1 Metric for Evaluating Data Extraction from Charts

The output and GT for each chart is a set of (*name*, *data series*) pairs. Each *name* is a string, possibly the empty string (if there is no legend to indicate names). There are different types of *data series*:

1. Continuous. This type of data series is generally represented as a line chart. The x-axis is some subset of real numbers, and the data series is represented by a sorted list of (x, y) pairs, where we assume that intermediate values can be computed by interpolation.

2. Point Set (e.g. scatter plots). A (multi)set of (x, y) pairs, but no notion of ordering. x and y are both real numbers. Duplicates are technically allowed, but would be rare, and probably not visually indicated in the chart (overlapping markers).

3. Discrete. All bar charts and some line plots (where the x-axis is discrete, including dates). A possibly ordered set of (x, y), where x is a string and y is a real number.

4. Boxplot. Represented by real numbered summary statistics: min, max, first-quartile, third-quartile, median. We are ignoring any outliers indicated.

The formulas for comparing pairs of same-type data series are distinct for each type of data series. The metric for each type has an output value in the range $[0, 1]$, which is part of the challenge of designing such metrics.

1.1 Continuous Metric

The idea behind this metric is to evaluate the difference of two functions as an integral of their point-wise differences. To get matching points between the predicted and GT functions, we use linear interpolation. So for predicted data series $P = [(x_1, y_1), \ldots, (x_N, y_N)]$ and reference data series $G = [(u_1, v_1), \ldots, (u_M, v_M)$, we can define precision and recall as

$$Recall(P, G) = \frac{1}{u_M - u_1} \sum_{i=1}^{M} (1 - Error(v_i, u_i, P)) * Interval(i, G) \tag{1}$$

$$Error(v_i, u_i, P) = \min\left(1, \left|\frac{v_i - I(P, u_i)}{v_i}\right|\right) \tag{2}$$

$$Interval(i, G) = \left\{ \begin{array}{ll} \frac{u_{i+1} - u_i}{2}, & \text{for } i = 1 \\ \frac{u_i - u_{i-1}}{2}, & \text{for } i = M \\ \frac{u_{i+1} - u_{i-1}}{2}, & \text{for } 1 < i < M \end{array} \right\} \tag{3}$$

$$Precision(P, G) = Recall(G, P) \tag{4}$$

Where $I(P, u_i)$ in Eq. 2 computes the value of P at the point u_i. Generally I will be linear interpolation/extrapolation, though higher order interpolation methods could be used. The error corresponding to any u_i is relative to the magnitude of v_i and is capped

at 1 to deal with cases where the magnitude of v_i is very small (or 0). An alternative error formulation would be

$$Error(v_i, u_i, P, \epsilon) = \min\left(1, \left|\frac{v_i - I(P, u_i)}{v_i + \epsilon}\right|\right) \tag{5}$$

where ϵ is a small hyperparameter that controls the minimum size of detectable errors. That is, absolute errors smaller than ϵ are decreasingly penalized regardless of how big or small $|v_i|$ is. We set ϵ to be the range (max minus min) of the GT data points divided by 100.

Because Eq. 1 is a weight average of errors, where the maximum individual error is 1 and where weights are proportional to the interval length and sum to 1, the maximum Recall is 1 and since it is non-negative, the minimum is 0.

Note that this is also a good metric for comparing two lines in pixel space for task 6a. Instead of using the units of the axis, each $(x_n, y_n), (u_m, v_m)$ is expressed in terms of pixels.

1.2 Point Set

For this metric, we must match the predicted points $P = \{p_i\}$ to the ground truth points $G = \{g_i\}$, where $p_i, g_i \in \mathbb{R}^2$. First, we define a capped distance function to define pairwise-wise point comparisons:

$$D(p_i, g_i) = \min\left(1, \frac{\sqrt{(g_i - p_i)V^{-1}(g_i - p_i)}}{\gamma}\right) \tag{6}$$

where V^{-1} is the inverse of the covariance matrix of G, and γ is a hyperparameter. Note that Eq. 6 corresponds to a scaled Mahalanobis distance. Scaling the distance computation by the inverse covariance matrix normalizes data series to have the same variance, and thus have more comparable distances.

However, some data series have extremely small variance (e.g. exactly 0), so in order to avoid over-penalizing such cases, we ensure that the diagonal elements of V^{-1} have a maximum value of 400 divided by the squared mean of the corresponding dimension of the GT data. This corresponds to enforcing that the maximum penalty, i.e. 1, of D can only occur if the predicted value is at least 5% different that the GT value.

Then, we create the pairwise cost matrix $\mathbf{C}$, where $\mathbf{C}_{n,m} = D(p_n, g_m)$. If $\mathbf{C}$ is not square, then it can be padded with 1s to make it square, so that each dimension is of size $K = \max(N, M)$. Padded entries correspond to points that have no matches. We then solve the job-assignment problem to find the minimum cost pairing that associates each GT point with a predicted point:

$$cost = \min_{\mathbf{X}} \sum_{i}^{K} \sum_{j}^{K} \mathbf{C}_{i,j}\mathbf{X}_{i,j} \tag{7}$$

subject to $\mathbf{X}$ being a binary assignment matrix: $\mathbf{X} \in \{0, 1\}^{N \times M}$, and $\forall i$, $\sum_k^K \mathbf{X}_{i,k} = 1$ and $\sum_k^K \mathbf{X}_{k,i} = 1$.

The value of the metric is then

$$1 - \frac{cost}{\max(N, M)} \tag{8}$$

1.3 Discrete

Given that the x's are strings, there are two cases to consider. One is when the x's in the chart are given, and the other when the x's are derived from OCR.

1.3.1 Exact Match Text

For the former case, where we can expect exact match strings, we can use string equality. We can define the distance between a predicted point, (x, y) and a ground truth point(u, v) as

$$D(x, y, u, v) = 1 - (\delta(u, x)(1 - \min\left(1, \left|\frac{v - y}{v}\right|\right))) \tag{9}$$

where δ is the Kronecker delta function (1 if equal, otherwise 0). While Eq. 9 is more complex than needed for this scenario, it allows us to apply the job assignment optimization from Eq. 7 to obtain the optimal *cost*, and the value of the metric as in Eq. 8.

Equivalently, we could just run pairwise string equality between all x's and all u's, and sum up for all matches $1 - D(x, y, u, v)$.

1.3.2 Fuzzy Text Matching

For fuzzy matching, we can just redefine D to be

$$D(x, y, u, v) = 1 - ((1 - L(u, x)^{\alpha})(1 - \min\left(1, \left|\frac{v - y}{\gamma\sigma^2}\right|\right))) \tag{10}$$

where L is the normalized edit distance between u and x, *alpha* is a hyperparameter that controls how important it is to closely match the text, σ^2 is the variance of the v values of G, and γ is a hyperparameter (default 1). $\alpha < 1$ means more sensitive and $\alpha > 1$ means less sensitive to textual differences. In the limit of $\alpha \to 0$, Eq. 10 reduces to Eq. 9.

1.4 Box Plot

Box Plot data series matching reduces to Section 1.3.1, where the predicted and GT strings are exactly *min, max, first-quartile, third-quartile, median.*

2 Total Metric

The previous sub-sections discuss how to compare two data series of the same type. This section describes how to compare two sets of (*name*, data series) pairs.

The distance between each pair can be computed as

$$D(n_1, d_1, n_2, d_2) = 1 - \max(\frac{metric(d_1, d_2)}{\beta}, ((1 - L(n_1, n_2)^{\alpha})metric(d_1, d_2))) \quad (11)$$

where *metric* is the data series metric corresponding to the data series type of d_1 and d_2. We can then apply the job assignment formulation (Eq. 7) to find the best set of corresponding pairs. We then take this optimal cost and normalize it into a metric score as in Eq. 8, except we divide by the number of predicted pairs or the number of ground truth pairs, whichever is larger.