Appendix A: Sample Selection

The data used in this paper is a subset of a larger dataset that TELCO collected to study households' response to free trials of the cinema pack product. While our interest in this paper is to understand the behavior of the population of BitTorrent users, TELCO was also interested in learning if the average household would be more likely to subscribe to Cinema Pack after a full featured trial of the service.

To ensure their goals, TELCO used a stratified sampling to learn whether offering the new TV content would lead households to subscribe to the product afterwards, and to use less Internet data and reduce piracy.

With stratified sampling, the units of observation are split into stratum and are randomly assigned to the treatment and control in each stratum separately (Simon 1979). This design allows TELCO to increase statistical power, in particular to the sub-population of pirates (Assmann et al. 2000) without compromising the generalizability of the analysis to the entire population of client households.

TELCO used data from April and May 2014 (before the experiment started) to build a classification algorithm to stratify a sample of households according to observable features that correlate to BitTorrent use. The Caret framework was used to train and evaluate the performance of different machine learning algorithms (Kuhn 2008) on their ability to predict if a household would show up in future BitTorrent logs. All algorithms were trained and tested using 5 fold cross validation repeated 10 times. The outcome of this analysis is depicted in Figure 9. This figure shows that most models fit the data well. In particular, the Area Under the Curve (AUC) is near or above 80% in all cases. This threshold is usually used as rule of thumb to assume that a model is good for predictive purposes(Swets et al. 1988).

Variable selection is an integral part of gradient boosted model trees (GBM) (Friedman 2001), random forests (RFOREST) (Breiman 2001) and classification and regression trees (RPART) (Breiman et al. 1984). For Support Vector Machines with radial Kernel (SVM) (Hearst et al. 1998, Suykens and Vandewalle 1999) and for the Logit model, feature selection is a separate step. These models were trained using LASSO (Tibshirani 1996) for feature selection. GBM was used to stratify the household sample because it yielded better performance scores in all the metrics usually used to gauge the predictive performance of these algorithms. Using the output of this model, the population classifier was constructed such that households with GBM scores above 50% were marked as pirates, while households with GBM scores below 50% were marked as non-pirates. Figure 10 plots the ROC curve we obtained. The black dot identifies the classifier used to stratify households.

After classifying households TELCO looked for whether they showed up in the BitTorrent logs. This allowed for creating four household strata. Households that were found in the BitTorrent logs were called "Confirmed Pirates" – C. Otherwise they were marked as non-pirates – NC. Households that the algorithm predicted as being pirates were called "Predicted Pirates" – P. Otherwise, they were marked as non-pirates – NP. Therefore, the four strata considered were (C,P), (C,NP), (NC,P) and (NC,NP). Figure 11 summarizes the features that the GBM algorithm used to classify households as pirates and non pirates. This figure shows that Internet upload traffic is the main determinant for this characterization, followed by how long ago households subscribed Internet service and by whether they have legacy or up-to-date equipment. We

note that in this paper we end up analyzing only confirmed pirates, that is, strata (C,NP) and (C,P). Also we only focus on households with up-to-date equipment because households with legacy set-top-box devices cannot be tracked with respect to their television viewing habits.

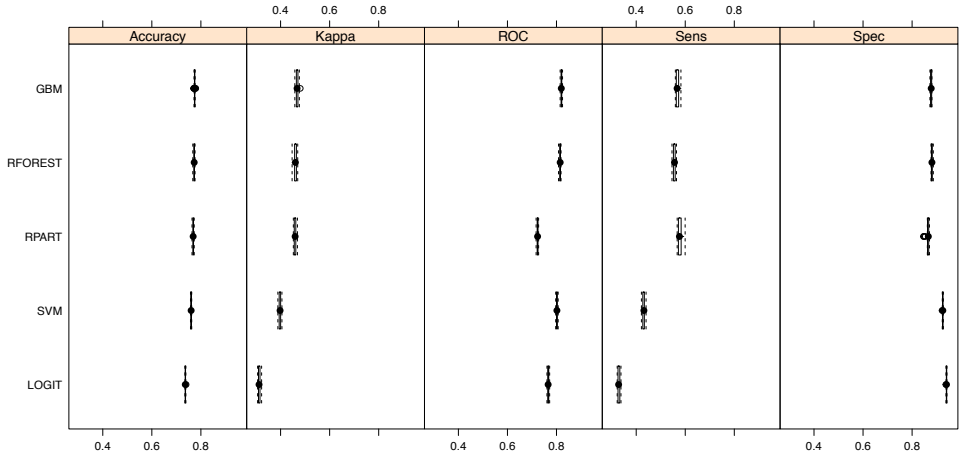


Figure 9 Performance of several machine learning algorithms used.

Table 12 shows the average daily amount of traffic downloaded and TV usage per stratum in April and May 2014. In the absence of priors for the potential effect of treatment, TELCO assumed that, on average, treated households would watch their preferred TV show on TV rather than download it illegally from the Internet. Identifying a smaller effect is arguably uninteresting from an economic point of view. According to Netflix, the average TV show consumes 450 MB of bandwidth. According to Youtube, this corresponds to 15 minutes of video at 1080p. Therefore, TELCO planed this experiment to identify changes of 15 minutes in TV consumption (which is a worst case scenario because the average Netflix show is likely longer than 15 minutes) and changes of 450 MB in download traffic, with a confidence level of 95% and with a power of 80%. Table 12 shows how many households would be needed in each stratum to obtain this level of power.

Table 12 Minimum sample size required to identify changes of 450MB in downloads per day and changes of 15 minutes of TV time per day with a 95% confidence level and 80% power, final sample size and number of treated households per stratum. Stratum statistics computed with data from April and May 2014

Stratum	Download (GB/Day)			TV (hours/Day)			Final Sample	
	Avg.	StDev.	Min Sample per treatment group	Avg.	StDev.	Min Sample per treatment group	All	Treated
NC,NP	1.2	2.0	311	4.4	2.4	1,329	6,107	3,077
NC,P	3.8	3.0	698	4.4	2.6	1,698	5,134	2,508
C,NP	2.2	2.4	447	4.4	2.9	2,113	4,307	2,153
C,P	4.4	6.1	2,885	4.4	2.5	1,570	5,918	2,963

To avoid running an underpowered experiment TELCO needed at least 3,057 treated households per stratum. In fact, and to account for potential practical problems that may arise during the

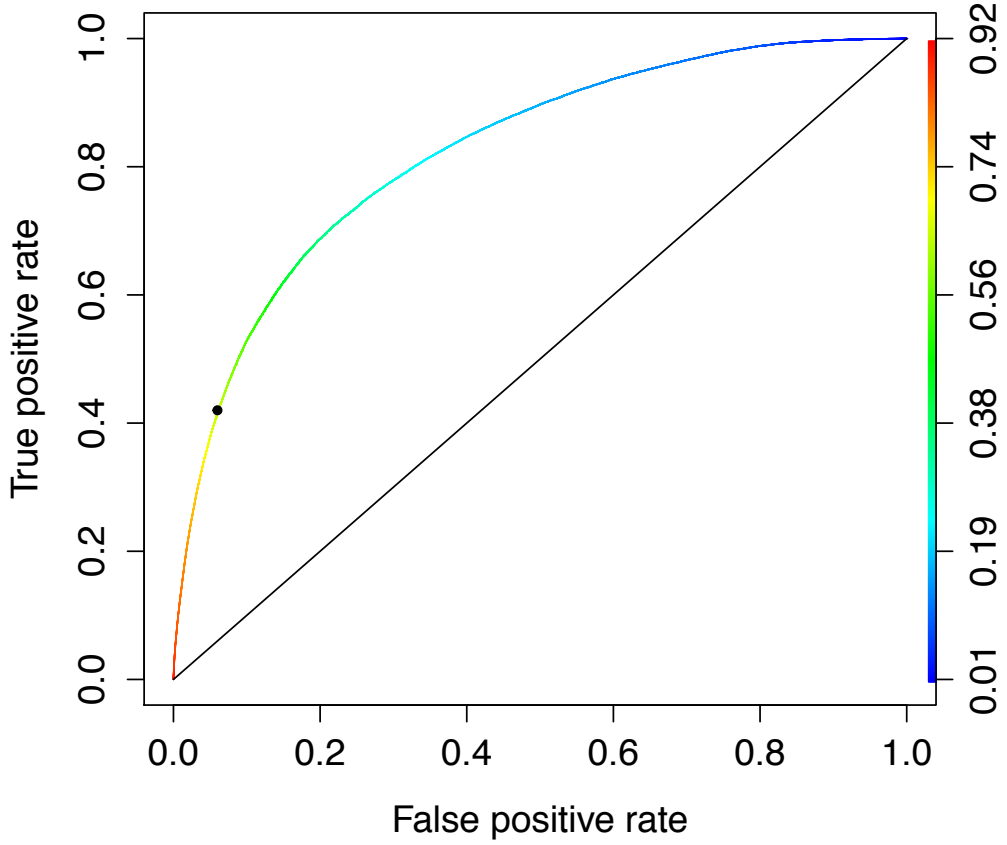


Figure 10 **ROC chart for gradient boosted model trees algorithm.**

experiment, TELCO randomly sampled 9,000 households in each stratum. Subsequently, treatment assignment within each stratum followed a simple randomized scheduled: half of TELCO's households in each stratum were randomly assigned to receive the free gift.

From the initial sample of 36,0000 households (9,000 per stratum), 7,590 households were removed from the because they had a legacy set-top-box equipment which can not be used to accurately track TV consumption. Another 3,270 households were removed from the sample because they opted out of marketing campaigns. 2,125 households were removed because they did not register a single day of TV or Internet usage during the experiment and another 1,549 households churned during the experiment.

A total of 21,466 households remained in the sample, distributed by strata as shown in the last two columns of Table 12.

The number of households in each strata was still well above the minimum threshold computed to identify the effect of interest.

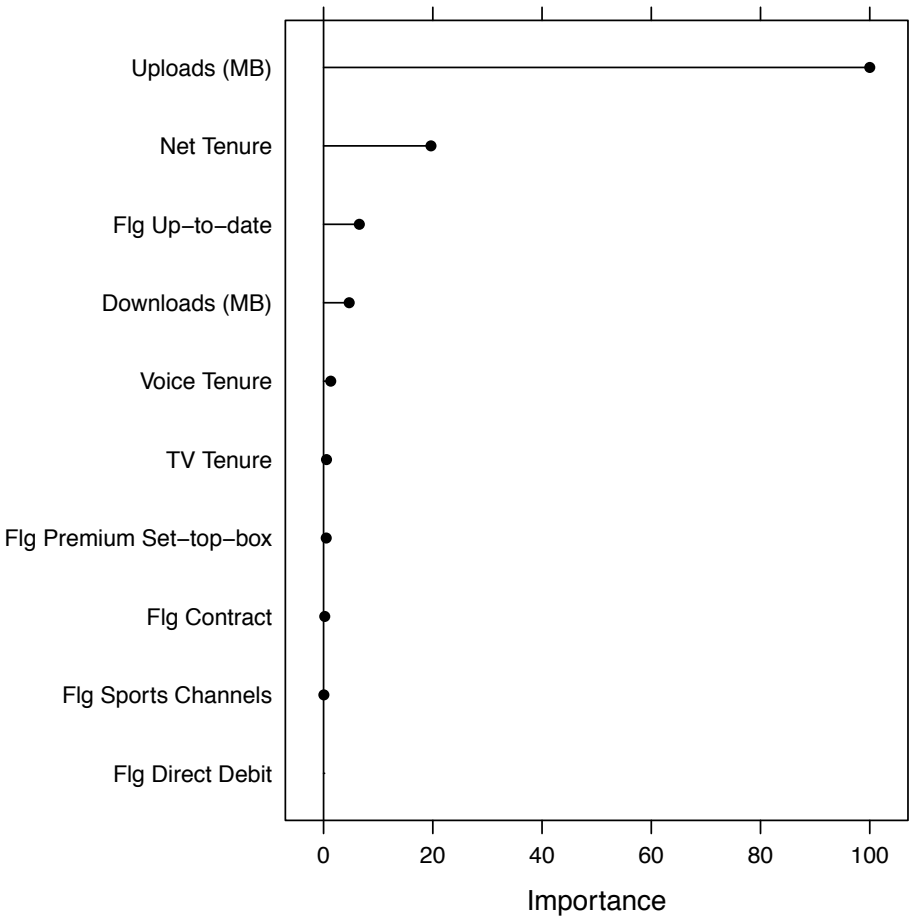


Figure 11 Variable importance plot for the gradient boosted model tree algorithm.

In this paper we focus only on the population of confirmed pirates that include the 10,225 households in strata C,NP and C,P.

Table 13 shows that the experimental design described above achieved good balance in key observed household characteristics across treatment and control households in all strata. Balance for each covariate is assessed using a T-test for the difference in means between treated and control households. In all cases, we cannot reject the null hypothesis that treated and control households are statistically similar at the 95% confidence level.

Table 13 Balance in observed household covariates across strata

	Treated		Control		T-test		
	Avg.	StDev	Avg.	StDev	Std. Effect	T-stat	p-value
NC,NP: Confirmed non-pirate, predicted non-pirate							
Pirate Score	0.181	0.135	0.183	0.135	-0.013	-0.492	0.623
TV tenure	86.878	60.930	87.588	60.957	-0.012	-0.454	0.650
Internet Tenure	56.289	35.253	57.165	36.007	-0.025	-0.959	0.338
Telephone tenure	49.037	21.923	49.151	21.792	-0.005	-0.204	0.839
Active Contract	0.824	0.381	0.818	0.386	0.016	0.633	0.527
Download (MB per day)	1,217.236	1,913.677	1,242.850	2,124.813	-0.013	-0.494	0.621
Upload (MB per day)	269.733	1,531.998	229.026	862.270	0.033	1.280	0.200
TV Channels zapped	11.676	7.940	11.518	8.036	0.020	0.773	0.439
CPTV	0.055	0.256	0.056	0.240	-0.003	-0.125	0.901
TV	4.393	2.403	4.474	2.476	-0.033	-1.289	0.198
VoD streams	0.927	1.074	0.943	1.113	-0.015	-0.591	0.555
NC,P: Confirmed non-pirate, predicted pirate							
Pirate Score	0.763	0.065	0.764	0.064	-0.003	-0.121	0.904
TV tenure	99.347	55.084	99.905	55.204	-0.010	-0.362	0.717
Internet Tenure	75.900	34.593	75.829	33.590	0.002	0.074	0.941
Telephone tenure	60.441	17.468	60.789	17.241	-0.020	-0.717	0.474
Active Contract	0.808	0.394	0.809	0.393	-0.004	-0.151	0.880
Download (MB per day)	3,770.558	4,935.325	3,738.642	5,185.037	0.006	0.225	0.822
Upload (MB per day)	2,331.399	5,367.296	2,229.316	4,445.092	0.021	0.739	0.460
TV Channels zapped	11.809	8.427	12.094	8.339	-0.034	-1.213	0.225
CPTV	0.061	0.270	0.074	0.312	-0.046	-1.652	0.099
TV	4.402	2.601	4.390	2.429	0.005	0.181	0.857
VoD streams	0.983	1.112	0.950	1.085	0.030	1.064	0.288
C,NP: Confirmed pirate, predicted non-pirate							
Pirate Score	0.304	0.151	0.311	0.150	-0.044	-1.456	0.145
TV tenure	93.136	57.657	91.215	55.605	0.034	1.111	0.267
Internet Tenure	66.586	34.231	66.391	34.349	0.006	0.186	0.852
Telephone tenure	54.911	18.944	55.335	18.639	-0.023	-0.740	0.459
Active Contract	0.802	0.398	0.793	0.405	0.024	0.775	0.439
Download (MB per day)	2,193.011	3,156.323	2,212.563	3,544.914	-0.006	-0.191	0.849
Upload (MB per day)	806.465	2,448.565	722.401	2,329.261	0.035	1.153	0.249
TV Channels zapped	11.640	8.179	11.948	7.965	-0.038	-1.249	0.212
CPTV	0.063	0.280	0.067	0.311	-0.014	-0.452	0.651
TV	4.375	2.505	4.464	2.512	-0.035	-1.150	0.250
VoD streams	1.020	1.180	1.010	1.108	0.009	0.286	0.775
C,P: Confirmed pirate, predicted pirate							
Pirate Score	0.757	0.080	0.757	0.082	0.001	0.036	0.971
TV tenure	96.826	56.475	94.636	56.055	0.039	1.495	0.135
Internet Tenure	71.509	33.653	70.442	33.810	0.032	1.214	0.225
Telephone tenure	58.385	16.955	57.846	17.373	0.031	1.206	0.228
Active Contract	0.796	0.403	0.803	0.397	-0.019	-0.716	0.474
Download (MB per day)	4,446.922	5,292.187	4,522.641	5,917.408	-0.013	-0.518	0.605
Upload (MB per day)	3,037.274	5,766.615	3,195.712	6,039.995	-0.027	-1.030	0.303
TV Channels zapped	11.900	8.621	11.622	8.056	0.033	1.279	0.201
CPTV	0.070	0.284	0.069	0.284	0.005	0.211	0.833
TV	4.362	2.495	4.381	2.526	-0.007	-0.285	0.776
VoD streams	0.967	1.108	0.998	1.105	-0.028	-1.057	0.290

Appendix B: IBCF Recommendation Technology

We adapted the Recommender Lab R package (Hahsler 2011) to implement our Item-Based Collaborative Filtering (IBCF) algorithm (Sarwar et al. 2001). Item-to-item collaborative filtering matches each of the target users downloads to similar content, called items. Then it combines those similar items into a recommendation list. To determine the most-similar match for a given item, the algorithm builds an item-similarity table by finding items that customers tend to purchase together.

We note that we only use this algorithm to recommend movies and thus the proxy for fit that it provides applies only to movies. Unfortunately, we are unable to apply this algorithm to TV series because our torrent logs do not have episode level information. This means that we are able to know users preferences across series, but are unable to recommend specific seasons or episodes they should watch.

Figures 12 and 13 show the standard out-of-sample performance metrics used to evaluate the top-N recommendations for each household. As benchmarks we compare the output of our algorithm to that of: (1) a non-personalized algorithm based on item popularity which recommends the titles that are most shared using BitTorrent by our sample of households; (2) a personalized algorithm that provides recommendations at random.

The performance of our personalized IBCF algorithm is in line with the best results that are achieved in datasets of comparable complexity. In particular, they are in-line with the performance of the algorithms reported in Cremonesi et al. (2010) when applied to the Netflix and Movielens datasets. The latter have been repeatedly used to benchmark the performance of recommendation technologies in several academic and industry competitions. Similar to Cremonesi et al. (2010) we find that the performance of the non-personalized algorithm on the top-N recommendations is comparable to the performance of more sophisticated, personalized algorithms. However, the non-personalized popularity-based algorithm does not suit the goal of our exercise because it provides only (trivial) recommendations that capture the preferences of the average household. This, however, does not inform us about the preferences of each particular household in our sample, which is what we need in order to construct a measure of fit between each household's preferences and the contents offered as part of the Cinema Pack during the 45 days that it was available to treated households.

Finally, we determine the overlap between the set of titles recommended to households using our IBCF algorithm and the set of titles available via the Cinema Pack. Figure 14 shows the distribution of this overlap for the case of the recommender system that suggests popular items (those most shared content using BitTorrent by households in our sample before the experiment

Figure 12 Precision and Recall of the Top-N recommendations generated by the models implemented.

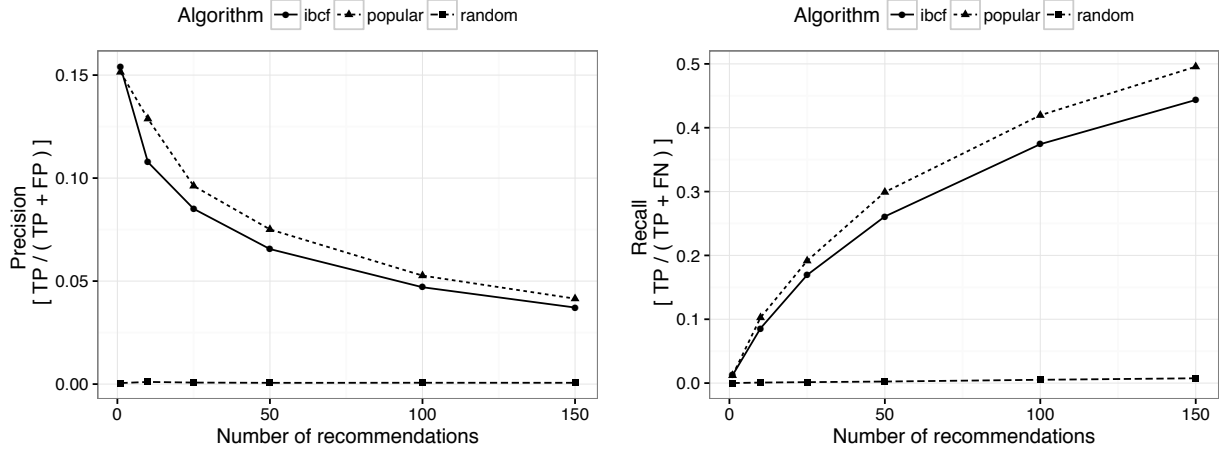
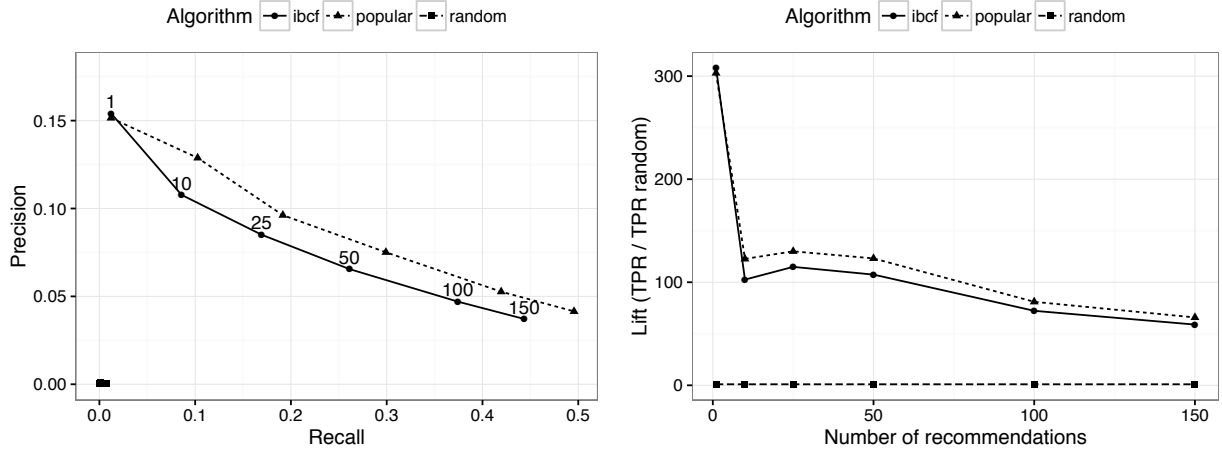


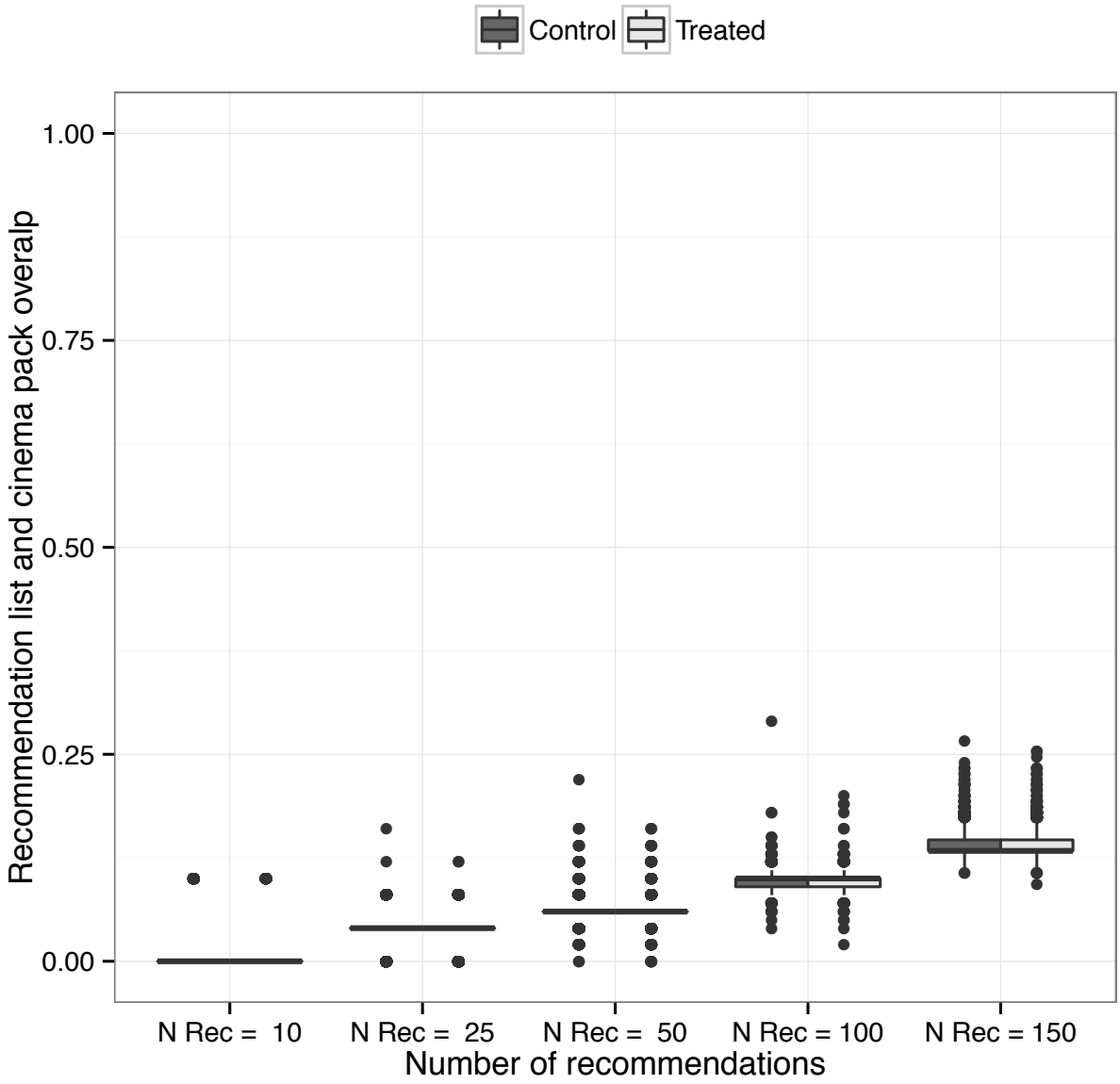
Figure 13 Left panel presents the Precision vs. Recall plot. The numbers on top of each point denote the size of the recommendation catalog issued. The right panel presents the Lift of the recommender systems implemented versus a random recommendations.



took place). The average overlap is 10% for the case of a list with 100 recommendations. This means that, on average across households in our sample, the cinema pack includes 10 titles out of the 100 recommended by the recommender system. This figure highlights the very low variation in the overlap across households. In fact, the existing variation is solely explained by the fact that our recommender system removes titles that households downloaded before.

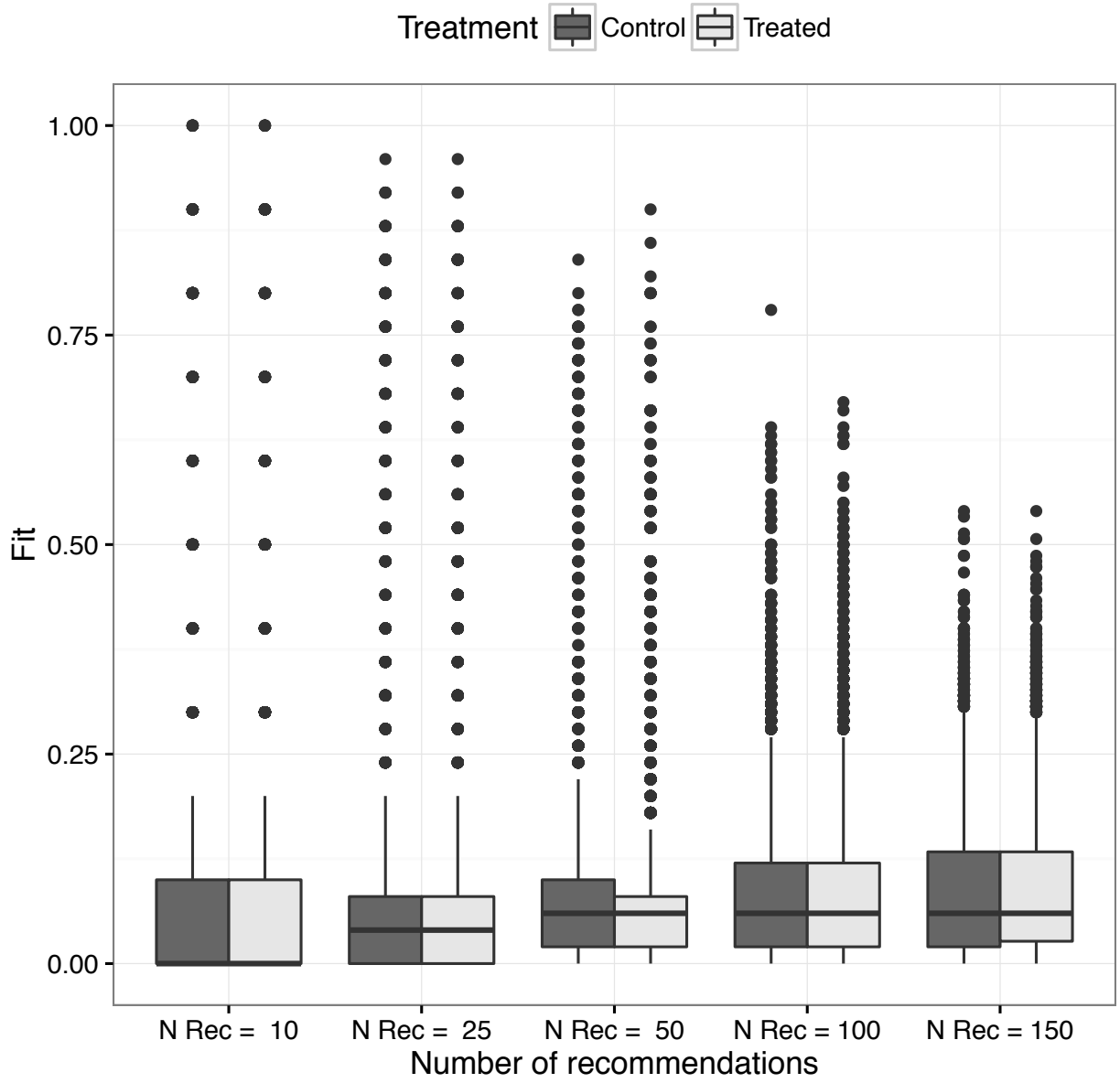
Figure 15 shows the distribution of the overlap for the case of our personalized IBCF recommender system. Similarly to the recommendation system based on content popularity, the average overlap is 10% for the case of a list with 100 recommendations. However, with the IBCF algorithm the range of the overlap is 0% to 78%, while with the non-personalized recommendation system the overlap never exceeds 25%. In short, the IBFC algorithm is able to recommend content to the

Figure 14 Overlap between the top-N popular recommendations and the content offered as part of the Cinema Pack.



tail of the distribution of preferences across households in our sample, while a popularity-based recommendation system does not.

Figure 15 **Overlap between the top-N IBCF recommendations and the content offered as part of the Cinema Pack.**



Appendix C: Local Average Treatment Effect

We define that a treated household complies with the treatment (and a control household does not) when the household uses the Cinema Pack for more than 90 minutes at least once during our experiment. Most movies in the Cinema Pack are 90 to 120 minutes long. Results using 20, 30, 60 and 120 minutes for the definition of compliance are qualitatively similar to those reported below and are available upon request. Figure 16 shows that across all strata in our sample, roughly 65% of the treated households used the cinema pack, compared to around 18% of the control households. Therefore, in our setting, using treatment assignment as an instrument for treatment compliance will yield the Local Average Treatment Effect (LATE) (Frangakis and Rubin 1999, Hollis and Campbell 1999). Table 14 shows the results obtained. In short, and in line with the main results in the paper, we find that the introduction of the Cinema Pack did not change the behavior of the average household in our sample, but that households whose preferences align well with the content offered as part of the cinema pack reduced their likelihood of using BitTorrent during the experiment. As expected, the magnitude of the effects reported in this table is larger than those reported in the main paper because the effect of the Intention-To-Treat averages out compliers and non-compliers. In this table, we report the effect for the sub-population of households in our sample that indeed changed their behavior due to using the Cinema Pack. We observe that among these households those whose preferences fit 100% with the content offered as part of the Cinema Pack reduce their likelihood of using BitTorrent during the experiment by more than 33%.

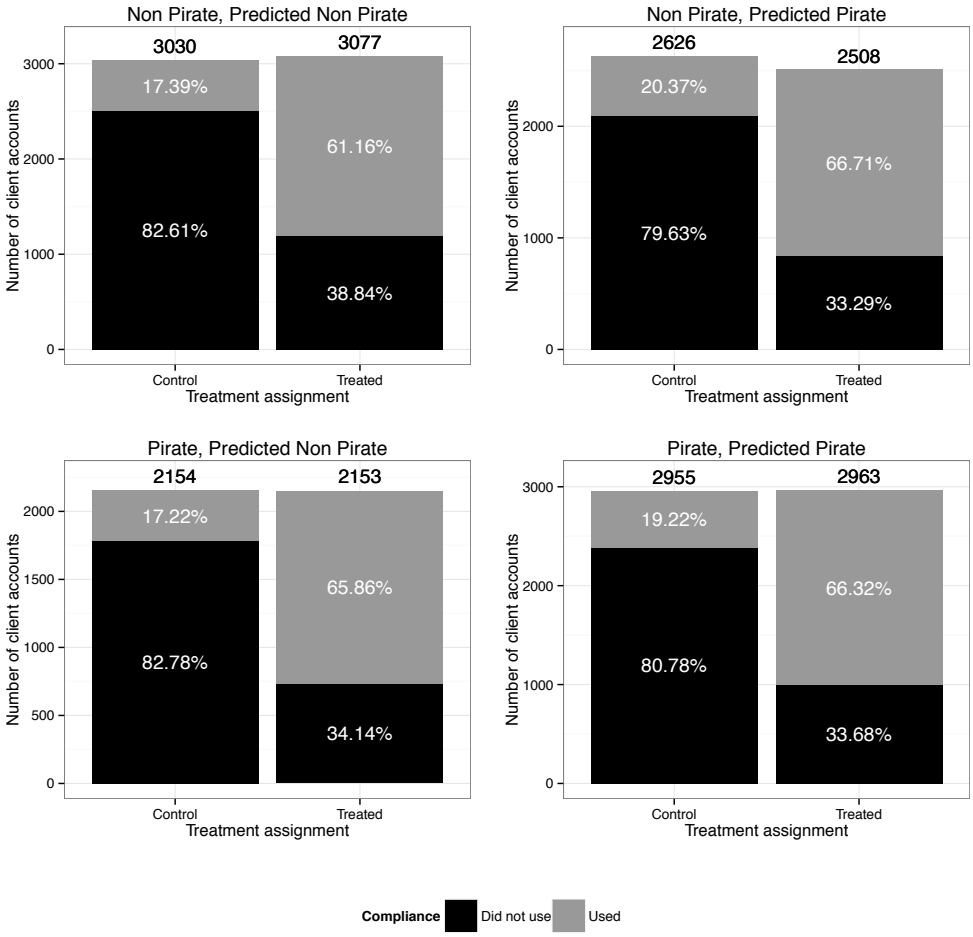


Figure 16 Compliance with treatment assignment in each stratum.

Table 14 The Local Average Treatment Effect (LATE) on BitTorrent usage.

	Dependent variable:					
	Flg. Torrent			Flg. Movie Torrent		
	2SLS			*2SLS*		
	50 Recs.	100 Recs.	150 Recs.	50 Recs.	100 Recs.	150 Recs.
	(1)	(2)	(3)	(4)	(5)	(6)
Used	0.016	0.016	0.017	0.008	0.016	0.022
	(0.019)	(0.021)	(0.022)	(0.021)	(0.022)	(0.023)
Used * Offer Fit	−0.397**	−0.352*	−0.363*	−0.334*	−0.398**	−0.476**
	(0.179)	(0.183)	(0.186)	(0.182)	(0.201)	(0.210)
Offer Fit	−0.154*	0.038	0.220**	0.023	0.467***	0.885***
	(0.087)	(0.090)	(0.092)	(0.091)	(0.103)	(0.108)
Flg. No Recs	−0.213***	−0.195***	−0.179***	−0.295***	−0.257***	−0.224***
	(0.011)	(0.012)	(0.012)	(0.011)	(0.011)	(0.011)
Log(BExp. TV Time)	0.034***	0.033***	0.033***	0.049***	0.047***	0.047***
	(0.006)	(0.006)	(0.006)	(0.007)	(0.007)	(0.007)
Log(BExp. Download)	0.002	0.004	0.005	−0.029***	−0.026***	−0.023***
	(0.005)	(0.005)	(0.005)	(0.005)	(0.005)	(0.005)
Log(BExp. Upload)	0.070***	0.071***	0.070***	0.081***	0.081***	0.078***
	(0.003)	(0.003)	(0.003)	(0.003)	(0.003)	(0.003)
BExp. Torrents	0.002**	0.003**	0.003**	0.016***	0.017***	0.017***
	(0.001)	(0.001)	(0.001)	(0.004)	(0.004)	(0.004)
Constant	0.327***	0.294***	0.270***	0.173***	0.111***	0.066**
	(0.030)	(0.031)	(0.031)	(0.031)	(0.031)	(0.031)
Observations	10,225	10,225	10,225	10,225	10,225	10,225
R^2	0.180	0.175	0.175	0.190	0.192	0.200
Adjusted R^2	0.179	0.175	0.175	0.190	0.191	0.200
Residual Std. Error	0.393	0.394	0.394	0.447	0.447	0.444

Note: *p<0.1; **p<0.05; ***p<0.01
Analysis pertains to the period during the experiment
Robust standard errors in ()

References

Assmann, Susan F, Stuart J Pocock, Laura E Enos, Linda E Kasten. 2000. Subgroup analysis and other (mis) uses of baseline data in clinical trials. *The Lancet* **355**(9209) 1064–1069.

Breiman, Leo. 2001. Random forests. *Machine learning* **45**(1) 5–32.

Breiman, Leo, Jerome Friedman, Charles J Stone, Richard A Olshen. 1984. *Classification and regression trees*. Chapman and Hall.

Cremonesi, Paolo, Yehuda Koren, Roberto Turrin. 2010. Performance of recommender algorithms on top-n recommendation tasks. *Proceedings of the fourth ACM conference on Recommender systems*. ACM, 39–46.

Frangakis, Constantine E, Donald B Rubin. 1999. Addressing complications of intention-to-treat analysis in the combined presence of all-or-none treatment-noncompliance and subsequent missing outcomes. *Biometrika* **86**(2) 365–379.

Friedman, Jerome H. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics* 1189–1232.

Hahsler, Michael. 2011. recommenderlab: A framework for developing and testing recommendation algorithms. *Working Paper* URL `https://cran.r-project.org/web/packages/recommenderlab/recommenderlab.pdf`.

Hearst, Marti A., Susan T Dumais, Edgar Osman, John Platt, Bernhard Scholkopf. 1998. Support vector machines. *Intelligent Systems and their Applications, IEEE* **13**(4) 18–28.

Hollis, Sally, Fiona Campbell. 1999. What is meant by intention to treat analysis? survey of published randomised controlled trials. *BMJ* **319**(7211) 670–674. doi:10.1136/bmj.319.7211.670.

Kuhn, Max. 2008. Building predictive models in r using the caret package. *Journal of Statistical Software* **28**(5) 1–26.

Sarwar, Badrul, George Karypis, Joseph Konstan, John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. *Proceedings of the 10th international conference on World Wide Web*. ACM, 285–295.

Simon, Richard. 1979. Restricted randomization designs in clinical trials. *Biometrics* 503–512.

Suykens, Johan AK, Joos Vandewalle. 1999. Least squares support vector machine classifiers. *Neural processing letters* **9**(3) 293–300.

Swets, John A, et al. 1988. Measuring the accuracy of diagnostic systems. *Science* **240**(4857) 1285–1293.

Tibshirani, Robert. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* 267–288.