolio of N teams is to solve problem with a single entry and replicate N times.
- When $N = 1$ the double-up problem

$$\max_{\mathbf{w} \in \mathbb{W}} \mathbb{P} \left\{ \underbrace{\mathbf{w}^\top \boldsymbol{\delta}}_{\text{our fantasy points}} > \underbrace{G^{(r)}(\mathbf{W}_{\text{op}}, \boldsymbol{\delta})}_{\text{stochastic benchmark}} \right\}$$

can be restated as

$$\max_{\mathbf{w} \in \mathbb{W}} \mathbb{P} \{ Y_{\mathbf{w}} > 0 \}$$

where $Y_{\mathbf{w}} := \mathbf{w}^\top \boldsymbol{\delta} - G^{(r)}$.

- Adopt a mean-variance approach to solve for $\mathbf{w}^*$
  - follow Morton et al (2003) on outperforming a stochastic benchmark.

Algorithm 1 For the Double-Up Problem with $N = 1$

```
if ∃w ∈ 𝕎 with μ_{Y_w} ≥ 0 then
    for all λ ∈ Λ do
        w_λ = argmax_{w∈𝕎, μ_{Y_w}≥0} { μ_{Y_w} − λ σ²_{Y_w} }
    end for
```

if $\exists \mathbf{w} \in \mathbb{W}$ with $\mu_{Y_\mathbf{w}} \geq 0$ **then**

  for all $\lambda \in \Lambda$ **do**

    $\mathbf{w}_\lambda = \underset{\mathbf{w}\in\mathbb{W},\, \mu_{Y_\mathbf{w}}\geq 0}{\text{argmax}} \left\{ \mu_{Y_\mathbf{w}} - \lambda \sigma^2_{Y_\mathbf{w}} \right\}$

  end for

Algorithm 1 For the Double-Up Problem with $N = 1$

```
if ∃w ∈ 𝕎 with μ_{Y_w} ≥ 0 then
    for all λ ∈ Λ do
        w_λ = argmax_{w∈𝕎, μ_{Y_w}≥0} { μ_{Y_w} − λ σ²_{Y_w} }
    end for
else
    for all λ ∈ Λ do
        w_λ = argmax_{w∈𝕎} { μ_{Y_w} + λ σ²_{Y_w} }
    end for
end if
```

- **if** $\exists \mathbf{w} \in \mathbb{W}$ with $\mu_{Y_\mathbf{w}} \geq 0$ **then**
  - **for all** $\lambda \in \Lambda$ **do**
    - $\mathbf{w}_\lambda = \operatorname*{argmax}_{\mathbf{w}\in\mathbb{W},\, \mu_{Y_\mathbf{w}}\geq 0} \left\{\mu_{Y_\mathbf{w}} - \lambda\sigma^2_{Y_\mathbf{w}}\right\}$
  - **end for**
- **else**
  - **for all** $\lambda \in \Lambda$ **do**
    - $\mathbf{w}_\lambda = \operatorname*{argmax}_{\mathbf{w}\in\mathbb{W}} \left\{\mu_{Y_\mathbf{w}} + \lambda\sigma^2_{Y_\mathbf{w}}\right\}$
  - **end for**
- **end if**

Algorithm 1 For the Double-Up Problem with $N = 1$

if $\exists \mathbf{w} \in \mathbb{W}$ with $\mu_{Y_\mathbf{w}} \geq 0$ **then**
    for all $\lambda \in \Lambda$ **do**
        $\mathbf{w}_\lambda = \underset{\mathbf{w} \in \mathbb{W},\, \mu_{Y_\mathbf{w}} \geq 0}{\operatorname{argmax}} \left\{ \mu_{Y_\mathbf{w}} - \lambda \sigma^2_{Y_\mathbf{w}} \right\}$
    end for
else
    for all $\lambda \in \Lambda$ **do**
        $\mathbf{w}_\lambda = \underset{\mathbf{w} \in \mathbb{W}}{\operatorname{argmax}} \left\{ \mu_{Y_\mathbf{w}} + \lambda \sigma^2_{Y_\mathbf{w}} \right\}$
    end for
end if
$\lambda^* = \underset{\lambda \in \Lambda}{\operatorname{argmax}}\, \mathbb{P}\{Y_{\mathbf{w}_\lambda} > 0\}$
return $\mathbf{w}_{\lambda^*}$

Algorithm 1 For the Double-Up Problem with $N = 1$

if $\exists \mathbf{w} \in \mathbb{W}$ with $\mu_{Y_\mathbf{w}} \geq 0$ **then**
  for all $\lambda \in \Lambda$ **do**
    $\mathbf{w}_\lambda = \underset{\mathbf{w} \in \mathbb{W},\, \mu_{Y_\mathbf{w}} \geq 0}{\text{argmax}} \left\{ \mu_{Y_\mathbf{w}} - \lambda \sigma^2_{Y_\mathbf{w}} \right\}$
  end for
else
  for all $\lambda \in \Lambda$ **do**
    $\mathbf{w}_\lambda = \underset{\mathbf{w} \in \mathbb{W}}{\text{argmax}} \left\{ \mu_{Y_\mathbf{w}} + \lambda \sigma^2_{Y_\mathbf{w}} \right\}$
  end for
end if
$\lambda^* = \underset{\lambda \in \Lambda}{\text{argmax}}\ \mathbb{P}\{Y_{\mathbf{w}_\lambda} > 0\}$
return $\mathbf{w}_{\lambda^*}$

- Algorithm 1 requires solving a series of binary quadratic programs.
- Optimal if mean-variance assumption holds.

Constructing Top-Heavy Portfolios

- Easy to adapt Algorithm 1 for top-heavy $N = 1$ problem.
- But what to do for $N > 1$?

Constructing Top-Heavy Portfolios

- Easy to adapt Algorithm 1 for top-heavy $N = 1$ problem.
- But what to do for $N > 1$?
- Consider following idealized greedy algorithm.

Algorithm 2 Idealized Greedy Algorithm for Construction of Top-Heavy Portfolio

$\mathbf{W}^* = \emptyset$
for all $i = 1, \ldots, N$ **do**
    $\mathbf{w}_i^* = \operatorname*{argmax}_{\mathbf{w} \in \mathbb{W}} \mathcal{R}eward(\mathbf{W}^* \cup \mathbf{w})$
    $\mathbf{W}^* = \mathbf{W}^* \cup \{\mathbf{w}_i^*\}$
end for
return $\mathbf{W}^*$

Support for Idealized Greedy Algorithm

1. Consider parimutuel betting - a specialized case of a DFS contest where:
   - Team (horse) budget = $\$1$
   - Every player costs $1
   - Exactly one player per team.
   - We know probabilities and opponents bets.

Support for Idealized Greedy Algorithm

1. Consider parimutuel betting - a specialized case of a DFS contest where:
   - Team (horse) budget = $\$1$
   - Every player costs $\$1$
   - Exactly one player per team.
   - We know probabilities and opponents bets.

   Can show greedy algorithm optimal in this setting.

Support for Idealized Greedy Algorithm

1. Consider parimutuel betting - a specialized case of a DFS contest where:
   - Team (horse) budget = $\$1$
   - Every player costs $\$1$
   - Exactly one player per team.
   - We know probabilities and opponents bets.

   Can show greedy algorithm optimal in this setting.

2. Top-heavy objective is monotone submodular
   - By Nemhauser et al. (1978) portfolio returned by greedy algorithm achieves $\geq 63\%$ of true unknown optimal portfolio.

Support for Idealized Greedy Algorithm

1. Consider parimutuel betting - a specialized case of a DFS contest where:
   - Team (horse) budget = $\$1$
   - Every player costs $1
   - Exactly one player per team.
   - We know probabilities and opponents bets.

   Can show greedy algorithm optimal in this setting.

2. Top-heavy objective is monotone submodular
   - By Nemhauser et al. (1978) portfolio returned by greedy algorithm achieves $\geq 63\%$ of true unknown optimal portfolio.

Problem: Cannot find $\mathbf{w}_i^*$ when $i > 1$.

Some Observations for General DFS Contest

- Still want to go "long" mean and variance.

Some Observations for General DFS Contest

- Still want to go "long" mean and variance.
- But also don't want teams in portfolio competing with each other.

Some Observations for General DFS Contest

- Still want to go "long" mean and variance.
- But also don't want teams in portfolio competing with each other.
- Easy to see in winner-takes-all setting:

Some Observations for General DFS Contest

- Still want to go "long" mean and variance.
- But also don't want teams in portfolio competing with each other.
- Easy to see in winner-takes-all setting:
  - Suppose O "small" relative to number of feasible teams.

Some Observations for General DFS Contest

- Still want to go "long" mean and variance.
- But also don't want teams in portfolio competing with each other.
- Easy to see in winner-takes-all setting:
  - Suppose O "small" relative to number of feasible teams.
  - Why replicate an entry when chances are no-one else has picked it?

Some Observations for General DFS Contest

- Still want to go "long" mean and variance.
- But also don't want teams in portfolio competing with each other.
- Easy to see in winner-takes-all setting:
  - Suppose O "small" relative to number of feasible teams.
  - Why replicate an entry when chances are no-one else has picked it?
  - Why even pick a "nearby", i.e. highly correlated, entry?

Some Observations for General DFS Contest

- Still want to go "long" mean and variance.
- But also don't want teams in portfolio competing with each other.
- Easy to see in winner-takes-all setting:
  - Suppose O "small" relative to number of feasible teams.
  - Why replicate an entry when chances are no-one else has picked it?
  - Why even pick a "nearby", i.e. highly correlated, entry?

Conclusion: Want to choose a diversified portfolio of teams where each team's fantasy points score has high mean and variance.

Algorithm 3 Top-Heavy Optimization for N Entries

$\mathbf{W}^* = \varnothing$

for all $i = 1, \ldots, N$ **do**

  for all $\lambda \in \Lambda$ **do**

    $\mathbf{w}_\lambda = \operatorname{argmax}_{\mathbf{w} \in \mathbb{W}} \left\{ \mu_{Y_\mathbf{w}} + \lambda\, \sigma^2_{Y_\mathbf{w}} \right\}$

  end for

Algorithm 3 Top-Heavy Optimization for N Entries

$\mathbf{W}^* = \varnothing$

for all $i = 1, \dots, N$ **do**

  for all $\lambda \in \Lambda$ **do**

    $\mathbf{w}_\lambda = \operatorname{argmax}_{\mathbf{w} \in \mathbb{W}} \left\{ \mu_{Y_\mathbf{w}} + \lambda \, \sigma^2_{Y_\mathbf{w}} \right\}$

  end for

  $\lambda^* = \operatorname{argmax}_{\lambda \in \Lambda} \mathcal{R}eward(\mathbf{W}^* \cup \mathbf{w}_\lambda)$

  $\mathbf{W}^* = \mathbf{W}^* \cup \{ \mathbf{w}_{\lambda^*} \}$

Algorithm 3 Top-Heavy Optimization for N Entries

$\mathbf{W}^* = \emptyset$

for all $i = 1, \ldots, N$ **do**

  for all $\lambda \in \Lambda$ **do**

    $\mathbf{w}_\lambda = \underset{\mathbf{w} \in \mathbb{W}}{\text{argmax}} \left\{ \mu_{Y_\mathbf{w}} + \lambda \, \sigma^2_{Y_\mathbf{w}} \right\}$

  end for

  $\lambda^* = \underset{\lambda \in \Lambda}{\text{argmax}} \, \mathcal{R}eward(\mathbf{W}^* \cup \mathbf{w}_\lambda)$

  $\mathbf{W}^* = \mathbf{W}^* \cup \{\mathbf{w}_{\lambda^*}\}$

  $\mathbb{W} = \mathbb{W} \cap \{\mathbf{w} : \mathbf{w}^\top \mathbf{w}_i^* \leq \gamma\}$ % diversification constraint for next entry

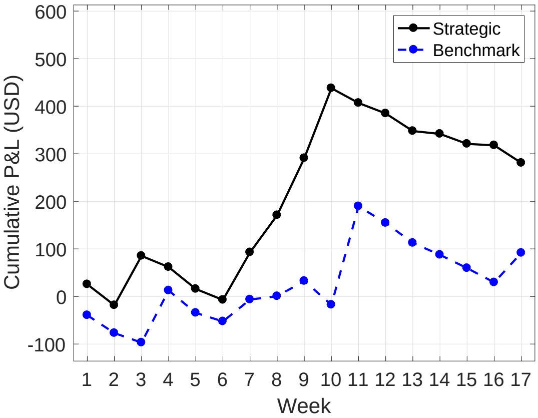
Algorithm 3 Top-Heavy Optimization for N Entries

$\mathbf{W}^* = \varnothing$
for all $i = 1, \dots, N$ **do**
  for all $\lambda \in \Lambda$ **do**
    $\mathbf{w}_\lambda = \underset{\mathbf{w} \in \mathbb{W}}{\text{argmax}} \left\{ \mu_{Y_\mathbf{w}} + \lambda \, \sigma^2_{Y_\mathbf{w}} \right\}$
  end for
  $\lambda^* = \underset{\lambda \in \Lambda}{\text{argmax}} \, \mathcal{R}eward(\mathbf{W}^* \cup \mathbf{w}_\lambda)$
  $\mathbf{W}^* = \mathbf{W}^* \cup \{\mathbf{w}_{\lambda^*}\}$
  $\mathbb{W} = \mathbb{W} \cap \{\mathbf{w} : \mathbf{w}^\top \mathbf{w}_i^* \le \gamma\}$ % diversification constraint for next entry
end for
return $\mathbf{W}^*$

Results

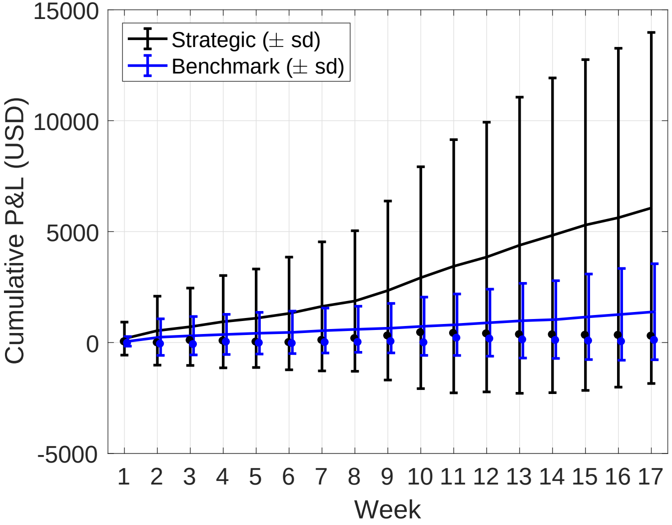
- Participated at FanDuel during the 2017-18 NFL season.
- Main focus on top-heavy for experiments.
- Benchmark model similar to Hunter, Vielma, and Zaman (2016).
- Invested $\$50$ per week for each of the two models with $N = 50$.

ROI of Over 350% in Just 17 Weeks!



Cumulative realized dollar P&Ls in top-heavy contests during 2017 NFL season with $N = 50$

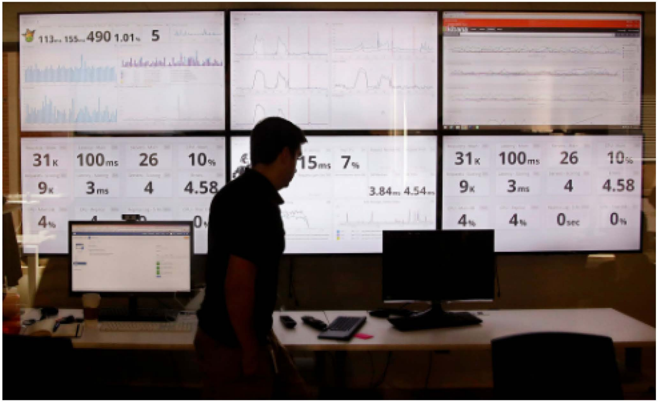
But a Very High Variance!



Predicted and realized cumulative P&L for the strategic and benchmark models for all seventeen weeks of the FanDuel DFS contests in the 2017 NFL regular season.

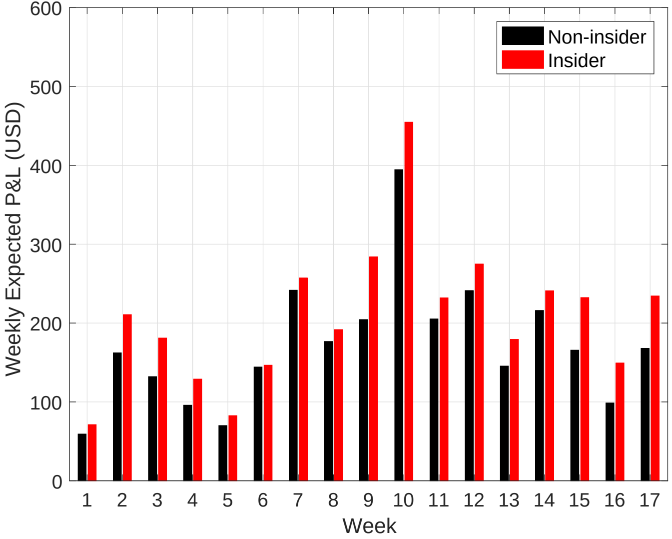
Scandal Erupts in Unregulated World of Fantasy Sports

By JOE DRAPE and JACQUELINE WILLIAMS OCT. 5, 2015



An employee in the DraftKings offices last month. DraftKings and FanDuel said "both companies have strong policies in place to ensure that employees do not misuse any information at their disposal."
Stephan Savoia/Associated Press

The Value of Insider Trading



Weekly expected P&L for the strategic model ($N = 50$) with and without inside information $\mathbf{p}$ in the top-heavy series.

THE WALL STREET JOURNAL.
U.S. Edition | April 28, 2018 | Today's Paper | Video

SPORTS

Fantasy-Sports Player Cleared in Collusion Case

DraftKings finds no wrongdoing after investigating whether co-winner of $1 million prize improperly w

The Value of Collusion

Consider following stylized model of collusion / non-collusion:

The Value of Collusion

Consider following stylized model of collusion / non-collusion:

- Colluders submit optimal portfolio of $N = E_{\text{max}} \times N_{\text{collude}}$ teams.

The Value of Collusion

Consider following stylized model of collusion / non-collusion:

- Colluders submit optimal portfolio of $N = E_{\text{max}} \times N_{\text{collude}}$ teams.
- Non-colluders submit optimal portfolio of $N = E_{\text{max}}$ teams replicated N_{collude} times.

The Value of Collusion

Consider following stylized model of collusion / non-collusion:

- Colluders submit optimal portfolio of $N = E_{\text{max}} \times N_{\text{collude}}$ teams.
- Non-colluders submit optimal portfolio of $N = E_{\text{max}}$ teams replicated N_{collude} times.

	Expected P&L (USD)			Probability of Loss		
$N_{\textbf{collude}}$	NC	C	Increase	NC	C	Decrease
1	6,053	6,053	0%	0.49	0.49	0%
2	9,057	10,240	13%	0.49	0.47	4%
3	10,975	13,776	26%	0.49	0.46	6%
4	12,411	16,883	36%	0.49	0.46	7%
5	13,632	19,677	44%	0.49	0.45	8%

Total expected dollar P&L (over 17 weeks) and average weekly probability of loss related to the top-heavy contests for both the non-colluding ("NC") and colluding ("C") portfolios with $E_{\text{max}} = 50$ and $N_{\text{collude}} \in \{1, \ldots, 5\}$.

The Value of Collusion

Consider following stylized model of collusion / non-collusion:

- Colluders submit optimal portfolio of $N = E_{\text{max}} \times N_{\text{collude}}$ teams.
- Non-colluders submit optimal portfolio of $N = E_{\text{max}}$ teams replicated N_{collude} times.

	Expected P&L (USD)			**Probability of Loss**		
N_{collude}	NC	C	Increase	NC	C	Decrease
1	6,053	6,053	0%	0.49	0.49	0%
2	9,057	10,240	13%	0.49	0.47	4%
3	10,975	13,776	26%	0.49	0.46	6%
4	12,411	16,883	36%	0.49	0.46	7%
5	13,632	19,677	44%	0.49	0.45	8%

Total expected dollar P&L (over 17 weeks) and average weekly probability of loss related to the top-heavy contests for both the non-colluding ("NC") and colluding ("C") portfolios with $E_{\text{max}} = 50$ and $N_{\text{collude}} \in \{1, \dots, 5\}$.

Caveat: Actual value of collusion likely much smaller.

Conclusions

- Developed a new framework for DFS team selection.
- Model opponent behaviour via Dirichlet regression.
- Leveraged mean-variance theory from finance.
- Results from parimutuel betting and submodular maximization motivate greedy algorithm for constructing portfolio of N entries.
- Demonstrated value in real-world contests.
- Can estimate value of insider trading and / or collusion.

Ongoing Research

- Test on other sports (baseball, basketball, ice hockey)
  - Very high variance in NFL contests due to injuries, roster size, weather, etc.
  - Only 16 games per team so also high seasonal variance.

- Actively update parameter estimates
  - Lots of news comes out just before games
  - Witnessed instances when reacting to such news would have been beneficial and possible.

- Improved Monte-Carlo algorithms.

- Heuristics for re-optimizing portfolios in event of late-breaking news.

- What if the opponents are strategic too?
  - handle this to some extent via stacking copula.

Thank you!

QB Matthew Stafford

GB 11 @ **DET** 35

FINAL

$7,800 SALARY

8.9% OWNED

27.12

RB Alex Collins

CIN 31 @ **BAL** 27

FINAL

$6,800 SALARY

9.6% OWNED

16.6

RB Dion Lewis

NYJ 6 @ **NE** 26

FINAL

$7,200 SALARY

25.3% OWNED

28.3

WR JuJu Smith-Schuster

CLE 24 @ **PIT** 28

FINAL

$7,300 SALARY

8.8% OWNED

30.8

WR Marvin Jones Jr.

GB 11 @ **DET** 35

FINAL

$7,300 SALARY

12.3% OWNED

16.1

WR Keenan Allen

OAK 10 @ **LAC** 30

FINAL

$8,600 SALARY

27% OWNED

29.8

TE Jack Doyle

HOU 13 @ **IND** 22

FINAL

Monte-Carlo and Order Statistics

Need to estimate $\mu_{G^{(r)}}$, $\sigma^2_{G^{(r)}}$, $\boldsymbol{\sigma}_{\boldsymbol{\delta},G^{(r)}}$ for various algorithms.

Monte-Carlo and Order Statistics

Need to estimate $\mu_{G^{(r)}}$, $\sigma^2_{G^{(r)}}$, $\boldsymbol{\sigma}_{\boldsymbol{\delta},G^{(r)}}$ for various algorithms.

- Can use Monte-Carlo to simulate a sample of $(\boldsymbol{\delta}, \mathbf{p}, \mathbf{W}_{\text{op}})$ and hence a sample of $(\boldsymbol{\delta}, G^{(r)})$.
- So generate many samples and use them to estimate $\mu_{G^{(r)}}$, $\sigma^2_{G^{(r)}}$, $\boldsymbol{\sigma}_{\boldsymbol{\delta},G^{(r)}}$.

Monte-Carlo and Order Statistics

Need to estimate $\mu_{G^{(r)}}$, $\sigma^2_{G^{(r)}}$, $\boldsymbol{\sigma}_{\boldsymbol{\delta},G^{(r)}}$ for various algorithms.

- Can use Monte-Carlo to simulate a sample of $(\boldsymbol{\delta}, \mathbf{p}, \mathbf{W}_{\text{op}})$ and hence a sample of $(\boldsymbol{\delta}, G^{(r)})$.
- So generate many samples and use them to estimate $\mu_{G^{(r)}}$, $\sigma^2_{G^{(r)}}$, $\boldsymbol{\sigma}_{\boldsymbol{\delta},G^{(r)}}$.

Problem: Generating $\mathbf{W}_{\text{op}}$ is expensive when O large.

Monte-Carlo and Order Statistics

Need to estimate $\mu_{G^{(r)}}$, $\sigma^2_{G^{(r)}}$, $\boldsymbol{\sigma}_{\boldsymbol{\delta},G^{(r)}}$ for various algorithms.

- Can use Monte-Carlo to simulate a sample of $(\boldsymbol{\delta}, \mathbf{p}, \mathbf{W}_{\text{op}})$ and hence a sample of $(\boldsymbol{\delta}, G^{(r)})$.
- So generate many samples and use them to estimate $\mu_{G^{(r)}}$, $\sigma^2_{G^{(r)}}$, $\boldsymbol{\sigma}_{\boldsymbol{\delta},G^{(r)}}$.

Problem: Generating $\mathbf{W}_{\text{op}}$ is expensive when O large.

Solution

- $\mathbf{W}_{\text{op}}$ only affects $G^{(r)}$ so much easier if we can sample $G^{(r)}$ directly.

Monte-Carlo and Order Statistics

Need to estimate $\mu_{G^{(r)}}$, $\sigma^2_{G^{(r)}}$, $\boldsymbol{\sigma}_{\boldsymbol{\delta},G^{(r)}}$ for various algorithms.

- Can use Monte-Carlo to simulate a sample of $(\boldsymbol{\delta}, \mathbf{p}, \mathbf{W}_{\text{op}})$ and hence a sample of $(\boldsymbol{\delta}, G^{(r)})$.
- So generate many samples and use them to estimate $\mu_{G^{(r)}}$, $\sigma^2_{G^{(r)}}$, $\boldsymbol{\sigma}_{\boldsymbol{\delta},G^{(r)}}$.

Problem: Generating $\mathbf{W}_{\text{op}}$ is expensive when O large.

Solution

- $\mathbf{W}_{\text{op}}$ only affects $G^{(r)}$ so much easier if we can sample $G^{(r)}$ directly.
- Since $G_o \mid (\boldsymbol{\delta}, \mathbf{p})$ IID for $o = 1, \ldots, O$ order statistics theory implies

$$G^{(qO)} \mid (\boldsymbol{\delta}, \mathbf{p}) \xrightarrow{p} F^{-1}_{G|(\boldsymbol{\delta},\mathbf{p})}(q) \qquad \text{as} \;\; O \to \infty$$

- So just simulate $(\boldsymbol{\delta}, \mathbf{p})$, then estimate CDF $F_{G|(\boldsymbol{\delta},\mathbf{p})}$ to obtain $(\boldsymbol{\delta}, \mathbf{p}, G^{(r)})$.

Other improvements also used. e.g. splitting.