TTIC 31290: Machine Learning for Algorithm Design (Fall 2025)
Avrim Blum and Dravyansh Sharma

Lecture 16: 12/02/25 Lecturer: Avrim Blum (Notes by Dravyansh Sharma)

Outline for today

- Minimum-Weight Perfect Matching
- Learning Predictions
- Online Learning

1 Minimum-Weight Perfect Matching

Let $G = (V, E)$ be an undirected bipartite graph with $|V| = n$ even. Each edge $e \in E$ has a weight (cost) $c_e \in [0, C]$. A *perfect matching* $M \subseteq E$ is a set of $n/2$ edges such that every vertex of V is incident to exactly one edge of M.

Definition 1 (Minimum-Weight Perfect Matching)**.** *Given weights $c \in \mathbb{R}^E$, the minimum-weight perfect matching problem is*

$$\min_{M \in \mathcal{M}} c(M) := \sum_{e \in M} c_e,$$

where $\mathcal{M}$ denotes the family of perfect matchings of G.

A standard linear programming formulation admits a dual with one dual variable per vertex (potentials). For our learning discussion we will focus on learning useful duals (potentials) as warm-starts.

1.1 Dual potentials

Write the (standard) LP relaxation for matching using edge-incidence constraints:

$$\begin{aligned}
\min \sum_{e \in E} c_e x_e \\
\text{s.t. } \sum_{e \ni v} x_e = 1 \quad \forall v \in V, \\
x_e \geq 0 \quad \forall e \in E.
\end{aligned}$$

The dual variables are potentials $y \in \mathbb{R}^V$

$$\begin{aligned}
&\max \sum_{v \in V} y_v \\
&\text{s.t. } y_u + y_v \leq c_{uv}, \qquad \forall (u, v) \in E.
\end{aligned}$$

Classical algorithms (e.g. the Hungarian algorithm) start with a feasible dual solution (say set all $y_i = 0$ for one side of the bipartite graph, and the maximum feasible weight for the vertices on the other side) and adjust the dual solution until an optimal solution $y^* = y^*(c)$ is found. If we have a good prediction $\hat{y}$ for the optimal solution y^* for the dual problem, we can achieve a speed up that depends on $\|\hat{y} - y^*\|_1$ (Dinitz et al. show that one can achieve a running time of $\tilde{O}(|E|\sqrt{n} \cdot \min\{\|\hat{y} - y^*\|_1, \sqrt{n}\})$, achieving a graceful degradation in the performance with the quality of prediction).

2 Learning predictions for duals

Assume there is an underlying distribution $\mathcal{D}$ over cost vectors $c \in [0, C]^E$. We receive m independent samples $c^{(1)}, \ldots, c^{(m)} \sim \mathcal{D}$. The learner outputs a learned dual $\hat{y} \in \mathbb{R}^V$.

Since the running time and the quality of prediction depend on $\|\hat{y} - y^*\|_1$, our goal is to minimize expected error in ℓ_1 between predicted and optimal duals,

$$L(\hat{y}) = \mathbb{E}_{c\sim\mathcal{D}}\big[\|\hat{y} - y^*(c)\|_1\big].$$

Define the hypothesis class

$$\mathcal{H} = g_y(c) = \|y - y^*\|_1 : y \in \mathbb{R}^n.$$

Because y^* depends on c, the statistical complexity is controlled by the simpler class

$$H_n := f_y(x) = \|y - x\|_1 : y \in \mathbb{R}^n,$$

where x ranges over possible optimal dual vectors.

Theorem 1 (Pseudo-dimension bound)**.**

$$\mathrm{Pdim}(H_n) = O(n \log n).$$

Consequently, standard uniform convergence yields a sample complexity of learning $\hat{y}$ that scales as $\tilde{O}\big((n \cdot C)^2 n \log n/\varepsilon^2\big)$, where C is a bound on the cost on any edge.

Proof. The ℓ_1 distance decomposes as

$$f_y(x) = \sum_{i=1}^{n} |y_i - x_i|.$$

Dinitz et al. use a careful counting argument for the number of cells induced by hyperplanes to give a bound on the pseudo-dimension from first principles. Using the tools we have learned in the course, we can give a much simpler proof.

Indeed, we will analyze the structure of the dual function $f^*_x(y)$. If we consider the pieces induced by n hyperplanes $y_i - x_i = 0$, the function value is linear in y within any induced piece. Thus, the dual function class is $(\mathcal{F}, \mathcal{G}, n)$ decomposable where the piece functions in $\mathcal{F}$ are linear functions and the boundary functions in $\mathcal{G}$ are linear thresholds. Together, this implies the stated pseudo-dimension bound. □

3 Online learning and improved sample complexity

Turns out that there is additional structure in the loss function that allows us to both do online learning and improve the piecewise-structure based sample complexity bounds above.

3.1 Online Convex Optimization

We consider the standard Online Convex Optimization (OCO) setting. Let $\mathcal{K} \subseteq \mathbb{R}^d$ be a convex set of diameter D, meaning

$$\|x - y\| \le D \qquad \forall\, x, y \in \mathcal{K}.$$

At each round $t = 1, \ldots, T$:

- the learner chooses $x_t \in \mathcal{K}$,
- the adversary reveals a convex loss function $f_t : \mathcal{K} \to \mathbb{R}$,
- the learner incurs loss $f_t(x_t)$.

We assume that each loss f_t is L-Lipschitz with respect to the Euclidean norm:

$$\|\nabla f_t(x)\| \le L \qquad \forall\, x \in \mathcal{K},\ t = 1, \ldots, T.$$

Online Gradient Descent. Online Gradient Descent (OGD) performs the update

$$y_{t+1} = x_t - \eta \nabla f_t(x_t), \qquad x_{t+1} = \Pi_{\mathcal{K}}(y_{t+1}),$$

where $\Pi_{\mathcal{K}}$ denotes Euclidean projection onto $\mathcal{K}$.

Theorem 2 (Regret of Online Gradient Descent). *For any comparator $x^\star \in \mathcal{K}$, the regret of Online Gradient Descent satisfies*

$$R_T(x^\star) := \sum_{t=1}^{T} f_t(x_t) - \sum_{t=1}^{T} f_t(x^\star) \le \frac{D^2}{2\eta} + \frac{\eta L^2 T}{2}.$$

In particular, choosing

$$\eta = \frac{D}{L\sqrt{T}}$$

gives the bound

$$R_T(x^\star) \le DL\sqrt{T}.$$

Proof. By convexity of f_t we have

$$f_t(x_t) - f_t(x^\star) \le \langle \nabla f_t(x_t),\, x_t - x^\star \rangle.$$

Using the OGD update and Pythagorean theorem for Euclidean projections,

$$\|x_{t+1} - x^\star\|^2 \le \|x_t - \eta \nabla f_t(x_t) - x^\star\|^2.$$

Expanding the right-hand side gives

$$\|x_{t+1} - x^\star\|^2 \le \|x_t - x^\star\|^2 - 2\eta\langle \nabla f_t(x_t), x_t - x^\star\rangle + \eta^2\|\nabla f_t(x_t)\|^2.$$

Rearranging and using $\|\nabla f_t(x_t)\| \le L$ yields

$$\langle \nabla f_t(x_t), x_t - x^\star\rangle \le \frac{\|x_t - x^\star\|^2 - \|x_{t+1} - x^\star\|^2}{2\eta} + \frac{\eta L^2}{2}.$$

Summing over $t = 1$ to T, telescoping the norms, and using $\|x_1 - x^\star\| \le D$ and $\|x_{T+1} - x^\star\|^2 \ge 0$, we obtain

$$\sum_{t=1}^{T} \langle \nabla f_t(x_t), x_t - x^\star\rangle \le \frac{D^2}{2\eta} + \frac{\eta L^2 T}{2}.$$

Combining with the convexity inequality gives the claimed regret bound. □

3.2 Online learning for duals

We notice that the loss function $f_x^*(y)$ is convex in y and $\sqrt{n}$-Lipschitz. This allows us to apply Theorem 2 to get the following result.

Theorem 3. *Let $c^{(1)}, \ldots, c^{(T)} \in [0, C]^E$ be a sequence of cost vectors. Then OGD with step size $\eta = \frac{C}{\sqrt{T}}$ gives online predictions for duals $y_1, \ldots, y_T$ with regret*

$$\sum_{t=1}^{T} \|y_t - y^*(c^{(t)})\|_1 - \min_{y \in [-C,C]^n} \sum_{t=1}^{T} \|y - y^*(c^{(t)})\|_1 \le Cn\sqrt{T}.$$

Proof. It is sufficient to show that $f_x^*(y)$ is convex, $\sqrt{n}$-Lipschitz and $C\sqrt{n}$-bounded, and apply Theorem 2. □

By online-to-batch conversion, this implies that we can PAC-learn $\hat{y}$ with a smaller sample complexity $\tilde{O}\big((n \cdot C)^2/\varepsilon^2\big)$.

Additional Resources:

- Michael Dinitz, Sungjin Im, Thomas Lavastida, Benjamin Moseley, and Sergei Vassilvitskii. "Faster matchings via learned duals." Advances in Neural Information Processing Systems 34 (2021): 10393-10406.
- Mikhail Khodak, Maria-Florina Balcan, Ameet Talwalkar, and Sergei Vassilvitskii. "Learning predictions for algorithms with predictions." Advances in Neural Information Processing Systems 35 (2022): 3542-3555.
- Martin Zinkevich. "Online convex programming and generalized infinitesimal gradient ascent." International Conference on Machine Learning, ICML (2003): 928-935.