Robust Quadratic Programming for Price Optimization

Akihiro Yabe, Shinji Ito, Ryohei Fujimaki
NEC Corporation
a-yabe@cq.jp.nec.com, s-ito@me.jp.nec.com, rfujimaki@nec-labs.com

Abstract

The goal of price optimization is to maximize total revenue by adjusting the prices of products, on the basis of predicted sales numbers that are functions of pricing strategies. Recent advances in demand modeling using machine learning raise a new challenge in price optimization, i.e., how to manage statistical errors in estimation. In this paper, we show that uncertainty in recently-proposed *prescriptive price optimization* frameworks can be represented by a matrix normal distribution. For this particular uncertainty, we propose novel robust quadratic programming algorithms for conservative lower-bound maximization. We offer an asymptotic probabilistic guarantee of conservativeness of our formulation. Our experiments on both artificial and actual price data show that our robust price optimization allows users to determine best risk-return trade-offs and to explore safe, profitable price strategies.

1 Introduction

Product price is the most significant factor in determining sales. Better pricing strategies (i.e., pricing for multiple products) lead to higher total profits, and determination of the best price strategy is one of the most important strategic business decisions. Price optimization has been actively studied in marketing science and revenue management [Kunz and Crone, 2014] and has already achieved great success in such industries as online retail [Ferreira *et al.*, 2015], fast fashion [Caro and Gallien, 2012], hotels [Koushik *et al.*, 2012; Lee, 2011], and airlines [Côté *et al.*, 2003]. Most price optimization frameworks consist of two stages. The first is demand modeling, which reveals such complicated relationships among prices and sales quantities as price elasticity of demand [Marshall, 2009], cross price elasticity (a.k.a. cannibalization) [van Ryzin and Mahajan, 1999], and the law of diminishing marginal utility [Marshall, 2009]. The second derives the best price strategy by maximizing a utility function, typically total revenue or profit. Most existing studies for multi-product price optimization employ mixed-integer programming [Caro and Gallien, 2012; Koushik *et al.*, 2012; Lee, 2011] due to the discrete nature of individual prices.

Recently, [Ito and Fujimaki, 2017] have proposed *prescriptive price optimization*. They first employ recent advanced regression techniques (e.g. non-linear, sparse) to flexibly model complex relationships between demand and price with respect to multiple products, and they have shown that the profit maximization problem can be naturally expressed by means of a binary quadratic programming (BQP) problem. [Ito and Fujimaki, 2017] has proposed a relaxation method using semi-definite programming (SDP). Although the SDP method is more efficient than existing mixed-integer programming based methods, its scalability is still limited since it theoretically requires $O(M^6)$ computational time, where M is the number of products. [Ito and Fujimaki, 2016] has utilized connection between submodularity and the *substitute-goods property* [Koushik *et al.*, 2012] known in economics and has proposed an extremely fast proximal gradient algorithm with network-flow optimization that, under certain conditions, guarantees the global optimality of the original BQP problem.

A key drawback in previous works is its optimism with respect to its optimization. There exist two types of uncertainty in prescriptive price optimization, i.e., system noise and estimation error. System noise represents stochastic uncertainty in demand. Estimation error occurs due to the stochastic nature of machine learning, i.e., regression coefficients vary due to stochastic changes in sales records. Although previous works have not taken into account these uncertainties, it is known in the area of stochastic optimization that such uncertainties might significantly degrade the quality of "optimal" solutions due to optimistic bias in objective values [Delage and Ye, 2010; Duchi *et al.*, 2016].

This paper proposes a novel robust quadratic method for price optimization. Our key contributions are mainly twofold. First, we prove that uncertainty in prescriptive price optimization can be represented by a matrix normal distribution when the least square estimation is employed. This provides a natural robust formulation of price optimization as conservative lower-bound maximization. Although it is common for robust optimization to take system noise into account [Ben-Tal *et al.*, 2009], our study is unique in the sense that we explicitly model the uncertainty that occurs in estimation made using machine learning (i.e., estimation error). Second, we propose algorithms for robust quadratic optimization consisting of sequential relaxation to a non-robust counterpart that employs a non-robust algorithm (such as [Ito and Fujimaki, 2016; 2017]) as its sub-routine. The algorithm is guaranteed to obtain a local optimum by $O(\log(N/\delta))$ calls of the non-robust optimization oracle, which is much fewer than that of existing algorithms [Tütüncü and Koenig, 2004]. We have experimentally evaluated our robust pricing strategy for several values of conservativeness parameters, using an artificial price simulator, and are able to show its efficiency in terms of computational time and actual revenue improvement.

2 Problem Settings

2.1 Quadratic Price Optimization

Suppose we have M products and their prices and sales quantities are denoted by $x = (x_1, \ldots, x_M)^\top \in \mathcal{X} \subseteq \mathbf{R}^M$ and $y = (y_1, \ldots, y_M)^\top \in \mathbf{R}^M$, respectively, where $\mathcal{X}$ is a closed bounded set. Further, let us assume that the vector y of the sales quantities follows a multi-dimensional regression with non-linear basis functions as follows:

$$y = A^* v(x) + \varepsilon, \tag{1}$$

where $v : \mathcal{X} \to \mathbf{R}^N$ is a N-dimensional non-linear transformation of x, $A^* \in \mathbf{R}^{M\times N}$ is the true regression coefficient matrix representing cross price elasticity [Marshall, 2009] (this paper assumes $N \geq M$), and ε is system noise over $\mathbf{R}^M$ that follows a multidimensional normal distribution $\mathcal{N}(0, \Sigma^*)$ with zero mean and covariance matrix $\Sigma^* \in \mathbf{R}^{M\times M}$. Without loss of generality[1], we assume that the first M elements in $v(x)$ are linear features, i.e., $v(x) = (x^\top, N - M \text{ non-linear features})^\top$. The transformation v allows us to incorporate such non-linear effects as the law of diminishing marginal utility. We here omit non-price features but it is straightforward to incorporate such features as well, as pointed in [Ito and Fujimaki, 2016].

The goal of our price optimization is to maximize gross profit as defined by $(x - c)^\top y$, where $c = (c_1, \ldots, c_M)^\top \in \mathbf{R}^M$ is a cost vector. Hereinafter, for notational simplicity, this paper assumes $c = 0$ and thus maximizes gross revenue $x^\top y$. The problem can then be expressed by the following quadratic programming:

$$\min_{x\in\mathcal{X}} v(x)^\top Q^* v(x), \text{ where } Q^* := \begin{pmatrix} -A^* \\ 0 \end{pmatrix} \in \mathbf{R}^{N\times N}. \tag{2}$$

2.2 Link to Prescriptive Price Optimization

This subsection shows that the prescriptive optimization problem [Ito and Fujimaki, 2016; 2017] is reduced to (2), which considers the following general sparse additive models for demand modeling, $y_i = \sum_{j=1}^M f_{i,j}(x_j) + b_i$, where $f_{i,j} : P_j \to \mathbf{R}$ is any uni-variate feature transformation function and $P_i := \{p_{i,1}, p_{i,2}, \ldots, p_{i,L}\} \subseteq \mathbf{R}$ is a set of price candidates. The price optimization problem can then be formulated as follows:

$$\max_{x\in\mathcal{X}} \sum_{i=1}^M (x_i - c_i) \left(\sum_{j=1}^M f_{i,j}(x_j) + b_i \right), \tag{3}$$

where $\mathcal{X} := \{x \mid x_i \in P_i, i = 1, \ldots, M\}$. Again, for simplicity, we here assume $c = 0$, and this does not change the discussion that follows.

Without loss of generality, we can assume that there exists a set of basis $v_k : \mathbf{R} \to \mathbf{R}$ and coefficients $a'_{i,j,k}$ for $k = 1, 2, \ldots, K$ such that

$$f_{i,j}(x_j) = \sum_{n=1}^N a'_{i,j,k} v'_k(x_j) \tag{4}$$

for all $x_j \in P_j$. The most typical and simplest setting is $K = 1$ and $v_1(x_j) = x_j$ (linear basis), but in general we can represent (4) by having every $f_{i,j}$ as a basis for $i, j = 1, \ldots, M$. Then we have $y = Fv(x) + b_i$, where $F \in \mathbf{R}^{M\times KM}$ is defined by $F_{i,(k-1)M+j} = a'_{i,j,k}$. for $k = 1, 2, \ldots, K$ and $n = 1, 2, \ldots, N$. This shows the equivalence of (3) and (2).

3 Robust Price Optimization

3.1 Robust QP under Matrix Normal Uncertainty

Since the true model A^* cannot be obtained in practice, a natural way is to replace it by an estimator $\hat{A}$. Suppose we have a set of training samples denoted by $\{x_d, y_d\}_{d=1}^D$. We assume the least square estimator $\hat{A}$ of A^* defined as follows:

$$\hat{A} := \arg\min_A \sum_{d=1}^D \|y_d - Av(x_d)\|^2. \tag{5}$$

For $n \geq 1$, let I_n denote the identity matrix of size n. We define matrices V and W by

$$V := (v(x_1), v(x_2), \ldots, v(x_M)) \in \mathbf{R}^{N\times D} \tag{6}$$

$$W := VV^\top \in \mathbf{R}^{N\times N}. \tag{7}$$

Then by the polar decomposition, there exists a matrix $\Gamma \in \mathbf{R}^{N\times D}$ satisfying $\Gamma\Gamma^\top = I_N$ and $V = W^{1/2}\Gamma$. Let $U_{m,n}$ denote the random matrix over $\mathbf{R}^{m\times n}$, where each entry of $U_{m,n}$ is independently generated by $\mathcal{N}(0, 1)$ and where $\mathcal{N}$ represents a normal distribution.

The next proposition provides the distribution of $\hat{A}$.

Proposition 1. *Given $\{x_m\}_{m=1}^M$, $\hat{A}$ is generated by*

$$\hat{A} = A^* + \Sigma^{*1/2} U_{M,N} W^{-1/2}. \tag{8}$$

Proposition 1 indicates that $\hat{A}$ follows the matrix normal distribution [Gupta and Nagar, 1999], denoted by $\mathcal{MN}(A^*, \Sigma^*, W)$, as follows:

$$P(\hat{A}|A^*, \Sigma^*, W) \tag{9}$$

$$= \frac{\exp(-\frac{1}{2}\mathrm{tr}[W^{-1}(\hat{A} - A^*)^\top \Sigma^{*-1}(\hat{A} - A^*)])}{(2\pi)^{ND/2}|\Sigma^*|^{N/2}|W|^{D/2}}.$$

In order to derive a formulation for robust price optimization, we first consider the confidence region of $\hat{A}$. Let us define the estimator $\hat{\Sigma}$ of Σ^* and the confidence region $\mathcal{C}_\lambda$ as follows:

$$\hat{\Sigma} := \frac{1}{D}(Y - \hat{A}V)(Y - \hat{A}V)^\top \tag{10}$$

$$\mathcal{C}_\lambda : = \{A \mid A = \hat{A} + \hat{\Sigma}^{1/2} U W^{-1/2}, \|U\|_F \leq \lambda\}, \tag{11}$$

[1] If we don't use linear feature x, we can simply fix the first M columns of A^* to be zero.

where $U \in \mathbf{R}^{M \times N}$, and $\|U\|_F$ is the Frobenius norm of U. Then we derive the robust optimization problem with matrix normal uncertainty as

$$\max_{x \in \mathcal{X}} \min_{A \in \mathcal{C}_\lambda} x^\top A v(x) \Leftrightarrow \min_{x \in \mathcal{X}} \max_{Q \in \mathcal{C}'_\lambda} v(x)^\top Q v(x), \quad (12)$$

where

$$\mathcal{C}'_\lambda := \{Q = \hat{Q} + L_1^\top U L_2, \quad \|U\|_F \leq \lambda\}, \quad (13)$$

$$\hat{Q} := \begin{pmatrix} -\hat{A} \\ 0 \end{pmatrix}, L_1 := \begin{pmatrix} \hat{\Sigma}^{\frac{1}{2}} & 0 \\ 0 & 0 \end{pmatrix}, L_2 := W^{-\frac{1}{2}}.$$

Here $\lambda \geq 0$ is a scalar parameter for controlling the conservativeness (a smaller λ corresponds to a more aggressive strategy).

3.2 Probabilistic Guarantees of Robustness

This section presents an asymptotic probabilistic guarantee of our robust formulation, which can be regarded as a specialization of [Duchi *et al.*, 2016, Theorem 6]. It ensures that we can prevent optimistic bias in objective values and shows that our robust optimum value does not asymptotically over-estimate the true optimum revenue in designed probability.

Let χ_k^2 be a random variable generated by the chi-squared distribution with degree of freedom k.

Proposition 2. *Assume that the optimum solution of* (2) *is unique, and that the historical strategies x_d are generated by some distribution. Then we have*

$$\lim_{D \to \infty} P\left(\max_{Q \in \mathcal{C}'_\lambda} v(\hat{x})^\top Q v(\hat{x}) \geq v(\hat{x})^\top Q^* v(\hat{x}) \right)$$
$$= 1 - \frac{1}{2} P(\chi_1^2 \geq \lambda^2). \quad (14)$$

where $\hat{x} := \arg\min_{x \in \mathcal{X}} \max_{Q \in \mathcal{C}'_\lambda} v(x)^\top Q v(x)$.

4 Algorithms

4.1 Tight, Tractable Upper Bounds

Let us define the function $f : \mathcal{X} \to \mathbf{R}$ as follows:

$$f(x) = \max_U v(x)^\top Q v(x) \quad (15)$$

$$\text{s.t.} \quad Q = \hat{Q} + L_1^\top U L_2, \quad \|U\|_F \leq \lambda. \quad (16)$$

Our goal, then, is to obtain a strategy $x \in \mathcal{X}$ which minimizes f. We can remove the maximization in f and obtain another representation as follows:

Lemma 3. *It holds that*

$$f(x) = v(x)^\top \hat{Q} v(x) + \lambda \|L_1 v(x)\| \|L_2 v(x)\|, \quad (17)$$

where $\| \bullet \|$ *is the* ℓ_2 *norm of* $\bullet$.

Proof. We have

$$f(x) = v(x)^\top \hat{Q} v(x) + \max_{U : \|U\|_F \leq \lambda} v(x)^\top L_1^\top U L_2 v(x)$$
$$= v(x)^\top \hat{Q} v(x) + \lambda \|L_1 v(x)\| \|L_2 v(x)\|. \quad (18)$$

The last equality is attained when

$$U = \frac{\lambda (L_1 v(x)) \otimes (L_2 v(x))^\top}{\|(L_1 v(x)) \otimes (L_2 v(x))^\top\|_F}. \quad (19)$$

□

The term $\|L_1 v(x)\| \|L_2 v(x)\|$ is a product of the square roots of quartic functions w.r.t. $v(x)$ and is not tractable for optimization. The next proposition provides a tractable upper bound of f.

Proposition 4. *For any $x \in \mathcal{X}$ and $\gamma > 0$, it holds that*

$$f(x) \leq g(x, \gamma), \quad (20)$$

where

$$g(x, \gamma) := v(x)^\top \left(\hat{Q} + \lambda \frac{\gamma M_1 + M_2/\gamma}{2} \right) v(x). \quad (21)$$

and

$$M_1 := L_1^\top L_1, \quad M_2 := L_2^\top L_2. \quad (22)$$

The equality is attained if and only if $\gamma \|L_1 v(x)\| = \|L_2 v(x)\|$.

Proof. For any $v = v(x) \in \mathbf{R}^D$, we have

$$\|L_1 v\| \|L_2 v\| = \sqrt{\gamma (v^\top M_1 v))(v^\top M_2 v)/\gamma} \quad (23)$$
$$\leq \left(\gamma (v^\top M_1 v) + (v^\top M_2 v)/\gamma \right)/2 \quad (24)$$

The equality condition is $\gamma(v^\top M_1 v) = (v^\top M_2 v)/\gamma$, which is equivalent to $\gamma \|L_1 v(x)\| = \|L_2 v(x)\|$. □

The following corollary shows theoretical tightness of this upper bound, i.e., the optimal solution of the original problem is achievable with an appropriately chosen γ.

Corollary 5. *It holds that*

$$\min_{x \in \mathcal{X}} f(x) = \min_{x \in \mathcal{X}} \inf_{\gamma \in (0, \infty)} g(x, \gamma). \quad (25)$$

On the basis of Proposition 4 and Corollary 5, the optimization problem can be re-written as follows:

$$\min_x \inf_\gamma g(x, \gamma) \quad \text{s.t. } x \in \mathcal{X}, \gamma > 0. \quad (26)$$

4.2 Upper Bound Minimization Algorithms

Since robust optimization is essentially more difficult than its non-robust counterpart, we make the following assumption to ensure that the non-robust counterpart is solvable.

Assumption 6. For any Q' satisfying $Q' + Q'^\top \succeq \hat{Q} + \hat{Q}^\top$, we have a non-robust optimization oracle to solve $\min_{x \in \mathcal{X}} v(x)^\top Q' v(x)$.

This assumption immediately holds for convex non-robust counterparts and also holds for certain types of non-convex ones, such as with branch-and-bound methods [Burer and Vandenbussche, 2008], minimum cut methods for unconstrained binary quadratic programming under submodularity conditions [Kolmogorov and Zabin, 2004], and prescriptive price optimization under substitute goods property conditions [Ito and Fujimaki, 2017; 2016]. Note that this assumption is required for our theoretical analysis, and we can employ an approximation algorithm in practice.

For Eq. (21) and a given $\tilde{\gamma}$, $Q_{\tilde{\gamma}} + Q_{\tilde{\gamma}}^\top \succeq \hat{Q} + \hat{Q}^\top$ holds where $Q_{\tilde{\gamma}} := \hat{Q} + \lambda(\tilde{\gamma} M_1 + M_2/\tilde{\gamma})/2$, and hence the oracle in Assumption 6 is applicable. For later convenience, we formally summarize this in the following proposition[2].

[2] Even if such $\tilde{x}$ is not unique, the following discussion remains valid for an arbitrary choice of such $\tilde{x}$, and therefore here we do not consider uniqueness.

Algorithm 1 Golden Section Search

Require: $\hat{Q}$, L_1, L_2, λ, α, β, δ
  Initialize $a = \alpha, b = \beta, r = (\sqrt{5} - 1)/2$
  while $|a - b| \geq \delta$ **do**
    $c \leftarrow b - r * (b - a), d \leftarrow a + r * (b - a)$
    $b \leftarrow d$ if $h(c) < h(d)$, and $a \leftarrow c$ otherwise.
  end while
  Output $\tilde{x} := \arg\min_{x \in \mathcal{X}} g(x, \tilde{\gamma})$ where $\tilde{\gamma} = (a + b)/2$.

Algorithm 2 Coordinate Decent

Require: $\hat{Q}$, L_1, L_2, λ, γ_0, δ
  Check $\gamma_0 > 0$, and initialize $r \leftarrow \infty$, $\tilde{\gamma} \leftarrow \gamma_0$, $\tilde{x} \leftarrow \arg\min_x g(x, \tilde{\gamma})$.
  while $r - g(\tilde{x}, \tilde{\gamma}) > \delta$ and $\tilde{\gamma} \notin \{0, \infty\}$ **do**
    $r \leftarrow g(\tilde{x}, \tilde{\gamma})$
    $\tilde{x} \leftarrow \arg\min_x g(x, \tilde{\gamma})$
    $\tilde{\gamma} \leftarrow \arg\min_\gamma g(\tilde{x}, \gamma)$ by (27).
  end while
  Output $\tilde{x}$.

Proposition 7. *Under Assumption 6, for a given $\tilde{\gamma} \in (0, \infty)$, we can obtain $\tilde{x} := \arg\min_{x \in \mathcal{X}} g(x, \tilde{\gamma})$ by means of a single call of the non-robust optimization oracle.*

Further, on the basis of Proposition 4, for a given $\tilde{x}$, the infimum of $g(\tilde{x}, \gamma)$ with respect to γ is given by

$$\tilde{\gamma} = \arg\inf_{\gamma} g(\tilde{x}, \gamma) = \|L_2 v(\tilde{x})\| / \|L_1 v(\tilde{x})\|. \tag{27}$$

The above consideration indicates that the minimization of g w.r.t. one of x or γ is tractable when the other is fixed. This paper proposes two simple algorithms to solve Eq. (26) Let us define a function $h : (0, \infty) \to \mathbf{R}$ by

$$h(\gamma) := \min_{x \in \mathcal{X}} g(x, \gamma). \tag{28}$$

For robust prescriptive price optimization, we can employ algorithms proposed in [Ito and Fujimaki, 2017; 2016] as $h(\gamma)$.

Algorithm 1 is a golden section search [Kiefer, 1953] method in which we calculate $h(\gamma)$ by a single call of the oracle. An advantage of Algorithm 1 is a convergence guarantee for a fixed number of calculation of $h(\gamma)$ that provides theoretical validity to our upper bound minimization (see Section 4.3 for details). Algorithm 2 is a coordinate descent algorithm that alternates optimizations of $g(x, \gamma)$ w.r.t. x and γ. Though this algorithm offers no guarantee on the number of iterations needed for convergence, our experiments in Section 6 show that Algorithm 2 empirically converges with only a few iterations and is, in practice, much faster than Algorithm 1. Let us emphasize that, though more advanced optimization techniques might be applicable, our empirical results in Section 6 show that these simple algorithms converge fast enough in practice.

4.3 Convergence analysis

For a range $0 < \alpha \leq \beta$, let $h_{[\alpha,\beta]} : [\alpha, \beta] \to \mathbf{R}$ denote the function restricting the domain of h onto $[\alpha, \beta]$. Further, let us define an upper bound l by $l := (\|L_1\|_F + \|L_2\|_F) \max_{x \in \mathcal{X}} \|v(x)\| \geq \max_{x \in \mathcal{X}} \max\{\|L_1 v(x)\|, \|L_2 v(x)\|\}$. The following theorem guarantees the convergence of Algorithm 1 in a fixed number of iterations.

Theorem 8. *Suppose that v is continuous and that there does not exist $x \in \mathcal{X}$ such that $L_1 v(x) = L_2 v(x) = 0$. Under Assumption 6, for any $\delta' > 0$, setting $\alpha = 1/\beta$, $\beta = 2\lambda l^2/\delta$ and $\delta = \delta'/\lambda\beta^2 l^2$, Algorithm 1 outputs x with the following property by $O(\log(\lambda l/\delta'))$ calls of the non-robust optimization oracle: there exists a local optimum $x^* \in \mathcal{X}$ of f satisfying $f(x) \leq f(x^*) + \delta'$.*

For comparison with an existing method, as is discussed in the next subsection, we prove in the following proposition that, under a fairly natural assumption, the function f becomes convex. In this case, f has a unique minimum, and our algorithm is able to obtain the global minimum. Here $\mathcal{S}_+$ is a set of positive semidefinite matrices, and $\mathcal{Q}$ is defined by

$$\mathcal{Q} := \{(Q + Q^\top)/2 \mid Q = \hat{Q} + L_1^\top U L_2, \|U\|_F \leq \lambda\}. \tag{29}$$

Proposition 9. *If $\mathcal{X}$ is a convex set, v is affine, and $\mathcal{Q} \subseteq \mathcal{S}_+$, then f is a convex function.*

5 Related Work

Studies of robust optimization with ellipsoidal uncertainty were initiated independently by [Ben-Tal and Nemirovski, 1998] and [Ghaoui and Lebret, 1997]. For robust quadratic programming $\min_{x \in \mathcal{X}} x^\top Q x$, [Ben-Tal *et al.*, 2002] have considered the case in which A of $Q = A^\top A$ has ellipsoidal uncertainty. For the case in which Q has ellipsoidal uncertainty, which is also the case this paper considers, [Halldórsson and Tütüncü, 2003] have proposed an interior point method for a general self-concordant convex-concave function given by [Nemirovski, 1999] to this robust optimization. Further detailed information on robust optimization can be found in [Ben-Tal *et al.*, 2009; Bertsimas *et al.*, 2011].

In contrast to [Halldórsson and Tütüncü, 2003] which is applicable only when the convexity in Proposition 9 holds, our algorithm is applicable to non-convex robust quadratic programming under the existence of a (possibly approximate) algorithm for a non-robust counterpart. If Proposition 9 holds, both Algorithm 1 and [Halldórsson and Tütüncü, 2003] are applicable. In such a case, the non-robust counterpart is convex quadratic programming, which can be solved by $\sqrt{N}\log(1/\delta)$ Newton steps, and thus Algorithm 1 requires $O(\sqrt{N}\log(1/\delta)\log(N/\delta))$ Newton steps if $||v(x)||_\infty$, $||L_1||_\infty$ and $||L_1||_\infty$ are bounded[3]. If N is large, this is better than the $O(N\log(1/\delta))$ Newton steps of the algorithm in [Halldórsson and Tütüncü, 2003].

6 Experiments

We employed the state-of-the-art prescriptive price optimization of [Ito and Fujimaki, 2016] to calculate $h(\gamma)$ defined by Eq. (28), and compared our robust algorithm and non-robust

[3]If $||v(x)||_\infty$, $||L_1||_\infty$ and $||L_1||_\infty$ are bounded, $O(l) = O(\log N)$ holds.

counterpart. Note that the results with $\lambda = 0$ are reduced to the results of the algorithm in [Ito and Fujimaki, 2016].

6.1 Evaluation on Artificial Price Simulator

Simulation Data We conducted experiments with an artificial price simulator to support our robust formulation and theoretical analysis. We applied a similar simulation data generation process as that of the original paper [Ito and Fujimaki, 2016]. The true demand (regression) model followed (1), where $v(x) = (x_1, x_2, \ldots, x_M, 1)^\top$ were linear features with $N = M + 1$. Then the true coefficient matrix A^* was generated by $a^*_{i,i} \sim U([-2M, -M])$, $a^*_{i,j} \sim U([0, 2])$ if $i \neq j$, and $a^*_{i,N} \sim U([M/2, 3M/2])$. We defined the pricing strategies as $\mathcal{X} := \{0.6, 0.7, 0.8, 0.9, 1.0\}^M$ where 1.0 was the list price and 0.9 is 10%-off. The distribution of the non-diagonal entries $a^*_{i,j}$ for $j = 1, 2, \ldots, M$ were chosen to balance the effect of ith price and the other prices on the sales quantity of ith product. The linear term was arranged to obtain a moderate optimum strategy of ≈ 0.8 when the non-diagonal entries were zero. The training data $(\{x_d, y_d\})_{d=1}^D$ were generated by $x_{d,i} = 1.0, 0.9, 0.8, 0.7, 0.6$ with probability $0.5, 0.2, 0.1, 0.1, 0.1$, respectively for all $i = 1, 2, \ldots, M$, where the system noise in y_d followed $\mathcal{N}(0, 25I_M)$. Here $x_{d,i}$ has importance because it is natural to assume that the product is sold more at the list price than at discounted prices. In each experiment, we generated 10 true models randomly and 100 training datasets were generated for each dataset. The results are averages of all runs.

Effect of Robust Formulation Fig. 1 (left) demonstrates how robust solutions changed along with λ. We observed:

- The revenues were maximized with $\lambda = 2, 3, 4$ in all cases. By appropriately choosing the robustness parameter λ, we mitigate uncertainty in price optimization and achieve both better revenue and higher robustness.
- With the small training data ($D = 5M$), the non-robust solutions ($\lambda = 0$) result in much lower revenue than robust solutions. This is because the statistical uncertainty in estimation was large with small training data and therefore non-robust solutions easily became poor. As the size of training data increased, the difference became smaller.
- As λ increased, the standard deviation decreased (standard deviation is plotted only for $D = 5M$ cases for visualization purposes). This indicates that the solutions became more conservative and thus more stable.
- Even with the best chosen λ value, the obtained revenue is 5-10% worse than the true maximal revenue that was obtained with the true demand model. This difference came from estimation error, which we cannot avoid regardless of the choice of optimization algorithms. This result indicates that the true optimal value is not achievable in expectation and we must always undergo a certain reduction in the expected revenue because of the estimation error.

Probabilistic Guarantee with Finite Samples We next demonstrate that the asymptotic probabilistic guarantee on the upper-bound of the objective value holds with a practical number of training samples and reasonably large λ, and show that our robust optimization mitigates the over-estimation problem. Fig. 1 (middle) shows the result for $D = 100$ and $M = 10, 30, 50$ along with λ. The vertical axis is the probability of over-estimation $P(\max_{Q \in \mathcal{C}'_\lambda} \; v(x_\lambda)^\top Q v(x_\lambda) \leq v(x_\lambda)^\top Q^* v(x_\lambda))$, where x_λ is the robust optimum strategy with λ. We observed:

- The non-robust optimization over-estimated revenue in 70-80% of the cases, and the optimism of the non-robust optimization was empirically confirmed.
- Although the empirical probabilities (blue, green, red) were larger than the theoretical asymptotic value (pale blue) in this setting ($D = 100$), the difference rapidly decreased with increasing λ. With a reasonably large λ value (e.g. $\lambda \geq 3$), both the absolute value of empirical probabilities and the difference between empirical and theoretical probabilities became sufficiently small in practice.

Comparison of Algorithms We compared Algorithm 1 and Algorithm 2, with a sampling algorithm of min-max optimization that is a baseline for the robust optimization algorithm. The ith step of the sampling problem solves (12) with the replacement of $\mathcal{U}$ by finite samples $\{U_j\}_{j=1,\ldots,i-1}$, and we can then add a new U_i that is a maximizer for the current optimum solution. Observe that the sampling algorithm approximates the original robust optimization problem by adding a series of quadratic constraints, and thus in this framework the efficient price optimization algorithm [Ito and Fujimaki, 2016] is not applicable. For solving this sample approximation, we used the MIQCP solver of GUROBI optimizer version 6.0, after convex approximation of $\hat{Q} + U_j$.

Fig. 1 (right) shows convergence of the objective value over algorithm iterations. The majority of computational time was devoted to the evaluation of $h(\gamma)$ using non-robust optimization oracle in our two algorithms, and to the application of the MIQCP solver in the sampling algorithm. Thus we summarized this experiment along with algorithm iteration. All algorithms achieved the same objective values at convergence.

We observed that Algorithm 1 and the sampling algorithm needed almost the same number of iterations for convergence. However, while two applications of the non-robust optimization oracle in Algorithm 2 took 0.08 second per each iteration, the computational time needed for solving MIQCP grew as the number of quadratic constraints grew, and for $\lambda = 20$ it took 274 sec in the first 10 iterations. We also observed that Algorithm 2 took only a few iterations for convergence in all cases, and confirmed that Algorithm 2 was faster in practice and required only slightly more computational time than its non-robust counterpart.

6.2 Evaluation on Real Point-of-Sales Data

We applied the proposed method to real sales history of beers[4] [Ito and Fujimaki, 2017; Wang *et al.*, 2015]. The data consisted of prices and sales quantities on 50 products over 642 days. We used the price features for the prediction, and we obtained 50 regression models by means of the least

[4]The data has been provided by KSP-SP Co., LTD, http://www.ksp-sp.com.

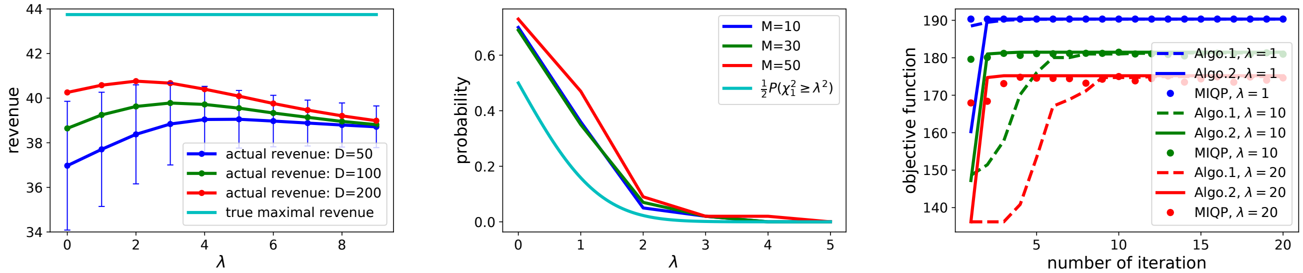

Figure 1: (left) Expected return and its deviation on robust strategies. The number of products is $M = 10$. The horizontal and vertical axes are, respectively, the robustness parameter λ and expected return. The blue, green, and red lines show the results for data size $D = 5M, 10M$, and $20M$, respectively. The pale blue line shows the revenue with the optimum strategy when true model is known.
(middle) The probability of an over-estimate of revenue for $D = 300$. The horizontal ax is the robustness parameter λ, and the vertical ax is the probability of over-estimation of revenue. The blue, green, and red line respectively show the result for the number of products $M = 10, 30, 50$. The pale blue line shows the theoretical guarantee in Proposition 2.
(right) Convergence of Algorithm 1, 2, and the sampling algorithm for $M = 20, D = 100$. The horizontal and vertical axes show the number of iteration and the value of objective functions, respectively. The blue, green, red line show the values for $\lambda = 1, 10, 20$, respectively.

Table 1: Real data experiments with sales histories of beers. On the basis of the least square estimate, we calculated robust optimum strategies for several risk-return parameter λ values. Pi is the product ID sorted by the ratio of (optimum price at $\lambda = 0$)/(average). Red and blue products indicate, respectively, risky and stable pricing strategy in small λ values. Green products indicate large cross price elasticity.

ID	λ=0	30	60	90	average
P1	326	502	519	541	542
P2	159	218	243	263	264
P3	326	457	493	537	541
P4	584	880	926	967	969
P5	148	219	233	244	245
P6	903	1058	1278	1478	1491
P7	743	743	1005	1210	1226
P8	903	1006	1234	1473	1490
P9	153	200	235	251	252
P10	153	229	240	252	252
P11	600	735	881	980	987
P12	107	142	158	174	176
P13	104	140	155	169	171
P14	903	903	1156	1463	1483
P15	153	222	237	251	251
P16	114	184	179	186	187
P17	675	675	855	1091	1107
P18	1019	1019	1041	1626	1670
P19	55	77	84	90	90
P20	55	69	81	89	90
P21	675	675	675	1071	1099
P22	138	138	189	220	222
P23	138	138	194	219	220
P24	230	285	285	294	294
P25	167	178	181	187	187
P26	313	313	307	312	312
P27	267	254	256	265	266
P28	265	252	253	264	264
P29	256	235	243	255	255
P30	1600	1364	1467	1584	1592
P31	189	189	184	188	188
P32	188	188	182	187	187
P33	162	143	148	160	161
P34	162	151	152	161	161
P35	255	229	239	251	253
P36	218	218	209	215	216
P37	208	202	200	206	206
P38	285	283	271	281	282
P39	189	189	183	187	187
P40	189	189	182	186	187
P41	189	189	180	186	187
P42	189	189	189	188	187
P43	91	79	82	89	90
P44	180	171	168	177	178
P45	255	232	239	251	252
P46	1124	694	894	1096	1110
P47	275	245	253	269	271
P48	1124	697	839	1084	1101
P49	1505	920	985	1438	1470
P50	1288	1062	912	1232	1257

square estimator. For each product, we set lower and upper bounds on the product price as 60% and 100% of its historically maximum price. We then conducted robust optimization for $\lambda = 0, 30, 60, 90$. Table 1 shows the optimized robust pricing strategies for each λ. We observed:

- For many products (red), non-robust prices ($\lambda = 0$) were drastically discounted from the average prices, and the prices monotonically increased optimal prices to average prices over λ. With the most conservative setting $\lambda = 90$, the optimized prices were very close to average values. These products are considered to have large price elasticity of demand. This is natural because the average prices and largely-discounted prices can be seen as, respectively, the most "tested" strategies and the most risky.
- For many different products (green), prices changed little over λ. These products are considered to have small price elasticity of demand and the list prices (average prices) are the safest and the most profitable.
- For some products (blue), non-robust prices were close to averaged prices. They once dropped to discounted prices and then returned to the averaged prices over λ. These products are mostly high-valued and therefore are considered to have large cross price elasticity. This complex behavior of the optimized prices was caused by the price changes in the other products.

Although we were unable to evaluate these strategies in real retail stores, we have learned here that our algorithm provides a way to simulate scenarios with different risk levels, and the users can explore for the best pricing strategies on the basis of their domain expertise.

7 Summary

This paper has proposed a novel robust quadratic optimization framework for prescriptive price optimization. Our statistical observations have revealed that the uncertainty occurring in estimation in machine learning follows a matrix normal distribution, which has lead us to formulate robust quadratic programming as a conservative upper-bound minimization. Our sequential algorithms for robust quadratic programming converge fast, both practically and theoretically, and can be implemented on the basis of existing non-robust price optimization algorithms. Experimental results on both artificial and actual price data showed that our method enables users to obtain both profitable and safe price strategies in prescriptive price optimization.

References

[Ben-Tal and Nemirovski, 1998] Aharon Ben-Tal and Arkadi Nemirovski. Robust convex optimization. *Mathematics of Operations Research*, 23(4):769–805, 1998.

[Ben-Tal *et al.*, 2002] Aharon Ben-Tal, Arkadi Nemirovski, and Cees Roos. Robust solutions of uncertain quadratic and conic-quadratic problems. *SIAM Journal on Optimization*, 13(2):535–560, 2002.

[Ben-Tal *et al.*, 2009] Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust optimization*. Princeton University Press, 2009.

[Bertsimas *et al.*, 2011] Dimitris Bertsimas, David B. Brown, and Constantine Caramanis. Theory and applications of robust optimization. *SIAM review*, 53(3):464–501, 2011.

[Burer and Vandenbussche, 2008] Samuel Burer and Dieter Vandenbussche. A finite branch-and-bound algorithm for nonconvex quadratic programming via semidefinite relaxations. *Mathematical Programming*, 113(2):259–282, 2008.

[Caro and Gallien, 2012] Felipe Caro and Jérémie Gallien. Clearance pricing optimization for a fast-fashion retailer. *Operations Research*, 60(6):1404–1422, 2012.

[Côté *et al.*, 2003] Jean-Philippe Côté, Patrice Marcotte, and Gilles Savard. A bilevel modelling approach to pricing and fare optimisation in the airline industry. *Journal of Revenue and Pricing Management*, 2(1):23–36, 2003.

[Delage and Ye, 2010] Erick Delage and Yinyu Ye. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations research*, 58(3):595–612, 2010.

[Duchi *et al.*, 2016] John C. Duchi, Peter W. Glynn, and Hongseok Namkoong. Statistics of robust optimization: A generalized empirical likelihood approach. *Stanford University*, 2016.

[Ferreira *et al.*, 2015] Kris J. Ferreira, Bin H. A. Lee, and David Simchi-Levi. Analytics for an online retailer: Demand forecasting and price optimization. *Manufacturing & Service Operations Management*, 2015.

[Ghaoui and Lebret, 1997] Laurent El Ghaoui and Hervé Lebret. Robust solutions to least-squares problems with uncertain data. *SIAM Journal on Matrix Analysis and Applications*, 18(4):1035–1064, 1997.

[Gupta and Nagar, 1999] Arjun K. Gupta and Daya K. Nagar. *Matrix variate distributions*, volume 104. CRC Press, 1999.

[Halldórsson and Tütüncü, 2003] Bjarni V. Halldórsson and Reha H. Tütüncü. An interior-point method for a class of saddle-point problems. *Journal of Optimization Theory and Applications*, 116(3):559–590, 2003.

[Ito and Fujimaki, 2016] Shinji Ito and Ryohei Fujimaki. Large–scale price optimization via network flow. In *Advances in Neural Information Processing Systems*, 2016.

[Ito and Fujimaki, 2017] Shinji Ito and Ryohei Fujimaki. Optimization beyond prediction: Prescriptive price optimization. In *Proceedings of the 23nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2017.

[Kiefer, 1953] Jack Kiefer. Sequential minimax search for a maximum. *Proceedings of the American Mathematical Society*, 4(3):502–506, 1953.

[Kolmogorov and Zabin, 2004] Vladimir Kolmogorov and Ramin Zabin. What energy functions can be minimized via graph cuts? *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(2):147–159, 2004.

[Koushik *et al.*, 2012] Dev Koushik, Jon A. Higbie, and Craig Eister. Retail price optimization at intercontinental hotels group. *Interfaces*, 42(1):45–57, 2012.

[Kunz and Crone, 2014] Timo P. Kunz and Sven F. Crone. Demand models for the static retail price optimization problem-a revenue management perspective. In *OASIcs-OpenAccess Series in Informatics*, volume 37. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2014.

[Lee, 2011] Seonah Lee. *Study of demand models and price optimization performance*. PhD thesis, Georgia Institute of Technology, 2011.

[Marshall, 2009] Alfred Marshall. *Principles of economics: unabridged eighth edition*. Cosimo, Inc., 2009.

[Nemirovski, 1999] Arkadi Nemirovski. On self-concordant convex–concave functions. *Optimization Methods and Software*, 11(1-4):303–384, 1999.

[Tütüncü and Koenig, 2004] R. H. Tütüncü and M. Koenig. Robust asset allocation. *Annals of Operations Research*, 132(1-4):157–187, 2004.

[van Ryzin and Mahajan, 1999] Garrett van Ryzin and Siddharth Mahajan. On the relationship between inventory costs and variety benefits in retail assortments. *Management Science*, 45(11):1496–1509, 1999.

[Wang *et al.*, 2015] Jialei Wang, Ryohei Fujimaki, and Yosuke Motohashi. Trading interpretability for accuracy: Oblique treed sparse additive models. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1245–1254. ACM, 2015.