# Optimisation methods: Linear and Integer optimisation

Siddharth Bhat

# Contents

## Chapter 1

# Introductory examples

### 1.1 Classification

Consider a classification problem where we have labeled data at $(x_i, y_i)$, with some labeled $+$ and other data points labeled $-$.

# Chapter 2

# LP-relaxation

We know that $z^*_{\mathtt{IP}} \leqslant z^*_{\mathtt{LP}}$. We can solve an LPproblem using a solver.

## 2.1 Bipartite Matching

Let $G \equiv (V \equiv X \cup Y, E \subset V \times V, w : E \to \mathbb{R})$. Graph is:

- Undirected, so $(x, y) \in E \iff (y, x) \in E$, $w((v, v')) = w((v', v))$.
- Bipartite, so that $(v, v') \in E \implies (v \in X \wedge v' \in Y) \vee (v \in Y \wedge v' \in X)$
- It's a little annoying to write the condition, but basically, for every edge, there's a unique weight which we adjust, even though the graph is undirected.

We wish to find $M \subseteq E$ such that:

$$\max_{e \in M} w_e$$

Can be transformed to:

$$\max_{e \in E} x_e w_e \qquad x_e \in \{0, 1\}$$
$$\sum_{e \in E, e = (v, v')} x_e = 1 \qquad \forall v \in V$$

Where the $x_e$ are variables to be discovered. We can now LP relax this, where $x_e \in [0, 1]$:

$$\max_{e \in E} x_e w_e \qquad x_e \in [0, 1]$$
$$\sum_{e \in E, e = (v, v')} x_e = 1 \qquad \forall v \in V$$

How do we go from the optimal solution to this problem, to an integer solution?

- Assume the LP is infeasible. This means that we have a vertex $u$ such that $\sum_{e \in E, e=(u,v)} x_e = 1$ fails. that is, there's a vertex in $u$ that is not connected to $v$. In this case, the IP is also infeasible.
- Now, we know that the LPis feasible. $a_1 \to b_1$ is not saturated means that $b_1 \to a_2$ is not saturated which implies that $a_2 \to b_2$ is not saturated, hence $b_2 \to a_1$ is not saturated. (TODO: add tikz picture). We can get a full cycle of edges with:

  $$x_{e_i} < 1$$
  $$x_{e_i} \in a_1 \xrightarrow{e_1} b_1 \xrightarrow{e_2} a_2 \xrightarrow{e_3} b_2 \xrightarrow{\dots}\xrightarrow{e_{i-1}} b_n \xrightarrow{e_i} a_1$$

  The number of edges here will be *even*. We can now pick a value $\epsilon \in (0, 1)$ such that:

  $$y_e \equiv \begin{cases} x_e^* + \epsilon & i \text{ is even, } x_e \text{ is in the cycle} \\ x_e^* - \epsilon & i \text{ is odd, } x_e \text{ is in the cycle} \\ x_e^* & \text{otherwise} \end{cases}$$

  Note that $y_e$ is a valid solution, since we can set $\epsilon$ to be smaller than the slack we had in the smallest value of $x_i$. We can show that the $\mathtt{cost}(y) \equiv \sum_{e \in E} w_e y_e$ is equal to:

  $$\mathtt{cost}(y) = \mathtt{cost}(x_e^*) + \epsilon\left(\Delta \equiv \sum_{i=1}^{n} (-1)^i w(e_i)\right)$$

  Remember that $x_e^*$ is the best solution, so we can have nothing better than $\mathtt{cost}(x_e^*)$. Hence, $\mathtt{cost}(y_e^*) \leqslant \mathtt{cost}(x_e^*)$, and hence, we are forced to conclude that $\Delta = 0$ (If $\Delta > 0$, pick $\epsilon > 0$, if $\Delta < 0$, pick $\epsilon < 0$).

  Hence, we can keep moving $\epsilon$ till an even edge becomes $1$ (alternatively, and odd edge becomes $0$). Hence, we can *keep rounding* till all our edges become $\{0, 1\}$.

So, we managed to start from an LP solution, and then *unrelax* it to construct an IP solution from it!

## 2.2 Min vertex cover

$G \equiv (V, E)$. We want to pick the smallest $F \subseteq V$, such that one end of all edges is in this cover.

$$\forall (u, v) \in E, u \in F \bigvee v \in F$$

Intuitively, these vertices $f \in F$ are watching over the edges, and each edge must be watched by at least one vertex.

TODO: add tikz picture

Integer program for the problem:

$x_v \in \{0, 1\}\ \forall v \in V \qquad \min \sum x_v \qquad \forall (u, v) \in E, x_u + x_v \geqslant 1$

LP relaxed program for the problem:

$$x_v \in [0,1]\ \forall v \in V \qquad \min \sum x_v \qquad \forall (u,v) \in E, x_u + x_v \geqslant 1$$

From the LP, we construct:

$$S_{\text{LP}} \equiv \left\{ u \,\middle|\, x_u^* \geqslant \frac{1}{2} \right\} \qquad \text{Claim: } S_{\text{LP}} \text{ is a vertex cover}$$

For each edge $(u,v) \in E$, since $u + v \geqslant 1$, we *cannot have that* $x_u < 0.5 \wedge x_v < 0.5$, since then $x_u + x_v < 1$. Hence, each edge will have one of its vertices with $x_{vertex} \geqslant 0.5$, and thus $S_{\text{LP}}$ is **a** vertex cover.

We now show **optimality** of $S_{\text{LP}}$.

$$\texttt{LP} \leqslant \texttt{IP} \qquad \text{since the problem is a minimization problem}$$

$$\sum_{u \in V} x_u \leqslant \sum_{u \in V} y_u \qquad x \text{ is } \texttt{LP} \text{ solution, } y \text{ is } \texttt{IP} \text{ solution}$$

$$|S_{\text{LP}}| = \sum_{x \in S_{\text{LP}}} 1(\text{counting}) \leqslant \sum_{u \in S_{\text{LP}}} 2x_u(\text{definition of } S_{\text{LP}}) \leqslant \sum_{u \in V} 2x_u(\text{enlarging } S_{\text{LP}} \text{ to V}) \leqslant \sum_{u \in V} 2y_u = 2|s_{opt}|$$

$$|S_{opt}| \leqslant S_{\text{LP}} \leqslant 2|S_{opt}|$$

So, we are at worst twice the size of the best vertex cover.

## 2.3 Maximum independent set

A set $I \subseteq V$ is independent if no two vertices in $I$ are connected by an edge. Once again, let $x_v \in \{0,1\}$ be indicators for a vertex being picked. We have the constraints:

$$\text{maximise} \sum_v x_v \qquad \text{subject to: } \forall (u,v) \in E,\ x_u + x_v \leqslant 1$$

# Chapter 3

# Formulating common operations in terms of `ILP`

## 3.1 TODO: MISSED CLASS! LOOKUP WHAT HAPPENED

# Chapter 4

# Matrix decompositions

### 4.0.1 Cholesky

Let $A$ be positive definite, $L$ be lower triangular. We decompose it as follows:

**Computing the decomposition**

$$\begin{aligned} A &= LL^T \\ \begin{bmatrix} a_{11} & A_{12} \\ A_{12} & A_{22} \end{bmatrix} &= \begin{bmatrix} l_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} l_{11} & L_{21}^T \\ 0 & L_{22}^T \end{bmatrix} = \begin{bmatrix} l_{11}^2 & l_{11}L_{21}^T \\ l_{11}L_{21}^T & L_{21}L_{21}^T + L_{22}L_{22}^T \end{bmatrix} \end{aligned}$$

**Solving** $Ax = b$

- We want to solve $Ax = b$, which is equivalent to $LL^Tx = b$
- let $L^Tx = u$. Now, $LL^Tx = b \equiv Lu = b$.
- Solve $Lu = b$ to find value of $u$.
- Solve $L^Tx = u$ to find value of $x$.

**Solving** $x = ((A^TA)^{-1}A^T)b$

- Let $B = A^TA$. Now, original equation is $Bx = A^Tb$.
- Compute $B$.
- Compute $d = A^Tb$
- Solve $Bx = d$. This is possible since $B$ is positive definite.

**Finding inverse**

- First decompose $A = LL^T$
- Solve $Ax_i = e_i$

### 4.0.2 LU

$$A = PLU$$

### 4.0.3 QR

$A = QR$ where $\dim(A) = (m, n)$ $\dim(Q) = (m, n)$, $\dim(R) = (n, n)$. $Q$ is orthogonal, $R$ is triangular.

We care about this decomposition in certain cases. For example, consider $x = (A^TA)^{-1}A^Tb$. Let $A = QR$. Now, the expression becomes

$$x = (A^TA)^{-1}A^Tb$$
$$x = ((R^TQ^T)(QR))^{-1}(R^TQ^T)b = (R^TR)^{-1})(R^TQ^T)b = (R^{-1}(R^T)^{-1}R^TQ^T)b = R^{-1}Q^Tb$$
$$Rx = Q^Tb \qquad \text{Let } Q^Tb = d \quad Rx = d$$
$$\text{Solve } Rx = d$$

**Comparison of Cholesky and QR for Least squares** For cholesky, we want to find $x = (A^TA)^{-1}A^Tb$. First, rewrite to $A^TAx = A^Tb$. $Bx = d$ where $B = A^TA$.

- Compute $B$
- Compute $d$
- Cholesky of $B = LL^T$
- Solve $Lv = d$
- Solve $Lx = v$

For QR:

- Factorize $A = QR$
- Compute $d = Q^Tb$
- Solve $Rx = d$

As we move from Cholesky to SVD, factorization cost increases, solution time decreases.

**Computing** QR

$$\begin{bmatrix} a_1 & A_2 \end{bmatrix} = \begin{bmatrix} q_1 & Q_2 \end{bmatrix} \begin{bmatrix} r_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix} = \begin{bmatrix} q_1r_{11} & q_1R_{12} + Q_2R_{22} \end{bmatrix}$$
$$a_1 = q_1r_{11} \quad A_2 = q_1R_{12} + Q_2R_{22}$$

Since $Q^TQ = I$, $\begin{bmatrix} q_1Q_2 \end{bmatrix}^T \begin{bmatrix} q_1Q_2 \end{bmatrix} = I$, hence $q_1^Tq_1 = 1$.

So,

$$a_1^T a_1 = (q_1 r_{11})^T (q_1 r_{11}) = (r_{11} q_1^T)(q_1 r_{11}) = r_{11}^2$$

. Hence,

$$r_{11} = \sqrt{a_1^T a_1} \qquad q_1 = a_1 / r_{11}$$

.

To find $R_{12}$, **TODO**

Next, Let $B = A_2 - q_1 R_{12} = Q_2 R_{22}$. Now perform QR on $B$.

### 4.0.4 SVD

$A = UDV^T$ where $\dim(A) = (m, n)$ $\dim(U) = (m, n)$, $\dim(D) = (n, n)$. $\dim(V) = (n, n)$. $U, V$ are orthogonal, $D$ is a diagonal matrix.

## 4.1 Bala's implicit enumeration algorithm

Used to solve $0, 1$ binary ILP problems.

good link for bala's algorithm