Appendix A  Details of binary search

Lemma 1. Let $\bar{F}(\lambda) = \frac{1}{4} \max_{y \in \mathcal{Y}^+} \left( \frac{1}{\lambda} h(y) + \lambda g(y) \right)^2$, then

$$\max_{y \in \mathcal{Y}} \Phi(y) \leq \min_{\lambda > 0} \bar{F}(\lambda)$$

and $\bar{F}(\lambda)$ is a convex function in $\lambda$.

Proof. First, let $\mathcal{Y}^+ = \{y|y \in \mathcal{Y}, h(y) > 0\}$, then $\max_{y \in \mathcal{Y}} \Phi(y) = \max_{y \in \mathcal{Y}^+} \Phi(y)$, since any solution $y$ such that $h(y) < 0$ is dominated by $y_i$, which has zero loss. Second, we prove the bound w.r.t. $y \in \mathcal{Y}^+$. In the following proof we use a quadratic bound (for a similar bound see [9]).

$$\max_{y \in \mathcal{Y}^+} \Phi(y) = \max_{y \in \mathcal{Y}^+} h(y)g(y) = \max_{y \in \mathcal{Y}^+} \frac{1}{4} \left( 2\sqrt{h(y)g(y)} \right)^2$$

$$= \frac{1}{4} \left( \max_{y \in \mathcal{Y}^+} \min_{\lambda > 0} \left\{ \frac{1}{\lambda} h(y) + \lambda g(y) \right\} \right)^2$$

$$\leq \frac{1}{4} \left( \min_{\lambda > 0} \max_{y \in \mathcal{Y}^+} \left\{ \frac{1}{\lambda} h(y) + \lambda g(y) \right\} \right)^2 \quad (13)$$

To see the convexity of $\bar{F}(\lambda)$, we differentiate twice to obtain:

$$\frac{\partial^2 \bar{F}(\lambda)}{\partial \lambda^2} = \frac{1}{4} \max_{y \in \mathcal{Y}^+} 6 \frac{1}{\lambda^4} h(y)^2 + 2g(y)^2 > 0$$

Similar to [11], we obtain a convex upper bound on our objective. Evaluation of the upper bound $\bar{F}(\lambda)$ requires using only the $\lambda$-oracle. Importantly, this alternative bound $\bar{F}(\lambda)$ does not depend on the slack variable $\xi_i$, so it can be used with algorithms that optimize the unconstrained formulation (4), such as SGD, SDCA and FW. As in [11], we minimize $\bar{F}(\lambda)$ using binary search over $\lambda$. The algorithm keeps track of $y_{\lambda_t}$, the label returned by the $\lambda$-oracle for intermediate values $\lambda_t$ encountered during the binary search, and returns the maximum label $\max_t \Phi(y_{\lambda_t})$. This algorithm focuses on the upper bound $\min_{\lambda > 0} \bar{F}(\lambda)$, and interacts with the target function $\Phi$ only through evaluations $\Phi(y_{\lambda_t})$ (similar to [11]).
Appendix B  An example of label mapping

Figure 5: A snapshot of labels during optimization with Yeast dataset. Each $2^{14} - 1$ labels is shown as a pot in the figure 5. X-axis is the $\Delta(y, y_1)$ and Y-axis is $1 + f_W(y) - f_W(y_1)$.

Appendix C  Monotonicity of $h$ and $g$ in $\lambda$

Proof. Let $g_1 = g(y_{\lambda_1}), h_1 = h(y_{\lambda_1}), g_2 = g(y_{\lambda_2}),$ and $h_2 = h(y_{\lambda_2})$.

\[
\begin{align*}
    h_1 + \lambda_1 g_1 & \geq h_2 + \lambda_1 g_2, \\
    h_2 + \lambda_2 g_2 & \geq h_1 + \lambda_2 g_1 \\
\Rightarrow h_1 - h_2 + \lambda_1(g_1 - g_2) & \geq 0, \\
- h_1 + h_2 + \lambda_2(g_2 - g_1) & \geq 0 \\
\Rightarrow (g_2 - g_1)(\lambda_2 - \lambda_1) & \geq 0
\end{align*}
\]

For $h$, change the role of $g$ and $h$. \hfill \Box

Appendix D  Improvements for the binary search

Appendix D.1  Early stopping

If $L = [\lambda_m, \lambda_M]$, and both endpoints have the same label, i.e., $y_{\lambda_m} = y_{\lambda_M}$, then we can terminate the binary search safely because from lemma 4, it follows that the solution $y_\lambda$ will not change in this segment.

Appendix D.2  Suboptimality bound

Let $K(\lambda)$ be the value of the $\lambda$-oracle, i.e.,

\[
K(\lambda) = \max_{y \in Y} h(y) + \lambda g(y).
\]

Lemma 5. $\Phi^*$ is upper bounded by

\[
\Phi(y^*) \leq \frac{K(\lambda)^2}{4\lambda}
\]
Proof.
\[ h(y) + \lambda g(y) \leq K(\lambda) \]
\[ \iff g(y)(h(y) + \lambda g(y)) \leq g(y)K(\lambda) \]
\[ \iff \Phi(y) \leq g(y)K(\lambda) - \lambda g(y)^2 \]
\[ = -\lambda \left( g(y) - \frac{K(\lambda)}{2\lambda} \right)^2 + \frac{K(\lambda)^2}{4\lambda} \leq \frac{K(\lambda)^2}{4\lambda} \]
\[ \square \]

Appendix E  Proof of the limitation of the $\lambda$-oracle search

**Theorem 1.** Let $\hat{H} = \max_y h(y)$ and $\hat{G} = \max_y g(y)$. For any $\epsilon > 0$, there exists a problem with 3 labels such that for any $\lambda \geq 0$, $y_\lambda = \arg\min_{y \in Y} \Phi(y) < \epsilon$, while $\Phi(y^*) = \frac{1}{4} \hat{H}\hat{G}$. Let $\hat{H} = \max_y h(y)$ and $\hat{G} = \max_y g(y)$. For any $\epsilon > 0$ and $\lambda > 0$, there exists a problem of 3 labels that $y_\lambda = \arg\min_{y \in Y} \Phi(y) < \epsilon$, and $\Phi(y^*) - \Phi(y_\lambda) = \frac{1}{4} \hat{H}\hat{G}$.

**Proof.** We will first prove following lemma which will be used in the proof.

**Lemma 6.** Let $A = [A_1 A_2] \in \mathbb{R}^2$, $B = [B_1 B_2] \in \mathbb{R}^2$, and $C = [C_1 C_2] \in \mathbb{R}^2$, and $A_1 < B_1 < C_1$. If $B$ is under the line $AC$, i.e., $\exists t, 0 \leq t < 1, D = tA + (1-t)C, D_1 = B_1, D_2 > B_2$. Then, $\hat{\lambda} \lambda \geq 0, v = [1 \lambda] \in \mathbb{R}^2$, such that
\[ v \cdot B > v \cdot A \text{ and } v \cdot B > v \cdot C \]
\[ \tag{16} \]

**Proof.** Translate vectors $A, B,$ and $C$ into coordinates of $[0, A_2], [a, b], [C, 0]$ by adding a vector $[-A_1, -C_2]$ to each vectors $A, B,$ and $C$, since it does not change $B - A$ or $B - C$. Let $X = C_1$ and $Y = A_2$.

If $0 \leq \lambda \leq \frac{X}{Y}$, then $v \cdot A = \lambda Y \leq X = v \cdot C, v \cdot (B - C) > 0 \iff (a - X) + \lambda b > 0$ corresponds to all the points above line $AC$. Similarly, if $\lambda \geq \frac{X}{Y}$, (16) corresponds to $a + \lambda (b - Y) > 0$ is also all the points above $AC$.

From lemma 6, if $y_1, y_2 \in Y$, then all the labels which lies under line $y_1$ and $y_2$ will not be found by $\lambda$-oracle. In the adversarial case, this holds when label lies on the line also. Therefore, Theorem 1 holds when there exists three labels, for arbitrary small $\epsilon > 0, A = [\epsilon, \hat{G}], B = [\hat{H}, \epsilon], \text{ and } C = [\frac{1}{2} \hat{H}, \frac{1}{2}G], Y = \{A, B, C\}$. In this case $\Phi \approx 0$.

Appendix F  Angular search

We first introduce needed notations. $\partial(\alpha)$ be the perpendicular slope of $\alpha$, i.e., $\partial(\alpha) = -\frac{1}{\partial(\alpha)} = -\frac{a_1}{a_2}$. For $A \subseteq \mathbb{R}^2$, let label set restricted to $A$ as $\hat{Y}_A = \hat{Y} \cap A$, and $y_{\alpha, A} = \arg\max_{y \in \hat{Y} \cap A} h(y) + \lambda g(y) = \arg\max_{y \in \hat{Y} \cap A} [\hat{y}]_1 + \lambda[\hat{y}]_2$.

Note that if $A = \mathbb{R}^2$, $y_{\alpha, \mathbb{R}^2} = y_\alpha$. For $P, Q \in \mathbb{R}^2$, define $\Lambda(P, Q)$ to be the area below the line $PQ$, i.e., $\Lambda(P, Q) = \{\hat{y} \in \mathbb{R}^2 | \hat{y}_1 \cdot [\hat{y}]_2 \geq \Phi(\hat{y}, \hat{A}) \}$ be the area above $C_\alpha$, and $\sum_\alpha = \{\hat{y} \in \mathbb{R}^2 | \hat{y}_1 \cdot [\hat{y}]_2 \geq \Phi(\hat{y}, \hat{A}) \}$ be the area below $C_\alpha$.

Recall the constrained $\lambda$-oracle defined in (8):
\[ y_{\lambda, \alpha, \beta} = \max_{y \in \hat{Y}, \alpha h(y) \geq g(y), \beta h(y) < g(y)} \Lambda_\lambda(y) \]
where $\alpha, \beta \in \mathbb{R}_+$ and $\alpha \geq \beta > 0$. Let $A(\alpha, \beta) \subseteq \mathbb{R}^2$ be the restricted search space, i.e., $A(\alpha, \beta) = \{a \in \mathbb{R}^2 | \beta \leq \partial(a) \leq \alpha\}$. Constrained $\lambda$-oracle reveals maximal $\Lambda_\lambda$ label within restricted area defined by $\alpha$ and $\beta$. The area is bounded by two lines whose slope is $\alpha$ and $\beta$. Define a pair $(\alpha, \beta), \alpha, \beta \in \mathbb{R}_+, \alpha \geq \beta > 0$ as an angle. The angular search recursively divides an angle into two different angles, which we call the procedure as a split. For $\alpha \geq \beta \geq 0$, let $\lambda = \frac{1}{\sqrt{\alpha \beta}}, z = \hat{y}_{\lambda, \alpha, \beta}$ and $z' = [\lambda[z]_2, \frac{1}{\lambda}[z]_1]$. Let $P$ be the point among $z$ and $z'$ which has the greater slope (any if two equal), and $Q$ be the other
point, i.e., if \( \partial(z) \geq \partial(z') \), \( P = z \) and \( Q = z' \), otherwise \( P = z' \) and \( Q = z \). Let \( R = \left[ \sqrt{\Lambda[2]^2 \cdot \Lambda[2]^2} \right] \).

Define split\((\alpha, \beta)\) as a procedure divides \((\alpha, \beta)\) into two angles \((\alpha^+, \gamma^+) = (\partial(P), \partial(R))\) and \((\gamma^+, \beta^+) = (\partial(R), \partial(Q))\).

First, show that \( \partial(P) \) and \( \partial(Q) \) are in between \( \alpha \) and \( \beta \), and \( \partial(R) \) is between \( \partial(P) \) and \( \partial(Q) \).

**Lemma 7.** For each split\((\alpha, \beta)\),

\[
\beta \leq \partial(Q) \leq \partial(R) \leq \partial(P) \leq \alpha
\]

**Proof.** \( \beta \leq \partial(z) \leq \alpha \) follows from the definition of constrained \( \lambda \)-oracle in (8).

\[
\partial(z') = \frac{1}{\lambda^2 \partial(z)} = \frac{\alpha \beta}{\partial(z)} \implies \beta \leq \partial(z') \leq \alpha \implies \beta \leq \partial(Q) \leq \partial(P) \leq \alpha.
\]

\( \partial(Q) \leq \partial(R) \leq \partial(P) \iff \min \left\{ \partial(z), \frac{1}{\lambda^2 \partial(z)} \right\} \leq \frac{1}{\lambda} \leq \max \left\{ \partial(z), \frac{1}{\lambda^2 \partial(z)} \right\} \) from \( \forall a, b \in \mathbb{R}_+, b \leq a \implies \beta \leq \partial(z) \leq \partial(P) \leq \alpha \).

After each split, the union of the divided angle \((\alpha^+, \gamma)\) and \((\gamma, \beta^+)\) can be smaller than angle \((\alpha, \beta)\). However, following lemma shows it is safe to use \((\alpha^+, \gamma)\) and \((\gamma, \beta^+)\) when our objective is to find \( y^* \).

**Lemma 8.**

\[
\forall a \in \mathcal{Y}_{A(\alpha, \beta)}, \Phi(a) > \Phi(y_{\lambda, \alpha, \beta}) \implies \beta^+ < \partial(a) < \alpha^+
\]

**Proof.** From lemma 2, \( \mathcal{Y}_{A(\alpha, \beta)} \subseteq \Lambda(P, Q) \). Let \( U = \{ a \in \mathbb{R}^2 | \partial(a) \geq \alpha + = \partial(P) \} \), \( B = \{ a \in \mathbb{R}^2 | \partial(a) \leq \beta + = \partial(Q) \} \), and two contours of function \( C = \{ a \in \mathbb{R}^2 | \Phi(a) = \Phi(y_{\lambda, \alpha, \beta}) \} \), \( S = \{ a \in \mathbb{R}^2 | \lambda \leq \lambda(a) = \lambda(\gamma_{\lambda, \alpha, \beta}) \} \). \( S \) is the upper bound of \( \Lambda(P, Q) \), and \( C \) is the upper bound of \( C = \{ a \in \mathbb{R}^2 | \Phi(a) \leq \Phi(y_{\lambda, \alpha, \beta}) \} \). \( P \) and \( Q \) are the intersections of \( C \) and \( S \). For area of \( U \) and \( B \), \( S \) is under \( C \), therefore, \( \Lambda(P, Q) \cap U \subseteq C \), and \( \Lambda(P, Q) \cap B \subseteq C \). It implies that \( \forall a \in (\Lambda(P, Q) \cap U) \cup (\Lambda(P, Q) \cap B) \implies \Phi(a) \leq \Phi(y_{\lambda, \alpha, \beta}) \). And the lemma follows from \( \Lambda(\alpha, \beta) = U \cup B \cup \{ a \in \mathbb{R}^2 | \beta^+ < \partial(a) < \alpha^+ \} \).

We associate a quantity we call a *capacity* of an angle, which is used to prove the suboptimality of the algorithm. For an angle \((\alpha, \beta)\), the capacity of an angle \( v(\alpha, \beta) \) is

\[
v(\alpha, \beta) := \frac{\sqrt{\alpha}}{\beta}
\]

Note that from the definition of an angle, \( v(\alpha, \beta) \geq 1 \). First show that the capacity of angle decreases exponentially for each split.

**Lemma 9.** For any angle \((\alpha, \beta)\) and its split \((\alpha^+, \gamma^+)\) and \((\gamma^+, \beta^+)\),

\[
v(\alpha, \beta) \geq v(\alpha^+, \gamma^+) = v(\alpha^+, \gamma^+)^2 = v(\gamma^+, \beta^+)^2
\]

**Proof.** Assume \( \partial(P) \geq \partial(Q) \) (the other case is follows the same proof with changing the role of \( P \) and \( Q \)), then \( \alpha^+ = \partial(P) \) and \( \beta^+ = \partial(Q) \). \( \partial(Q) = \frac{1}{\lambda^2 \partial(P)} = \frac{\alpha \beta}{\partial(P)} \), \( v(\alpha^+, \beta^+) = v(\partial(P), \partial(Q)) = \lambda \partial(P) = \partial(P) \frac{\partial(P)}{\sqrt{\alpha \beta}} \) Since \( \alpha \) is the upper bound and \( \beta \) is the lower bound of \( \partial(P), \sqrt{\frac{\beta}{\alpha}} \leq v(\partial(P), \partial(Q)) \leq \sqrt{\frac{\alpha}{\beta}} \). Last two equalities in the lemma are from \( v(\partial(P), \partial(R)) = v(\partial(R), \partial(Q)) = \sqrt{\frac{\alpha}{\sqrt{\beta}}} \) by plugging in the coordinate of \( R \).

**Lemma 10.** Let \( B(a) = \frac{1}{4} \left( a + \frac{1}{a} \right)^2 \). The suboptimality bound of an angle \((\alpha, \beta)\) with \( \lambda = \frac{1}{\sqrt{\alpha \beta}} \) is

\[
\max_{\tilde{y} \in \mathcal{Y}_{A(\alpha, \beta)}} \frac{\tilde{y}}{\Phi(y_{\lambda, \alpha, \beta})} \leq B(v(\alpha, \beta)).
\]
Proof. From lemma 2, $\hat{\mathcal{Y}}_{A(\alpha,\beta)} \subseteq \Lambda(P,Q) = \Lambda(z,z')$. Let $\partial(z) = \gamma$. From 7, $\beta \leq \gamma \leq \alpha$. Let $m = \arg\max_{\alpha \in A(z,z')} \hat{\Phi}(a)$. $m$ is on line $zz'$ otherwise we can move $m$ increasing direction of each axis till it meets the boundary $zz'$ and $\Phi$ only increases, thus $m = tz + (1 - t) z'$. $\hat{\Phi}(m) = \max_t \hat{\Phi}(tz + (1 - t)z')$. $\frac{\partial \hat{\Phi}(tz + (1 - t)z')}{\partial t} = 0 \implies t = \frac{1}{2}$. $m = \frac{1}{2}[z_1 + \lambda z_2 + \frac{z_1}{\lambda}]$.

$$\frac{\max_{\hat{y} \in \hat{\mathcal{Y}}_{A(\alpha,\beta)}} \hat{\Phi}(\hat{y})}{\hat{\Phi}(y_{\alpha,\beta})} = \frac{1}{4} \left( \sqrt{\frac{z_1}{\lambda z_2}} + \sqrt{\frac{\lambda z_2}{z_1}} \right)^2 \implies \frac{1}{4} \left( \sqrt{\frac{\alpha \beta}{\gamma}} + \sqrt{\frac{\gamma}{\alpha \beta}} \right)^2$$

Since $v(a) = v(\frac{1}{a})$ and $v(a)$ increases monotonically for $a \geq 1$, $B(a) \leq B(b) \iff \max \left\{ a, \frac{1}{a} \right\} \leq \max \left\{ b, \frac{1}{b} \right\}$

If $\frac{\sqrt{\alpha \beta}}{\gamma} \geq \frac{\gamma}{\sqrt{\alpha \beta}}$, then $\frac{\sqrt{\alpha \beta}}{\gamma} \leq \sqrt{\frac{\alpha}{\beta}}$ since $\gamma \geq \beta$. If $\frac{\gamma}{\sqrt{\alpha \beta}} \geq \frac{\sqrt{\alpha \beta}}{\gamma}$, then $\frac{\gamma}{\sqrt{\alpha \beta}} \leq \sqrt{\frac{\alpha}{\beta}}$ since $\gamma \leq \alpha$. Therefore, $\frac{\max_{\hat{y} \in \hat{\mathcal{Y}}_{A(\alpha,\beta)}} \hat{\Phi}(\hat{y})}{\hat{\Phi}(y_{\alpha,\beta})} = B \left( \frac{\sqrt{\alpha \beta}}{\gamma} \right) \leq B(v(\alpha,\beta))$. \:

Now we can prove the theorems.

**Theorem 2.** Angular search described in algorithm 2 finds optimum $y^* = \arg\max_{y \in \mathcal{Y}} \Phi(y)$ at most $t = 2M + 1$ iteration where $M$ is the number of the labels.

**Proof.** Denote $y_t, \alpha_t, \beta_t, z_t, z'_t, K_1^t$, and $K_2^t$ for $y, \alpha, \beta, z, z', K_1$, and $K_2$ at iteration $t$ respectively. $A(\alpha_t, \beta_t)$ is the search space at each iteration $t$. At the first iteration $t = 1$, the search space contains all the labels with positive $\Phi$, i.e., $\{ y | \Phi(y) \geq 0 \} \subseteq A(\alpha, \beta)$. At iteration $t > 1$, firstly, when $y_t = \emptyset$, the search area $A(\alpha_t, \beta_t)$ is removed from the search since $y_t = \emptyset$ implies there is no label inside $A(\alpha_t, \beta_t)$. Secondly, when $y_t \neq \emptyset$, $A(\alpha_t, \beta_t)$ is dequeued, and $K_1^t$ and $K_2^t$ is enqueued. From lemma 8, at every step, we are ensured that do not loose $y^*$. By using strict inequalities in the constrained oracle with valuable $s$, we can ensure $y_t$ which oracle returns is an unseen label. Note that split only happens if a label is found, i.e., $y_t \neq \emptyset$. Therefore, there can be only $M$ splits, and each split can be viewed as a branch in the binary tree, and the number of queries are the number of nodes. Maximum number of the nodes with $M$ branches are $2M + 1$. \:

**Theorem 4.** In angular search, described in Algorithm 2, at iteration $t$,

$$\Phi(\hat{y}^t) \geq \Phi(y^*)(v_1^{-\frac{1}{2}})$$

where $\hat{y}^t = \arg\max_t y^t$ is the optimum up to $t$, $v_1 = \max \left\{ \frac{\lambda_0}{\partial(\hat{y}^t)}, \frac{\partial(\hat{y}^t)}{\lambda_0} \right\}$. $\lambda_0$ is the initial $\lambda$ used, and $y_1$ is the first label returned by the constrained $\lambda$-oracle.

**Proof.** After $t \geq 2^r - 1$ iteration as in algorithm 2 where $r$ is an integer, for all the angle $(\alpha, \beta)$ in the queue $Q$, $v(\alpha, \beta) \leq (v_1)^{2^{1-r}}$. This follows from the fact that since the algorithm uses the depth first search, after $2^r - 1$ iterations all the nodes at the search is at least $r$. At each iteration, for a angle, the capacity is square rooted from the lemma 9, and the depth is increased by one. And the theorem follows from the fact that after $t \geq 2^r - 1$ iterations, all splits are at depth $r^t \geq r$, and at least one of the split contains the optimum with suboptimality bound with lemma 10. Thus,

$$\frac{\Phi(y^*)}{\Phi(\hat{y})} \leq B \left( (v_1)^{2^{1-r}} \right) < (v_1)^{2^{2-r}} \leq (v_1)^{r_1} \frac{1}{r_1}$$

\:

---

**Fast and Scalable Structural SVM with Slack Rescaling**

---
Theorem 5. Assuming $\Phi(y^*) > \phi$, angular search described in algorithm 2 with $\lambda_0 = \frac{\tilde{G}}{\tilde{H}}$, $\alpha_0 = \frac{\tilde{G}^2}{\phi}$, $\beta_0 = \frac{\phi}{H^2}$, finds $\epsilon$-optimal solution, $\Phi(y) \geq (1 - \epsilon)\Phi(y^*)$, in $T$ queries and $O(T)$ operations where $T = 4 \log \left( \frac{\tilde{G}H}{\phi} \right) \cdot \frac{1}{\epsilon}$, and $\delta$-optimal solution, $\Phi(y) \geq \Phi(y^*) - \delta$, in $T'$ queries and $O(T')$ operations where $T' = 4 \log \left( \frac{\tilde{G}H}{\phi} \right) \cdot \frac{\Phi(y^*)}{\delta}$.

Proof. $\Phi(y^*) > \phi \iff \frac{\phi}{H^2} < \frac{g(y^*)}{h(y^*)} = \partial(y^*) < \frac{\tilde{G}^2}{\phi}$. $v_1 = \max \left\{ \frac{\lambda_0}{\partial(y_1)}, \frac{\partial(y_1)}{\lambda_0} \right\}$ from Theorem 4. Algorithm finds $y^*$ if $\beta \leq \partial(y^*) \leq \alpha$, thus set $\alpha = \frac{\tilde{G}^2}{\phi}$ and $\beta = \frac{\phi}{H^2}$. Also from the definition of constrained $\lambda$-oracle, $\beta = \frac{\phi}{H^2} \leq \partial(y_1) \leq \alpha = \frac{\tilde{G}^2}{\phi}$. Therefore, $v_1 \leq \max \left\{ \frac{\lambda_0}{\partial(y_1)}, \frac{\partial(y_1)}{\lambda_0} \right\}$. And the upper bound of two terms equal when $\lambda_0 = \frac{\tilde{G}}{\tilde{H}}$, then $v_1 \leq \frac{\tilde{G}H}{\phi}$. $\delta$ bound follows plugging in the upper bound of $v_1$, and $\epsilon = \frac{\delta}{\Phi(y^*)}$.

Appendix G  Illustration of the angular search

Following figure 6 illustrates Angular search. Block dots are the labels from figure 5. Blue X denotes the new label returned by the oracle. Red X is the maximum point. Two straight lines are the upper bound and the lower bound used by the constrained oracle. Constrained oracle returns a blue dot between the upper and lower bounds. We can draw a line that passes blue X that no label can be above the line. Then, split the angle into half. This process continues until the $y^*$ is found.

![Figure 6: Illustration of the Angular search.](image)

(a) Iteration 1  (b) Iteration 2  (c) Iteration 3

Appendix H  Limitation of the constraint $\lambda$-oracle search

Theorem 3. Any search algorithm accessing labels only through $\lambda$-oracle with any number of the linear constraints cannot find $y^*$ in less than $M$ iterations in the worst case where $M$ is the number of labels.

Proof. We show this in the perspective of a game between a searcher and an oracle. At each iteration, the searcher query the oracle with $\lambda$ and the search space denoted as $\mathcal{A}$, and the oracle reveals a label according to the query. And the claim is that with any choice of $M - 1$ queries, for each query the oracle can either give an consistent label or indicate that there is no label in $\mathcal{A}$ such that after $M - 1$ queries the oracle provides an unseen label $y^*$ which has bigger $\Phi$ than all previous revealed labels.

Denote each query at iteration $t$ with $\lambda_t > 0$ and a query closed and convex set $\mathcal{A}_t \subseteq \mathbb{R}^2$, and denote the revealed label at iteration $t$ as $y_t$. We will use $y_t = \emptyset$ to denote that there is no label inside query space $\mathcal{A}_t$. Let $\mathcal{Y}_t = \{ y_{t'} | t' < t \}$.

Algorithm 3 describes the pseudo code for generating such $y_t$. The core of the algorithm is maintaining a rectangular area $\mathcal{R}_t$ for each iteration $t$ with following properties. Last two properties are for $y_t$. 

---

Heejin Choi, Ofer Meshi, Nathan Srebro
Algorithm 3 Construct a consistent label set \( \mathcal{Y} \).

**Input:** \( \{\lambda_i, A_i\}_{i=1}^{M-1}, \lambda_i > 0, A_i \subseteq \mathbb{R}^2, A_i \) is closed and convex region.

**Output:** \( \{y_t \in \mathbb{R}^2\}_{t=1}^{M-1}, y^* \in \mathbb{R}^2 \)

**Initialize:** \( R_0 = \{(a,b) | 0 < a, 0 < b\}, \mathcal{Y}_0 = \emptyset \).

1. for \( t = 1, 2, \ldots, M - 1 \) do
   2. if \( \mathcal{Y}_{t-1} \cap A_t = \emptyset \) then
   3. \( \hat{y} = \arg\max_{y \in \mathcal{Y}_t} h(y) + \lambda_t g(y) \).
   4. \( \tilde{R} = R_{t-1} \cap \{y|h(y) + \lambda_t g(y) < h(\hat{y}) + \lambda_t g(\hat{y}) \text{ or } y \notin A_t\} \).
  5. else
   6. \( \hat{y} = \emptyset, \tilde{R} = R_{t-1} - A_t \).
   7. if \( \tilde{R} \neq \emptyset \) then
   8. \( y_t = \emptyset, R_t = \text{FindRect}(\tilde{R}) \).
   9. else
   10. \( y_t = \text{FindPoint}(R_{t-1}, \lambda_t) \).
   11. \( R_t = \text{FindRect}(\text{Shrink}(R_{t-1}, y_t, \lambda_t)) \).
   12. if \( y_t \neq \emptyset \) then
   13. \( \mathcal{Y} = \mathcal{Y} \cup \{y_t\} \).
14. Pick any \( y^* \in R_{M-1} \)

1. \( \forall t' < t, \forall y \in R_t, \Phi(y) > \Phi(y_{t'}) \).
2. \( \forall t' < t, \forall y \in R_t \cap A_{t'}, h(y_{t'}) + \lambda_{t'} g(y_{t'}) > h(y) + \lambda_t g(y) \).
3. \( R_t \subseteq R_{t-1} \).
4. \( R_t \) is a non-empty open set.
5. \( y_t \in R_t \cap A_t \).
6. \( y_t = \arg\max_{y \in \mathcal{Y}_t \cap A_t} h(y) + \lambda_t g(y) \).

Note that if these properties holds till iteration \( M \), we can simply set \( y^* \) as any label in \( R_M \) which proves the claim.

First, we show that property 4 is true. \( R_0 \) is a non-empty open set. Consider iteration \( t \), and assume \( R_{t-1} \) is a non-empty open set. Then \( \tilde{R} \) is an open set since \( R_{t-1} \) is an open set. There are two unknown functions, \text{Shrink} and \text{FindRect}. For open set \( A \subseteq \mathbb{R}^2, y \in \mathbb{R}^2 \), let \( \text{Shrink}(A, y, \lambda) = A - \{y' | \Phi(y') \leq \Phi(y) \text{ or } h(y') + \lambda g(y') \geq h(y) + \lambda g(y)\} \).

Note that \( \text{Shrink}(A, y, \lambda) \subseteq A \), and \( \text{Shrink}(A, y, \lambda) \) is an open set. Assume now that there exists a \( y \) such that \( \text{Shrink}(R_{t-1}, y, \lambda_t) \neq \emptyset \) and \( \text{FindPoint}(R_{t-1}, \lambda_t) \) returns such \( y \). Function \text{FindPoint} will be given later. \( \text{FindRect}(A) \) returns an open non-empty rectangle inside \( A \). Note that \( \text{Rect}(A) \subseteq A \), and since input to \( \text{Rect} \) is always non empty open set, such rectangle exists. Since \( R_0 \) is non-empty open set, \( \forall t, R_t \) is a non-empty open set.

Property 3 and 5 are easy to check. Property 1 and 2 follows from the fact that \( \forall t \in \{t | y_t \neq \emptyset\}, \forall t' > t, R_{t'} \subseteq \text{Shrink}(R_{t-1}, y_t, \lambda_{t-1}) \).

Property 6 follows from the facts that if \( \mathcal{Y}_{t-1} \cap A_t \neq \emptyset, \tilde{R} = 0 \implies R_{t-1} \subseteq \{y|h(y) + \lambda_t g(y) > h(\hat{y}) + \lambda_t g(\hat{y}) \text{ and } y \in A_t\} \), otherwise \( \mathcal{Y}_{t-1} \cap A_t = \emptyset, \) and \( R_{t-1} \subseteq A_t \).

\( \text{FindPoint}(A, \lambda) \) returns any \( y \in A - \{y \in \mathbb{R}^2 | \lambda y_2 = y_1\} \). Given input \( A \) is always an non-empty open set, such \( y \) exists. \( \text{Shrink}(R_{t-1}, y, \lambda_t) \neq \emptyset \) is ensured from the fact that two boundaries, \( c = \{y' | \Phi(y') = \Phi(y)\} \) and \( d = \{h(y) + \lambda g(y) = h(y) + \lambda g(y)\} \) meets at \( y \). Since \( c \) is a convex curve, \( c \) is under \( d \) on one side. Therefore the intersection of set above \( c \) and below \( d \) is non-empty and also open.
Appendix I  Additional Plots from the Experiments

Figure 7: Additional experiment plot (RCV)