IEOR8100: Economics, AI, and Optimization
Lecture Note 14: Large-Scale Fisher Market Equilibrium

Christian Kroer*

April 21, 2020

1 Introduction

We saw that market equilibrium comes up in Internet scale settings such as fair recommender systems and budget-smoothed auctions (via pacing equilibrium). In this lecture note we will look at methods for computing market equilibrium at scale. In particular, we will consider two complementary approaches: 1) how to run fast iterative methods in order to compute a market equilibrium, and 2) how to abstract the market, either down to a manageable size, or in order to deal with incomplete valuations.

2 Setup Recap

As in previous lecture notes, we study Fisher markets: we have a set of m infinitely-divisible goods that we wish to divide among n buyers. Without loss of generality we assume that each good has supply 1. We will denote the bundle of goods given to buyer i as x_i, where x_{ij} is the amount of good j that is allocated to buyer i. We shall use x to denote an assignment of goods to buyers. Each buyer is endowed with a budget B_i of currency.

Each buyer is assumed to have a linear utility function $u_i(x_i) = \langle v_i, x_i \rangle$ denoting how much they like the bundle x_i. The results in this lecture note all carry over to quasi-linear utilities $u_i(x_i, p) = \langle v_i - p, x_i \rangle$ unless otherwise noted. Since we will be solving the Eisenberg-Gale convex program, the quasi-linear results also carry over to computing a first-price pacing equilibrium.

As mentioned in a prior note, a *market equilibrium* is a set of prices $p \in \mathbb{R}^m_+$ for each of the m goods, as well as an allocation x of goods to buyers such that everybody is assigned an optimal allocation given the prices and their budget. Formally, the *demand set* of an buyer i with budget B_i is

$$D(p) = \text{argmax}_{x_i \geq 0} u_i(x_i) \text{ s.t. } \langle p, x_i \rangle \leq B_i$$

A market equilibrium is an allocation-price pair (x, p) s.t. $x_i \in D(p)$ for all buyers i, and $\sum_i x_{ij} = 1$.

3 Interlude on Convex Conjugates

Given a function $f : \mathbb{R}^n \to \mathbb{R}$ we say that its *convex conjugate* is the function

$$f^*(y) = \sup_x \langle y, x \rangle - f(x)$$

*Department of Industrial Engineering and Operations Research, Columbia University. Email: christian.kroer@columbia.edu.

We will be interested in the convex conjugate of the function $f(x) = -\log x$. We get

$$f^*(y) = \sup_x \; yx + \log x$$

and using first-order optimality we get $x^* = -1/y$, so we get that for $y < 0$

$$f^*(y) = -1 + \log(-1/y) = -1 - \log(-y) \tag{1}$$

4 Duals of the Eisenberg-Gale Convex Program

In a previous lecture we saw that the following convex program, which we called the *Eisenberg-Gale convex program* (EG) yields a market equilibrium for Fisher markets with linear utilities:

$$\begin{array}{llll} \max\limits_{x \geq 0} & \sum\limits_i B_i \log u_i & & \text{Dual variables} \\ s.t. & u_i \leq \langle v_i, x_i \rangle, & \forall i = 1, \ldots, n, & \beta_i \\ & \sum\limits_i x_{ij} \leq 1, & \forall j = 1, \ldots, m, & p_j \end{array} \tag{EG}$$

Remember that $x_i \in \mathbb{R}^m$ is the allocation for buyer i, and u_i is the utility.

We will now show how to derive the dual of this convex, and eventually use a further duality step to derive an interesting and very practical algorithm for solving EG. We introduce dual variables β_i (corresponding to the utility price of buyer i), and p_j (the price of item j). The dual variables are listed on the right of their corresponding primal constraint in EG. We construct the Lagrangian

$$L(x, \beta, p) = \sum_i B_i \log u_i + \sum_i \beta_i(\langle v_i, x_i \rangle - u_i) + \sum_j p_j(1 - \sum_i x_{ij})$$

The standard Lagrangian dual is then

$$\min_{p \geq 0, \beta \geq 0} \max_{x \geq 0} \; L(x, \beta, p) \tag{2}$$

Now, we simplify the inner max:

$$\begin{aligned} \max_{x \geq 0} \; L(x, \beta, p) &= \sum_j p_j + \sum_i \left[\max_{u_i} \left(B_i \log u_i - \beta_i u_i \right) + \max_{x_i \geq 0} \langle \beta_i v_i - p, x_i \rangle \right] \\ &= \sum_j p_j + \sum_i \left[\max_{u_i} \left(B_i \log u_i - \beta_i u_i \right) + \delta \left[\beta_i v_i \leq p \right] \right] \\ &= \sum_j p_j + \sum_i \left[B_i \max_{u_i} \left(\log u_i - \frac{\beta_i}{B_i} u_i \right) + \delta \left[\beta_i v_i \leq p \right] \right] \\ &= \sum_j p_j + \sum_i \left[B_i \left(-1 - \log \beta_i + \log B_i \right) + \delta \left[\beta_i v_i \leq p \right] \right] \end{aligned}$$

The first equality is by rearranging terms. The second equality is by noting that the max over $x_i \geq 0$ is positive infinity if $\beta_i v_{ij} > p_j$ for any j. The third equality is by rearranging B_i. The fourth equality is by (1).

Thus we get that the dual (2) is equal to

$$\begin{aligned} \min_{p\geq 0,\beta\geq 0} \quad & \sum_j p_j - \sum_i B_i \log(\beta_i) + \sum_i (\log B_i - B_i) \\ & p_j \geq v_{ij}\beta_i, \quad \forall i,j \end{aligned} \tag{3}$$

Finally we may drop the terms $\sum_i (\log B_i - B_i)$ since they are constant, which finally yields the standard dual of EG:

$$\begin{aligned} \min_{p\geq 0,\beta\geq 0} \quad & \sum_j p_j - \sum_i B_i \log(\beta_i) \\ & p_j \geq v_{ij}\beta_i, \quad \forall i,j \end{aligned} \tag{4}$$

4.1 Shmyrev's Convex Program

Now we introduce a change of variables to (4), by letting $q_j = \log p_j$ and $\gamma_i = -\log \beta_i$. Plugging these definitions into (4) we get

$$\begin{aligned} \min_{q,\gamma} \quad & \sum_j e^{q_j} + \sum_i B_i\gamma_i \\ & q_j + \gamma_i \geq \log v_{ij}, \quad \forall i,j \end{aligned} \tag{5}$$

Now we introduce Lagrangian variables b_{ij} for the constraint in (5) to get the following dual:

$$\begin{aligned} &\max_{b\geq 0} \min_{q,\gamma} \sum_j e^{q_j} + \sum_i B_i\gamma_i + \sum_{ij} b_{ij} \left[\log v_{ij} - q_j - \gamma_i\right] \\ &= \max_{b\geq 0} \left[\sum_{ij} b_{ij} \log v_{ij} + \sum_j \min_{q_j} \left[e^{q_j} - \sum_i b_{ij} q_j \right] + \sum_i \min_{\gamma_i} \left[B_i - \sum_j b_{ij} \right] \right] \end{aligned}$$

Now first-order optimality on γ_i shows $B_i = \sum_j b_{ij}$ and first-order optimality on q_j shows $e^{q_j} = \sum_i b_{ij}$. In a slight abuse of notation, we will introduce a dual variable $p_j = e^{q_j}$. Putting this together we get *Shmyrev's convex program*:

$$\begin{aligned} \max_{b\geq 0} \quad & \sum_i b_{ij} \log v_{ij} + \sum_j (p_j - p_j \log p_j) \\ s.t. \quad & \sum_i b_{ij} = p_j, \quad \forall j = 1,\ldots,m, \\ & \sum_j b_{ij} = B_i, \quad \forall i = 1,\ldots,n, \end{aligned} \tag{6}$$

Since $\sum_j p_j = \sum_i B_i$, which is a constant, we may rewrite Shmyrev's CP as

$$\begin{aligned} \max_{b\geq 0} \quad & \sum_i b_{ij} \log v_{ij} - \sum_j p_j \log p_j \\ s.t. \quad & \sum_i b_{ij} = p_j, \quad \forall j = 1,\ldots,m, \\ & \sum_j b_{ij} = B_i, \quad \forall i = 1,\ldots,n, \end{aligned} \tag{Shmyrev}$$

5 First-Order Methods

We will now apply online mirror descent (OMD) to (Shmyrev). Remember that OMD makes updates according to the rule:

$$x_{t+1} = \operatorname*{argmin}_{x \in X} \langle \eta \nabla f_t(x), x \rangle + D(x \| x_t).$$

where $\eta > 0$ is the stepsize and $D(x\|x_t)$ is the Bregman divergence between x and x_t.

In order to instantiate OMD, we first rewrite (Shmyrev) in terms of b_{ij} only (letting $p_j(b) = \sum_i b_{ij}$) to get the objective function

$$f(b) = -\sum_{ij} b_{ij} \log v_{ij} + \sum_j p_j(b) \log p_j(b) = -\sum_{ij} b_{ij} \log(v_{ij}/p_j(b)).$$

The feasible set is

$$X = \left\{ b \in \mathbb{R}_+^{n\times m} | \sum_j b_{ij} = B_i, \forall i \right\}.$$

Finally, we use the distance function $d(b) = \sum_{ij} b_{ij} \log b_{ij}$ which gives $D(b\|a) = \sum_{ij} b_{ij} \log(b_{ij}/a_{ij})$

At each time t, we simply see the loss $f(b^t)$. The gradient is $\nabla_{ij} f(b) = 1 - \log(v_{ij}/p_j(b))$. Similar to when using the negative entropy on the simplex, the OMD update becomes (setting $\eta = 1$):

$$\begin{aligned} b_{ij}^{t+1} &\propto b_{ij}^t \exp(-1 + \log(v_{ij}/p_j(b))) \\ &\propto b_{ij}^t \, (v_{ij}/p_j(b)) \\ &= \frac{1}{Z} b_{ij}^t \, (v_{ij}/p_j(b)) \end{aligned}$$

where Z is a normalization constant such that $\sum_j b_{ij}^{t+1} = B_i$.

Amazingly, OMD on (Shmyrev) using a stepsize of 1 becomes the following very natural algorithm:

- At each time t, each buyer i submits a bid vector b_i^t (the current OMD recommendation)
- Given the bids, a price $p_j^t = \sum_i b_{ij}^t$ is computed for each item
- Each buyer is given $x_{ij}^t = \frac{b_{ij}^t}{p_j^t}$ of each item
- Each buyer submits their next bid on item j proportional to the utility they received from item j in round t:

$$b_{ij}^{t+1} = B_i \frac{x_{ij}^t v_{ij}}{\sum_{j'} x_{ij'}^t v_{ij'}}$$

It remains to discuss the fact that we set $\eta = 1$. In past lecture notes we saw that the uniform average of OMD iterates converges to zero average regret at a rate of $O(1/\sqrt{T})$, when using a stepsize proportional to the inverse of the largest observed dual norm of gradients. However, our objective f does not admit such a bound: the gradient for i, j goes to infinity as $p_j(b)$ tends to zero. Thus based on our existing framework for OMD we are not even guaranteed a bound on regret.

However, it turns out that one can show the following "1-Lipschitz" condition relative to D:

Lemma 1. *For all $a, b \in S$,*

$$f(b) \leq f(a) + \langle \nabla f(a), b - a \rangle + D(b\|a), \quad \forall b, a \in X.$$

This inequality is a sort of generalized Lipschitz condition where we replace the ℓ_2 norm $\|a - b\|_2^2$ that is typically used with our Bregman divergence D (this is analogous to how OMD itself generalized projected gradient descent by changing the distance function).

To show this inequality, we will need the fact that the Bregman divergence $D(b\|a)$ is convex in both arguments for $b, a \in \mathbb{R}_{++}^{n \times m}$. To see that convexity holds, one can expand $D(b\|a) = \sum_{ij} b_{ij} \log(b_{ij}/a_{ij})$ and note that taking a sum preserves convexity. At that point, we only need to check convexity of the function $h(t, x) = t \log(t/x) = -t \log(x/t)$, which is simply the perspective of $-\log(x)$ with respect to t. Taking perspectives is known to preserve convexity, and the negative log is of course convex.

Proof. The proof of the inequality can be split into two parts. First, it can be observed that the difference between $f(b)$ and its linearization at a is the Bregman divergence $D(p(b)\|p(a))$:

$$\begin{aligned}
&f(b) - f(a) - \langle \nabla f(a), b - a \rangle \\
=&-\sum_{ij} b_{ij} \log(v_{ij}/p_j(b)) + \sum_{ij} a_{ij} \log(v_{ij}/p_j(a)) - \sum_{ij} \left(1 - \log(v_{ij}/p_j(a))\right)(b_{ij} - a_{ij}) \\
=&-\sum_{ij} b_{ij} \log(v_{ij}/p_j(b)) + \sum_{ij} b_{ij} \log(v_{ij}/p_j(a)) - \sum_{ij} (b_{ij} - a_{ij}) \\
=&\sum_{ij} b_{ij} \log(p_j(b)/p_j(a)) - \sum_{ij} (b_{ij} - a_{ij}) \\
=&\sum_{ij} b_{ij} \log(p_j(b)/p_j(a)) \quad ; \text{since } \|a\|_1 = \|b\|_1 = \sum_i B_i \\
=&\sum_j p_j(b) \log(p_j(b)/p_j(a)) \quad ; \text{since } p_j(b) = \sum_i b_{ij} \\
=&D(p(b)\|p(a))
\end{aligned}$$

Secondly, we can bound $D(p(b)\|p(a))$ as follows (where $h(t, x) = t \log(t/x)$)

$$\begin{aligned}
D(p(b)\|p(a)) =& n \sum_j \frac{1}{n} h(p_j(b), p_j(a)) \\
=& n \sum_j h\left(\frac{1}{n} p_j(b), \frac{1}{n} p_j(a)\right) \\
\leq& n \sum_j \frac{1}{n} \sum_i h\left(b_{ij}, a_{ij}\right) \\
=& D(b\|a)
\end{aligned}$$

Putting together the two bounds we get Lemma 1. □

Using the Lipschitz-like condition on f, one can show a stronger statement when running OMD on a static objective f (which means that it is the same as running normal mirror descent):

Theorem 1. *The OMD iterates with $\eta = 1$ converge at the rate:*

$$f(b^t) - f(b^*) \leq \frac{\log nm}{t}.$$

This holds for any convex and differentiable f and D satisfying 1

Note two very nice properties here: the convergence rate is improved by a factor of $\sqrt{t}$, and the iterates themselves converge, with no need for averaging. We won't prove the above theorem here, but it holds for any convex minimization problem that satisfies the relative Lipschitz condition in Lemma 1.

6 Abstraction Methods

So far we have described a scalable first-order method for computing market equilibrium. Still, this algorithms makes a number of assumptions that may not hold in practice. First, the size of an iterate b^t is nm; if both are on the order of 100,000 then writing down an iterate using 64-bit floats requires about 80 GB of memory. For an application such as an Internet advertising market we might expect n, and especially m, to be even larger than that. Thus we may need to find a way to abstract that market down to some manageable size where we can at least hope to write down iterates. Secondly, in practice we may not have access to all v_{ij}. Instead, we may only have samples from v_{ij}, and we need to somehow infer the remaining valuations.

We now move to considering abstraction methods, which will allow us to deal with both of the above issues.

For the purposes of abstraction, it will be useful to think of the set of valuations v_{ij} as a matrix V, where the i'th row corresponds to the valuation vector of buyer i. We will be interested in what happens if we compute a market equilibrium using some valuation matrix $\tilde{V} \neq V$, where $\tilde{V}$ would typically be obtained from some abstraction method. Can we say anything about how "close" to market equilibrium we are in terms of the original V, for example if $\|\tilde{V} - \|_F$ is small?

We first describe two reasons that we might compute a market equilibrium for $\tilde{V}$ rather than V:

1. *Low-rank markets*: When there are missing valuations, we need to somehow impute the missing values. Of course, if there is no relationship between the entries of V that we observed, and those that are missing, then we have no hope of recovering V. However, in practice this is typically not the case. In practice, the valuations are often assumed to (approximately) belong to some low-dimensional space. A popular model is to assume that the valuations are *low rank*, meaning that every buyer i has some d-dimensional vector ϕ_i, every good j has some d-dimensional vector ψ_j, and the valuation of buyer i for good j is $\tilde{v}_{ij} = \langle \phi_i, \psi_j \rangle$. One may interpret this model as every item having some associated set of d *features*, with ψ_j describing the value for each feature, and ϕ_i describes the value that i places on each feature. In a low-rank model d is expected to be much smaller than $\min(n, m)$, meaning that V is far from full rank. If the real valuations are approximately rank d (meaning that the remaining spectrum of V is very small), then $\tilde{V}$ will be close to V.

   This model can also be motivated via the singular-value decomposition (SVD). Assume that we wish to solve the following problem:

$$\min_{\tilde{V}} \sum_{ij} (v_{ij} - \tilde{v}_{ij})^2 = \|V - \tilde{V}\|_F^2$$
$$s.t.\ \mathrm{rank}(\tilde{V}) \leq d$$

The optimal solution to this problem can be found easily via the SVD: Letting $\sigma_1, \dots, \sigma_d$ be the first d singular values of V, and $\bar{u}_1, \dots, \bar{u}_d$ the first left singular vectors, and $\bar{v}_1, \dots, \bar{v}_d$ the first right singular vectors, the optimal solution is

$$\tilde{V} = \sum_{k=1}^{d} \sigma_k \bar{u}_k \bar{v}_k^T.$$

If the remaining singular values $\sigma_{k+1}, \dots$ are small relative to the first k singular values, then this model captures most of the valuation structure.

In practice we don't know V, and so we can't solve this mathematical program to get $\tilde{V}$. Instead, we search for a low-rank model that minimizes some loss on the observed entries, e.g. $\sum_{ij\in\Omega}(v_{ij} - \langle \phi_i, \psi_j \rangle)^2$ (in practice this objective is typically also regularized by the Frobenius norm of the low-rank matrices). Under the assumption that V is generated from a true low-rank model via some simple distribution, it is possible to recover the original matrix with only samples of entries by minimizing the loss on observed entries. In practice this approach is also known to perform extremely well, and it is used extensively at major Internet companies (the hypothesis here would be that in practice the data is approximately low rank, so we don't lose much accuracy from a rank-d model).

2. *Representative Markets*: We may wish to try to generate a smaller set of representative buyers, where each original buyer i maps to some particular representative buyer $r(i)$. Similarly, we may wish to generate representative goods that correspond to many non-identical but similar goods from the original market. In practice these representative buyers and goods would typically be generated via clustering techniques. In this case, our approximate valuation matrix $\tilde{V}$ has as row i the valuation vector of the representative buyer $r(i)$. This means that all i, i' such that $r(i) = r(i')$ have the same valuation vector in $\tilde{V}$, and thus they can be treated as a single buyer for equilibrium-computation purposes. The same grouping can also be applied to the goods. If the number of buyers and goods is reduced by a factor of 10, then the resulting mathematical program is reduced by a factor of 10^2, since we have $n \times m$ variables.

6.1 Measuring Solution Quality

We now analyze what happens when we compute a market equilibrium under $\tilde{V}$ rather than V. Throughout this section we will let $(\tilde{x}, \tilde{p})$ be a market equilibrium for $\tilde{V}$. We will use the error matrix $\Delta V = V - \tilde{V}$ to quantify the solution quality, and we will measure the size of ΔV using the $\ell_1 - \ell_\infty$ matrix norm:

$$\|\Delta V\|_{1,\infty} = \max_i \|\Delta v_i\|_1.$$

We will also use the norm of the error vector for an individual buyer $\|\Delta v_i\|_1 = \|v_i - \tilde{v}_i\|_1$.

A very useful property is that under linear utilities, the change in utility when going from v_i to $\tilde{v}_i$ is linear in Δv_i.

Proposition 1. *If* $\langle \tilde{v}_i, x_i \rangle + \epsilon \geq \langle \tilde{v}_i, x_i' \rangle$ *then* $\langle v_i, x_i \rangle + \epsilon + \|\Delta v_i\|_1 \geq \langle v_i, x_i' \rangle$

Proof. We have

$$\begin{aligned} &\langle \tilde{v}_i, x_i \rangle + \epsilon \geq \langle \tilde{v}_i, x_i' \rangle \\ \Leftrightarrow\ &\langle v_i - \Delta v_i, x_i \rangle + \epsilon \geq \langle v_i + \Delta v_i, x_i' \rangle \\ \Leftrightarrow\ &\langle v_i, x_i \rangle + \langle \Delta v_i, x_i' - x_i \rangle + \epsilon \geq \langle v_i, x_i' \rangle \end{aligned}$$

Now the proposition follows by $\langle \Delta v_i, x'_i - x_i \rangle \leq \|\Delta v_i\|_1$. □

This proposition can be used to immediately derive bounds on envy, proportionality, and regret (how far each buyer is from achieving the utility of their demand bundle). For example, we know that under $\tilde{V}$, each buyer i has no envy towards any other buyer k: $\langle \tilde{v}_i, \tilde{x}_i \rangle \geq \langle \tilde{v}_i, \tilde{x}_k \rangle$. By Proposition 1 each buyer i has envy at most $\|\Delta v_i\|_1$ under V when using $(\tilde{x}, \tilde{p})$. All envies are thus bounded by $\|\Delta V\|_{1,\infty}$. Regret and proportionality is bounded similarly using guaranteed inequalities under $\tilde{V}$.

Market equilibrium also guarantees Pareto optimality. Can we give any meaningful guarantees on how much social welfare improves under Pareto-improving allocati8ons for $\tilde{V}$? Unfortunately the answer to that is no, as the following example of real and abstracted matrices shows:

$$V = \begin{bmatrix} 1 & \epsilon & \epsilon \\ 0 & 1 & \epsilon \end{bmatrix}, \quad \tilde{V} = \begin{bmatrix} 1 & \epsilon & 0 \\ 0 & 1 & \epsilon \end{bmatrix}.$$

If we set $B_1 = B_2 = 1$, then for supply-aware market equilibrium, we end up with competition only on item 2, and we get prices $\tilde{p} = (0, 2, 0)$ and allocation $\tilde{x}_1 = (1, 0.5, 0), \tilde{x}_2 = (0, 0.5, 1)$. Under V this is a terrible allocation, and we can Pareto improve by using $x_1 = (1, 0, 0.5), x_2 = (0, 1, 0.5)$, which increases overall social welfare by $\frac{1}{2} - \epsilon$, in spite of $\|\Delta V\|_1 = \epsilon$.

On the other hand, we can show that under any Pareto-improving allocation, some buyer i improves by at most $\|\Delta V\|_{1,\infty}1$. To see this, note that for any Pareto improving allocation x, under $\tilde{V}$ there existed at least one buyer i such that $\langle \tilde{v}_i, \tilde{x}_i - x_i \rangle \geq 0$, and so this buyer must improve by at most $\|\Delta v_i\|_1$ under V.

7 Historical Notes

The Shmyrev CP was given by Shmyrev [10]. The observation that the Shmyrev CP is related to EG via duality and change of variables was by Cole et al. [5]. The original proportional response dynamics were given by Wu and Zhang [13], and was shown to be effective for BitTorrent sharing dynamics by Levin et al. [8]. The relationship of PR dynamics to Shmyrev's CP and mirror descent were given by Birnbaum et al. [1]. For rules on convexity-preserving operations, see Boyd and Vandenberghe [2].

There is a long history of algorithms for computing market equilibrium in various Fisher-market models. In this lecture note we focused on a particular method that is, to the best of our knowledge, one of the fastest simple and scalable first-order methods for computing a market equilibrium.

The material on abstracting large market equilibrium problems is from Kroer et al. [7].

A brief introduction to low-rank models can be found in Udell [11]. Udell et al. [12] gives a more through exposition and describes more general model types. There's also a fascinating theory of low-rank models, where a number of cool results are known: there's a class of nuclear-norm-regularized convex optimization problems that can recover the original matrix with only a small number of entry samples [3, 9]. One might think that this would then be the preferred method in practice, but surprisingly non-convex models are often preferred instead. These non-convex methods also have interesting guarantees on statistical recovery under certain assumptions. An overview of non-convex methods is given in Chi et al. [4].

Low-rank market equilibrium models were also studied in Kroer and Peysakhovich [6], where it is shown that large low-rank markets enjoy a number of properties not satisfied by small-scale markets.

References

[1] Benjamin Birnbaum, Nikhil R Devanur, and Lin Xiao. Distributed algorithms via gradient descent for fisher markets. In *Proceedings of the 12th ACM conference on Electronic commerce*, pages 127–136. ACM, 2011.

[2] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[3] Emmanuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717, 2009.

[4] Yuejie Chi, Yue M Lu, and Yuxin Chen. Nonconvex optimization meets low-rank matrix factorization: An overview. *IEEE Transactions on Signal Processing*, 67(20):5239–5269, 2019.

[5] Richard Cole, Nikhil R Devanur, Vasilis Gkatzelis, Kamal Jain, Tung Mai, Vijay V Vazirani, and Sadra Yazdanbod. Convex program duality, fisher markets, and Nash social welfare. In *18th ACM Conference on Economics and Computation, EC 2017*. Association for Computing Machinery, Inc, 2017.

[6] Christian Kroer and Alexander Peysakhovich. Scalable fair division for'at most one'preferences. *arXiv preprint arXiv:1909.10925*, 2019.

[7] Christian Kroer, Alexander Peysakhovich, Eric Sodomka, and Nicolas E Stier-Moses. Computing large market equilibria using abstractions. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, pages 745–746, 2019.

[8] Dave Levin, Katrina LaCurts, Neil Spring, and Bobby Bhattacharjee. Bittorrent is an auction: analyzing and improving bittorrent's incentives. In *Proceedings of the ACM SIGCOMM 2008 conference on Data communication*, pages 243–254, 2008.

[9] Benjamin Recht. A simpler approach to matrix completion. *Journal of Machine Learning Research*, 12(Dec):3413–3430, 2011.

[10] Vadim I Shmyrev. An algorithm for finding equilibrium in the linear exchange model with fixed budgets. *Journal of Applied and Industrial Mathematics*, 3(4):505, 2009.

[11] Madeleine Udell. Big data is low rank. *SIAG/OPT Views and News*, 2019. URL `http://wiki.siam.org/siag-op/images/siag-op/b/bd/ViewsAndNews-27-1.pdf`.

[12] Madeleine Udell, Corinne Horn, Reza Zadeh, Stephen Boyd, et al. Generalized low rank models. *Foundations and Trends® in Machine Learning*, 9(1):1–118, 2016.

[13] Fang Wu and Li Zhang. Proportional response dynamics leads to market equilibrium. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 354–363, 2007.