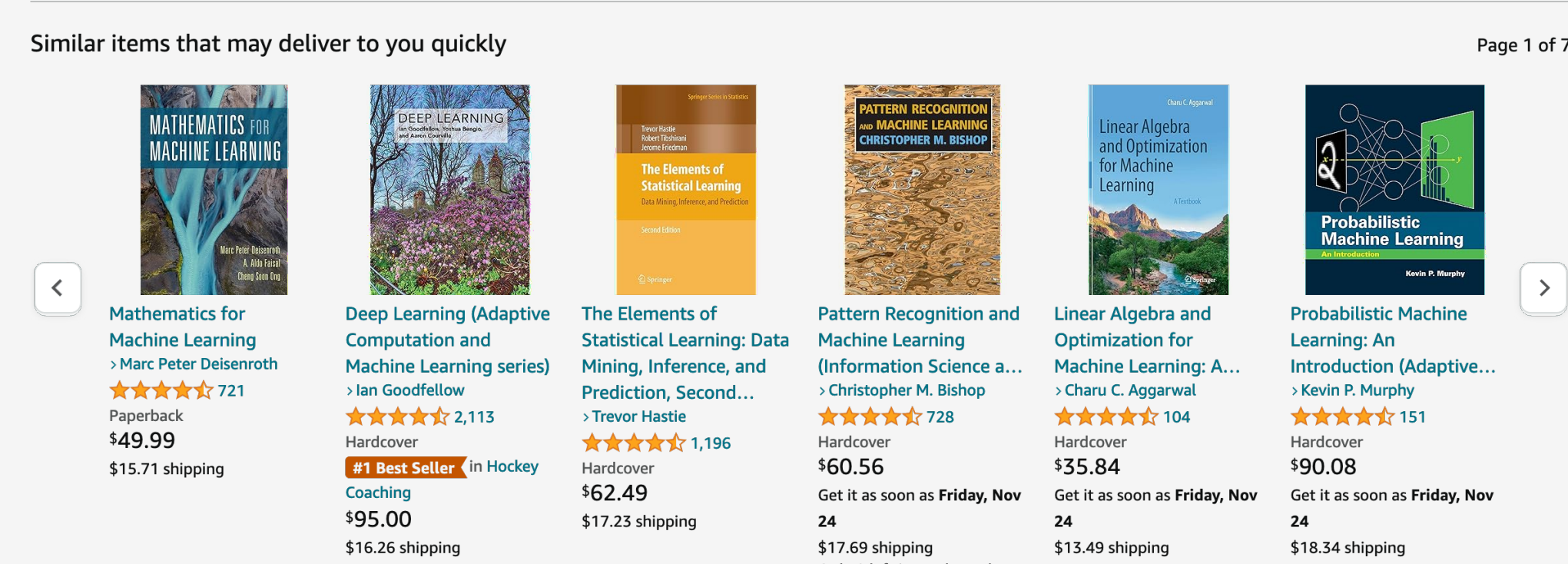


## Multi-label classification with budget at $k$

- Multi-label classification:  $\mathbf{x} \in \mathcal{X} \rightarrow \mathbf{y} \in \mathcal{Y} := \{0, 1\}^m$ ,
- In areas of **extreme classification** and **recommender systems**, many problems are naturally **budgeted at  $k$** , i.e., one needs to make exactly  $k$  predictions ( $\|\hat{\mathbf{y}}\|_1 = k$ ).



## Macro-at- $k$ metrics

Instead of calculating and averaging performance **instance-wise**, calculate per-label  $\psi^j$  and average over labels (**label-wise**):

$$\Psi @k(\mathbf{Y}, \hat{\mathbf{Y}}) = \sum_{j=1}^m \psi^j(\mathbf{y}_j, \hat{\mathbf{y}}_j).$$

Can balance contribution from all labels.

**Reduces to macro-average** if all  $\psi^j$  are the same.

Covers (instance) Precision@ $k$  and Hamming score.

These performance measures can be written as a function of **label-wise confusion tensor**:

$$\hat{\mathbf{C}}(\mathbf{Y}, \hat{\mathbf{Y}}) := [\mathbf{C}(\mathbf{y}_1, \hat{\mathbf{y}}_1), \dots, \mathbf{C}(\mathbf{y}_m, \hat{\mathbf{y}}_m)].$$

This tensor is a vector of **binary confusion matrices** for each label:

$$\hat{\mathbf{C}}(\mathbf{y}, \hat{\mathbf{y}}) := \begin{pmatrix} \text{tn} := \frac{1}{n} \sum_{i=1}^n (1 - y_i)(1 - \hat{y}_i) & \text{fp} := \frac{1}{n} \sum_{i=1}^n (1 - y_i)\hat{y}_i \\ \text{fn} := \frac{1}{n} \sum_{i=1}^n y_i(1 - \hat{y}_i) & \text{tp} := \frac{1}{n} \sum_{i=1}^n y_i\hat{y}_i \end{pmatrix}.$$

We call metrics  $\Psi$  defined on  $\hat{\mathbf{C}}$  **general confusion tensor measures**.

$\mathbf{Y} \backslash \hat{\mathbf{Y}}$	:1	:2	:3	:4	:5	:6	
1:	1	1	0	0	0	0	2/3
2:	1	1	0	0	0	0	2/3
3:	1	1	0	0	1	0	3/3
4:	1	0	1	0	0	0	1/3
5:	1	1	0	0	1	0	2/3

5/5 : 3/4 1/2 0/1 1/2 0/1

Metric	$\psi(\text{tp}, \text{fp}, \text{fn}, \text{tn})$
Precision	$\frac{\text{tp}}{\text{tp} + \text{fp}}$
Recall	$\frac{\text{tp}}{\text{tp} + \text{fn}}$
$F_1$	$\frac{2\text{tp}}{2\text{tp} + \text{fn} + \text{fp}}$
Jaccard	$\frac{\text{tp}}{\text{tp} + \text{fp} + \text{fn}}$
Balanced Acc., G-Mean, etc.	

## Population Utility Framework

Two **conflicting** statistical frameworks:

**Expected Test Utility (ETU):** Given a **specific set of instances** with unknown stochastic labels, how well can we expect a classifier to perform:

$$\Psi_{ETU} = \mathbb{E}_{\mathbf{Y}|\mathbf{X}}[\Psi(\hat{\mathbf{C}}(\mathbf{Y}, \hat{\mathbf{Y}}))]$$

**Population Utility (PU):** Given a **distribution of instances**, how well can we expect a classifier to perform on the population level, i.e., for an infinite sample:

$$\Psi_{PU} = \Psi\left(\mathbb{E}_{\mathbf{y}}[\hat{\mathbf{C}}(\mathbf{y}, \hat{\mathbf{y}})]\right).$$

The PU allows for **pointwise** predictions, independently for each  $\mathbf{x}$ , while ETU requires whole set of instances to provide the prediction.

## The optimal PU classifier

Conditional **marginal probability** of a label:  $\eta_j(\mathbf{x}) := \mathbb{P}[y_j = 1 | \mathbf{x}]$ .

**Randomized** classifier budgeted at  $k$  is defined as probability of predicting  $y_j$  for given  $\mathbf{x}$ :  $h_j(\mathbf{x}) := \mathbb{P}[\hat{y}_j = 1 | \mathbf{x}]$  and  $\|\mathbf{h}\|_1 = k$

**Population** confusion matrix of  $\mathbf{h}$ :

$$\mathbf{C}^j(h_j) := \mathbb{E}_{\mathbf{x}}[\hat{\mathbf{C}}] = \begin{pmatrix} \mathbb{E}_{\mathbf{x}}[(1 - \eta_j(\mathbf{x}))(1 - h_j(\mathbf{x}))] & \mathbb{E}_{\mathbf{x}}[(1 - \eta_j(\mathbf{x}))h_j(\mathbf{x})] \\ \mathbb{E}_{\mathbf{x}}[\eta_j(\mathbf{x})(1 - h_j(\mathbf{x}))] & \mathbb{E}_{\mathbf{x}}[\eta_j(\mathbf{x})h_j(\mathbf{x})] \end{pmatrix}.$$

Set of confusion tensors achievable by  $\mathbf{h}$ :  $\mathcal{C}_{\mathbb{P}}^{\otimes k} := \{\mathbf{C}(\mathbf{h}) : \mathbf{h} \in \mathcal{H}\}$ .

We prove that  $\mathcal{C}_{\mathbb{P}}^{\otimes k}$  is compact, simplifying the subsequent analysis.

Optimal classifier for **linear** metric  $\Psi(\mathbf{C}) = \mathbf{G} \cdot \mathbf{C} = \sum_{j=1}^m \mathbf{G}^j \cdot \mathbf{C}^j$  for **gain tensor**  $\mathbf{G} = (\mathbf{G}^1, \dots, \mathbf{G}^m)$  is:

$$\mathbf{h}^*(\mathbf{x}) = \text{top}_k(\mathbf{a} \odot \boldsymbol{\eta}(\mathbf{x}) + \mathbf{b}),$$

where the vectors  $\mathbf{a}$  and  $\mathbf{b}$  are given by:

$$a_j = G_{00}^j + G_{11}^j - G_{01}^j - G_{10}^j, \quad b_j = G_{01}^j - G_{00}^j.$$

The optimal classifier for **general confusion tensor metrics**:

- If:  $\boldsymbol{\eta}(\mathbf{x})$  is **absolutely continuous**,
- and  $\Psi$  is **strictly monotonic and differentiable**,

there exists an optimal  $\mathbf{C}^* \in \mathcal{C}_{\mathbb{P}}^{\otimes k}$ , that is  $\Psi(\mathbf{C}^*) = \Psi^*$ . Any classifier  $\mathbf{h}^*$  maximizing the linear utility  $\mathbf{G} \cdot \mathbf{C}(\mathbf{h})$  with  $\mathbf{G} = \nabla_{\mathbf{C}}\Psi(\mathbf{C}^*)$  also maximizes  $\Psi(\mathbf{h})$ .

## Difficulty of optimizing metrics-at- $k$

The budget at  $k$  **ouples** the label-wise binary problems:

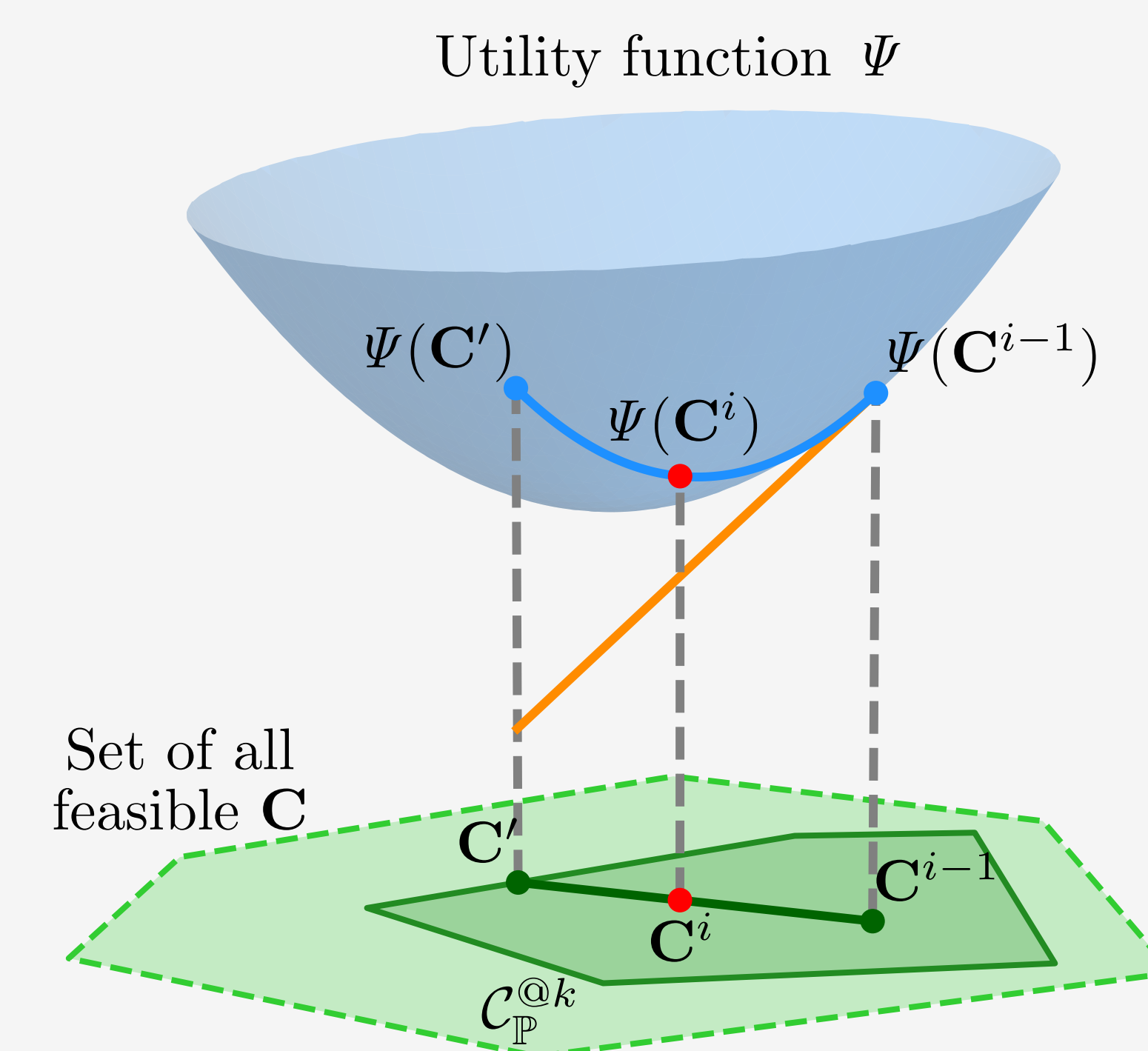
	Distribution $\mathbf{A}   \mathbf{B}$				Optimal $\mathbf{h}_A^*(\mathbf{x})   \mathbf{h}_B^*(\mathbf{x})$		
	$P$	$\eta_1$	$\eta_2$	$\eta_3$	$\pi_1$	$\pi_2$	$\pi_3$
$\mathbf{x}_1$	0.5	0.4	0.2	0.6	1   0	0   1	1
$\mathbf{x}_2$	0.5	0.8	0.4	0.4   0.8	1	1	0

$\Psi_{\text{Jaccard}}(\mathbf{C}(\mathbf{h}_A^*, \mathbf{A})) \approx 0.454, \quad \Psi_{\text{Jaccard}}(\mathbf{C}(\mathbf{h}_B^*, \mathbf{B})) \approx 0.471.$

## Finding the optimal PU classifier

We find the optimal classifier  $\mathbf{h}$  using **Frank Wolfe-based** optimization procedure:

- Split training set  $\mathcal{S}$  into  $\mathcal{S}_1$  (for learning  $\hat{\boldsymbol{\eta}}(\mathbf{x})$ ) and  $\mathcal{S}_2$  (used by Frank-Wolfe).
- Initialize probabilistic classifier as a set of deterministic classifiers  $\mathbf{h} \leftarrow \{\mathbf{h}^0\}$  and their probabilities  $\alpha \leftarrow \{\alpha^0 = 1\}$  and calculate  $\mathbf{C}^0 \leftarrow \hat{\mathbf{C}}(\mathbf{h}^0, \mathcal{S}_2)$ .
- Perform iteration  $i$  until convergence:
  - Calculate tensor of gradients of  $\mathbf{C}^{i-1}$  with respect to  $\Psi$ :  $\mathbf{G}^i \leftarrow \nabla_{\mathbf{C}}\Psi(\mathbf{C}^{i-1})$ .
  - Construct the next classifier:  $\mathbf{h}^i(\mathbf{x}) \leftarrow \text{top}_k(\mathbf{a}^i \odot \hat{\boldsymbol{\eta}}(\mathbf{x}) + \mathbf{b}^i)$  with  $\mathbf{a}^i \leftarrow \mathbf{G}_{11}^i + \mathbf{G}_{00}^i - \mathbf{G}_{01}^i - \mathbf{G}_{10}^i$ ,  $\mathbf{b}^i \leftarrow \mathbf{G}_{01}^i - \mathbf{G}_{00}^i$ .
  - Calculate the confusion tensor of the next classifier  $\mathbf{h}^i$ :  $\mathbf{C}^i \leftarrow \hat{\mathbf{C}}(\mathbf{h}^i, \mathcal{S}_2)$ .
  - Find the best combination of  $\mathbf{C}^{i-1}$  and  $\mathbf{C}^i$ :  $\alpha^i \leftarrow \text{argmax}_{\alpha \in [0,1]} \Psi((1 - \alpha)\mathbf{C}^{i-1} + \alpha\mathbf{C}^i)$ ,  $\mathbf{C}^i \leftarrow (1 - \alpha^i)\mathbf{C}^{i-1} + \alpha^i\mathbf{C}^i$ .
  - Update the probabilistic classifier:  $\mathbf{h} \leftarrow \mathbf{h} \cup \{\mathbf{h}^i\}$ ,  $\alpha \leftarrow \alpha \cup \{\alpha^i\}$ ,  $\alpha^j \leftarrow \alpha^j \times \alpha^i$  for  $j \in [i - 1]$ .

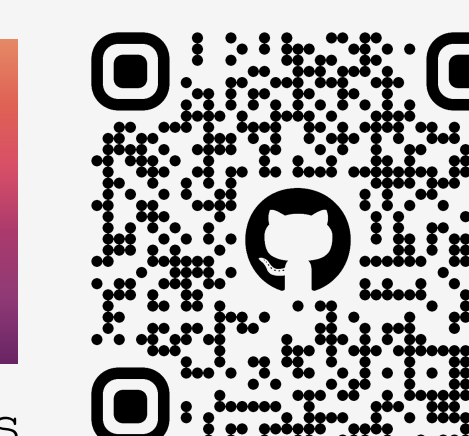


The Frank Wolfe **guarantees** to find the optimal classifier if  $\Psi$  is **concave** over  $\mathcal{C}_{\mathbb{P}}^{\otimes k}$ , **L-Lipschitz**, and  **$\beta$ -smooth** w.r.t. the 1-norm.

Read the paper



Try our library



## Experimental results

We compare classifiers found using the Frank Wolfe algorithm with with popular re-weighting strategies:

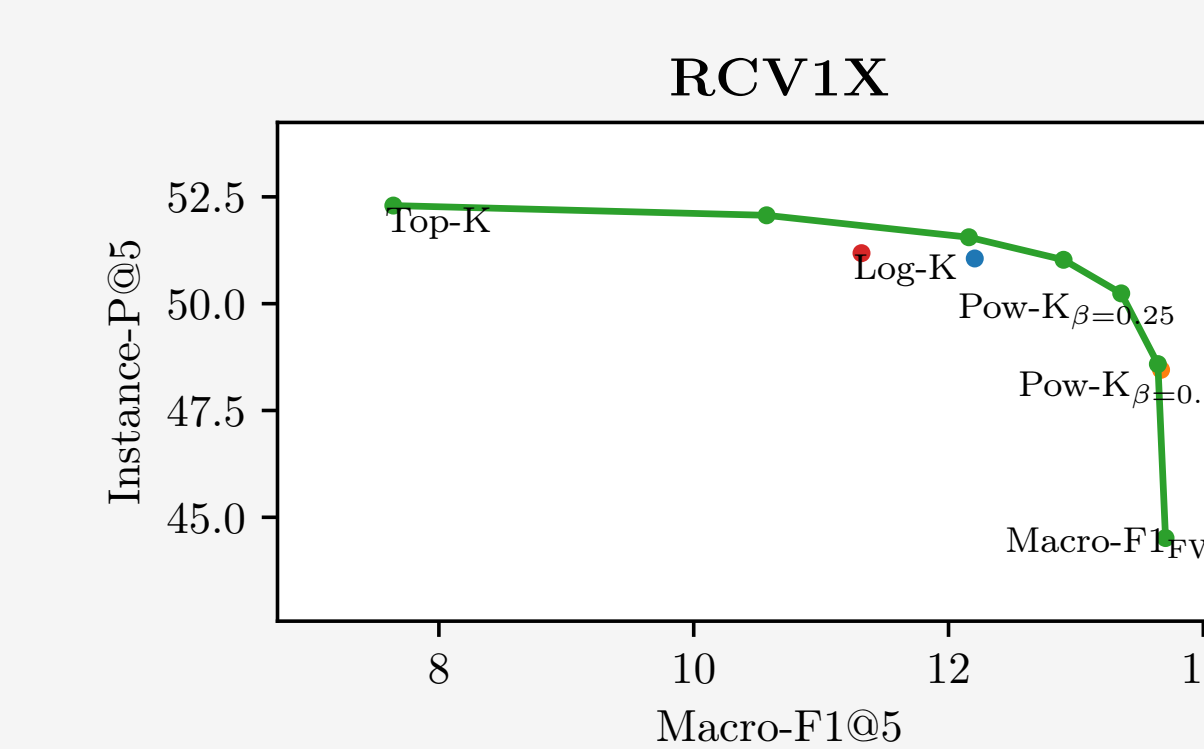
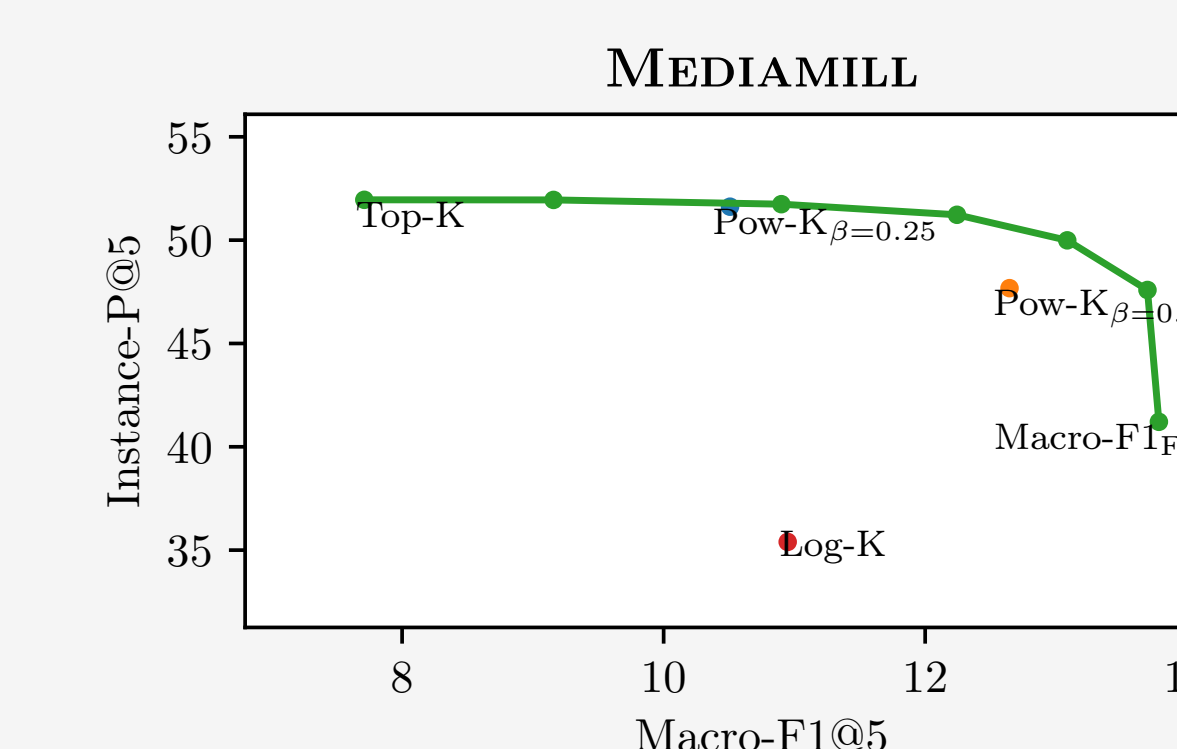
Inference strategy	Mediamill					RCV1X				
	Instance @5	Macro @5	Instance @5	Macro @5		Prec	Rec	Prec	Rec	F1
TOP-K	51.96	62.04	12.85	8.75	7.71	52.30	81.96	12.77	7.61	7.64
TOP-K+ $w^{\text{POW}}$	47.68	56.62	13.00	17.37	12.64	48.48	77.18	14.69	17.66	13.64
TOP-K+ $w^{\text{LOG}}$	35.40	39.96	11.38	15.33	10.95	51.18	80.49	16.03	11.75	11.29
MACRO-P <sub>FW</sub>	6.99	8.96	17.29	8.79	3.17	29.40	49.81	21.69	5.72	5.31
MACRO-R <sub>PRIOR</sub>	7.38	7.25	8.91	26.50	6.71	34.77	56.28	13.13	24.59	12.77
MACRO-R <sub>FW</sub>	7.38	7.25	8.91	26.50	6.71	34.15	55.24	13.15	24.89	12.73
MACRO-F1 <sub>FW</sub>	43.57	51.60	15.20	15.05	13.82	44.42	71.86	21.96	12.25	13.68

## Interpolated metrics

Optimizing macro-measures incurs a **significant drop** in instance-wise measures.

To achieve the desired trade-off between tail and head label performance, a straight-forward **combination** between standard instance-wise precision@ $k$ , and a macro-average @ $k$  metric can be used, e.g.:

$$\Psi(\mathbf{C}) = (1 - \lambda)\Psi_{\text{Instance-Precision@}k}(\mathbf{C}) + \lambda\Psi_{\text{Macro-F1@}k}(\mathbf{C})$$



A combination of a macro-metric and instance precision-at- $k$  can achieve good results on both metrics **simultaneously!**