МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ «МОСКОВСКИЙ ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ (ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ)»

ФАКУЛЬТЕТ ИННОВАЦИЙ И ВЫСОКИХ ТЕХНОЛОГИЙ
КАФЕДРА АНАЛИЗА ДАННЫХ

Выпускная квалификационная работа по направлению 01.03.02 "Прикладные математика и информатика"
На тему:

Гладкая метрика для задачи ранжирования

Студент ______________________ Левков Мирон Николаевич

Научный руководитель ______________________ Воронцов Александр Сергеевич

Зам. зав. кафедрой ______________________ Бунина Елена Игоревна

Москва, 2017

Гладкая метрика для задачи ранжирования

Рассматривается задача ранжирования документов в поисковых запросах. В работе исследованы имеющиеся варианты сглаживания метрики ранжирования. Предложены различные способы сглаживания метрики DCG. Проведен анализ зависимости качества сглаженных метрик от гиперпараметров и размера выборки поисковых запросов.

Содержание

Введение

Описание задачи ранжирования

Пусть имеется некий поисковый запрос. Задача ранжирования состоит в том, чтобы из списка всех доступных документов выбрать наиболее хорошие (релевантные) и показать их ползователю в поисковой выдаче. Для того, чтобы иметь возможность измерять качество алгоритма, выполняющего решение данной задачи, используются реальные оценки релевантности конкретных документов запросу, проставленные людьми.

Обозначения

Обозначение 1.1. Обозначим $\{d_i\}_{i=1}^n$ - набор *документов* релевантных данному поисковому запросу; $\{r_i\}_{i=1}^n$ - *рельные оценки* данных документов, проставленные ассесорами;
$\{s_i\}_{i=1}^n$ - оценки релевантности (*скоры*), выданные ранжирующим алгоритмом

Определение 1.1. Метрика качества ранжирования **DCG** определяется по формуле:

$$DCG = \sum_{i=1}^{k} \frac{r_{p_i}}{discount(i)}$$

где
$p_1, ..., p_n$ - перестановка на множестве $\{1, ..., n\}$, т.ч. $s_{p_1} > s_{p_2} > ... > s_{p_n}$
k - количество документов, по которым считается метрика $(k \leq n)$ - важен в том случае, когда нас интересуют лишь несколько первых документов в

выдаче

discount(i) - дисконтирующий фактор, как правило $\frac{1}{i}$

Постановка задачи

Традиционные метрики ранжирования имеют конструкцию похожую на DCG: для подсчета метрики считается сумма по всем документам, *порядок* слагаемых в которой *зависит* от скоров, выданных ранжирующим алгоритмом, а *значения - нет*.

Несложно заметить, что для такой конструции при фиксированном наборе документов и их оценок данная метрика может принимать лишь конечное количество различных значений. При этом изменение значения метрики происходит лишь в случае перестановки местами двух документов в выдаче.

Как было замечено выше, основная проблема подобных метрик в том, что они не имеют гладкой зависимости от скоров. В то же самое время логично ожидать, что, если ранжирующий алгоритм выдал трем документам оценки $\{100, 1, 0.5\}$, а другим трем документам - $\{10, 1, 0.5\}$, то он считает первый документ из первой тройки сильно лучшим, чем первый документ из второй тройки. В добавок, гладкая относительно скоров метрика является более чувствительным инструментом для оценивания качества алгоритма и отслеживания случаев недообучения или переобучения.

В данной работе исследуются разные подходы к построению метрик ранжирования, с целью получения метрики, которая бы удовлетворяла ряду свойств, вводимых далее

Метрики качества результатов

Хотелось бы, чтобы наша метрика обладала двумя свойствами. Во-первых, метрика должна быть "реалистичной". Т.е. ее локальные минимумы/максимумы должны соответствовать минимумам/максимумам DCG.

Определение 3.1. Пусть $\{y_i\}_{i=1}^{k}$ - значения метрики DCG на данном пуле запросов для k различных значений набора гиперпараметров. Пусть $\{\hat{y}_i\}_{i=1}^{k}$ - значения нашей метрики на том же пуле при тех же наборах гиперпараметров. Тогда **качеством аппроксимации** метрики DCG нашей метрикой назовем величину $\min\limits_{\alpha,\beta}\frac{1}{k}\sum\limits_{i=1}^{k}(y_i - \hat{y}_i)^2$

Обозначение 3.1. Далее будем обозначать *качество аппроксимации*, как approx(Metric Name)

Также придуманная метрика должна быть гладкой. При этом необходим способ подсчета гладкости конкретной функции, если она задана не аналитически, а численно. Определим метрик гладкости.

Определение 3.2. Пусть $\{y_i\}_{i=1}^{k}$ - значения нашей метрики на данном пуле запросов для k различных значений набора гиперпараметров. Тогда **гладкостью** функции, посчитанной **через усреднение модуля разности** в соседних точках, будем называть величину

$$\mathrm{smooth}_{\mathrm{abs}}(y_1, ..., y_k) = \sum_{i=2}^{k} |y_i - y_{i-1}| \cdot \frac{1}{|y_k - y_1|}$$

Определение 3.3. Пусть $\{y_i\}_{i=1}^{k}$ - значения нашей метрики на данном пуле запросов для k различных значений набора гиперпараметров. Тогда

гладкостью функции, посчитанной **через нормировку дисперсии** разности в соседних точках, будем называть величину

$$\mathrm{smooth_{std}}(y_1, ..., y_k) = \frac{\mathrm{diff}_{std}}{|\mathrm{diff}_{mean}|}$$

Здесь $\mathrm{diff}_{mean} = \frac{1}{k-1}\sum_{i=2}^{k}(y_i - y_{i-1}), \quad \mathrm{diff}_{std} = \frac{1}{k-1}\sum_{i=2}^{k}(y_i - y_{i-1} - \mathrm{diff}_{mean})^2$

Определение 3.4. Пусть $\{y_i\}_{i=1}^{k}$ - значения нашей метрики на данном пуле запросов для k различных значений набора гиперпараметров. Пусть $window \in \mathbb{Z}, deg \in \mathbb{N}$. Тогда **гладкостью** функции, посчитанной **с помощью аппроксимации полиномами**, будем называть величину

(1)

$$\mathrm{smooth_{poly}}(y_1, ..., y_k) = \sum_{i=1}^{k-window+1} \frac{\left(poly_{deg}(y_i, ..., y_{i+window-1})_{i+\lfloor\frac{window}{2}\rfloor} - y_{i+\lfloor\frac{window}{2}\rfloor}\right)^2}{k - window}$$

Здесь $poly_{deg}(y_i, ..., y_{i+window-1})$ - аппроксимирующий полином степени deg, построенный по window точкам. Т.е. такой полином стпепени deg, что среднеквадратичное отклонение в точках $\{i, ..., i + window - 1\}$ минимально.

В терминах данных нами определений, новая метрика тем лучше, чем меньше такие показатели, как гладкость и аппроксимация.

Возникает ряд проблем с тем, что выбор конкретного способа измерения гладкости для сравнения наших метрик между собой неочевиден. Первая метрика хорошо отображает гладкость в том случае, если функция монотонна (если не учитывать шумовые колебания) - однако же в иных случаях данная метрика может быть плоха из-за нормировки на разность значений в крайних точках.

Вторая метрика ведет себя лучше, однако тоже не является достаточно гибкой и интерпритируемой.

С последней метрикой встает проблема выбора параметров deg и window. Тем не менее эта метрика понятна и действительно способона достаточно хорошо отображать гладкость функции. После перебора разных вариантов выбор был сделан в пользу параметров $deg = 3$, $window = 11$. При этом стоит учитывать, что гладкость метрики ранжирования измерялась на множествах размера порядка 100 точек.

Метрика SoftDCG

Рассмотрим метрику SoftDCG. Данная метрика является попыткой сглаживания метрики DCG. Кратко разберем способ подсчета SoftDCG

Пусть наш набор документов в поисковом запросе уже упорядочен по скорам. Рассмотрим кокретный документ d_j со скором s_j. Логично предположить, что скор не является точным определением того, на каком месте должен быть документ. Более точно: пусть $s_j - s_{j+1} < \varepsilon, s_{j-1} - s_j > \varepsilon$, тогда, если добавить к оценками ранжирующего алгоритма случайный шум с дисперсией ε, документы s_j, s_{j+1} поменяются местами в выбаче более вероятно, чем s_{j-1}, s_j. Данный факт говорит о близости документов, однако DCG способен учитывать близость документов с точностью до 1 места.

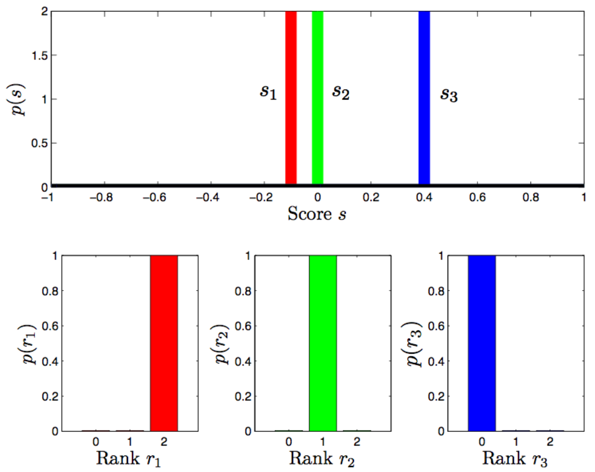


Figure 1: Deterministic scores and ranks: *Three document scores as point (deterministic) values and their corresponding rank distributions. The lowest scoring document s_1 is certain to be ranked in the lowest position 2.*

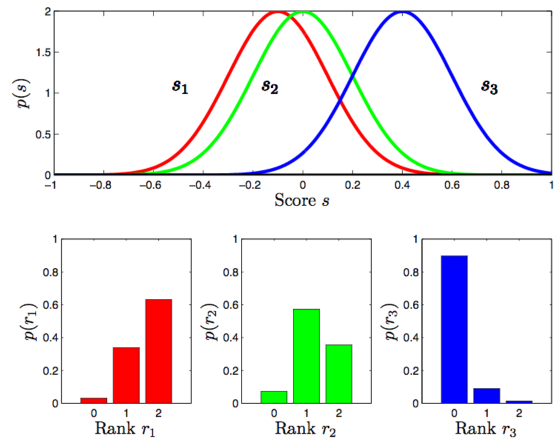


Figure 2: From score to rank distributions: *Smoothed scores for 3 documents and the resulting 3 rank distributions.*

Метрика SoftDCG предполагает слудующее: вместо того, чтобы каждому документу сопоставлять конкретное место в поисковой выдаче, можно сопоставить ему нормальное распределение на местах. Параметры этого распределения подбираются по формулам исходя из скоров документов.

Далее, зная распределение на местах, мы можем посчитать для каждого документа

$$\hat{d}(j) = \mathrm{E}\ \mathrm{discount}(d_j) - \text{средний дискаунт j-го документа}$$

.

Далее эти средние дискаунты используются для подсчета SoftDCG:

$$\mathrm{SoftDCG} = \sum_{i=1}^{n} \hat{d}_j \cdot r_j$$

Метрика NoisedSoftDCG

Рассмотрим другую метрику (назовем ее NoisedSoftDCG). Идея данной метрики берет свое начало в статье про SoftDCG, однако способ подсчета данной метрики несколько иной.

В SoftDCG каждому документу сопостовлялось некоторое вероятностное распределение на множестве возможных занимаемых позиций. При этом следует отметить, что данное распределение вполне четко задавалось аналитически. Благодаря этому имелась возможность задать такой параметр, как $\hat{d}(j)$ (средний дискаунт), формулой.

В данном же случае при подсчете метрики принимается во внимание лишь тот факт, что при небольшом изменении скоров на случайные величины значение метрики может меняться в силу того, что близкие документы будут переставляться местами.

Алгоритм подсчета NoisedSoftDCG следующий:

(1) Получим скоры ранжирующего алгоритма $s_1, ..., s_n$

(2) Сгенерируем случайный шум $\xi_1, ..., \xi_n$

(3) $\hat{s}_1 = s_1 + \xi_1, ..., \hat{s}_n = s_n + \xi_n$

(4) Посчитаем значение метрики $Value_{DCG}$

(5) Повторим шаги (2)-(4) достаточно большое количество раз (T) и вычислим $NoisedDCG = \sum\limits_{i=1}^{T} Value^{i}_{DCG}$

Можно дать некую интуицию по поводу того, почему метрика считается именно так. При достаточно большом T близкие документы будут часто меняться местами, а усреднение DCG при этом даст некий аналог среднего дисконта. Таким образом ожидается, что при больших T данная метрика будет вести себя хорошо в плане гладкости

Метрика FairSoftDCG

Еще один вариант метрики, идея которой схожа с NoisedDCG. Заметим, что, основываясь на скорах, выданных ранжирующим алгоритмом, можно ввести вероятностное распределение на перестановках всех документов.

При этом распределение вводится так, что $P(d_i \text{ выше } d_j) \sim e^{\sigma(s_j - s_i)}$ (данное распределение можно ввести единственным образом). Благодаря этому для каждой перестановки на документах $p_1, ..., p_n$ можно посчитать ее вероятность:

$$P(p_1, ..., p_n) = \prod_{i=1}^{n-1} \frac{e^{\sigma s_{p_i}}}{\sum\limits_{k=j}^{n} e^{\sigma s_{p_j}}}$$

Теперь перебрав все перестановки документов можно посчитать "честное"мат. ожидание DCG. Собственно это у будет значением FairSoft-DCG.

$$FairSoftDCG = \sum_{p_1, ..., p_n} P(p_1, ..., p_n) \cdot DCG(p_1, ..., p_n)$$

При этом возникает ряд проблем с данной метрикой, т.к. вычислительно данная задача достаточно сложная (всего имеется $n!$ возможных перестановок). В связи с этим был выбран способ подсчета близкого к данной метрике значения - подсчета по топ-k документам:

$$FairSoftDCG = \sum_{p_1, ..., p_k} P(p_1, ..., p_k) \cdot DCG(p_1, ..., p_k),$$

где сумма берется по всем возможным размещениям по k из n элементов

Применение результатов

В данной части будут приведены предполагаемые способы использования искомой метрики. На данный момент идеальной - т.е. гладкой и хорошо аппроксимирующей DCG - метрики не найдено. Но следует учитывать, что любое небольшое улучшение метрики в смысле этих двух показателей является результатом.

Итак. Первый и способ применения более гладкой метрики - это использование ее в качестве инструмента показывающего качество обучения модели. Благодаря большей чувствительности появляется возможность следить за тем, как малые изменения гиперпараметров влияют на качество модели.

Простой пример: благодаря этой метрики можно достаточно хорошо измерять влияние новых элементов ансамбля в моделях градиентного бустинга. Точно так же можно использовать данную метрику для active learning like задач. Благодаря высокой чувствительности можно будет немного дообучать алгоритм на небольших частях датасета и смотреть на изменение качества ранжирования.

Второй и, пожалуй, самый важный с практической точки зрения способ применения нашей метрики - смешивание нескольких моделей ранжирования. В наши дни используются в ранжирующих системах крупных компаний используются абсолютно разные по характеру формулы. Эти формулы имеют разный масштаб выдаваемых значений. Каждая из этих формул придумывалась для оптимизации конкретной метрики - причем метрики могут быть кардинально различными по смыслу. При этом хотелось бы уметь смешивать формулы, выданные несколькими различными моделями, в один ансамбль формул, получая таким образом более мощную формулу ранжирования.

Данная проблема является достаточно трудоемкой и в терминах задачи ранжирования не решена. Т.к. наша метрика является гладкой и хорошо аппроксимирует DCG, оптимальная комбинация формул согласно этой метрике должна быть в той же окрестности, что и оптимум с точки зрения DCG. В то же самое время логично ожидать, что при взятии выпуклой комбинации двух формул должна получаться некая выпуклая гладкая кривая для значений метрики.

В качестве тривиального применения - добавления шума к скорам ранжирующего алгоритма. Здесь имеется ввиду следующее: если добавить к скорам модели шум с неким коэффициентом, это эквивалентно смешиванию двух моделей - нормальной и абсолютно бесполезной. Логично полагать, что оптимум качества в данном случае будет при коэффициенте, с которым берется шум, равном 0 (или близком к 0). Данный способ применения гладкой метрики - своеобразная sanity-check.

Эксперименты

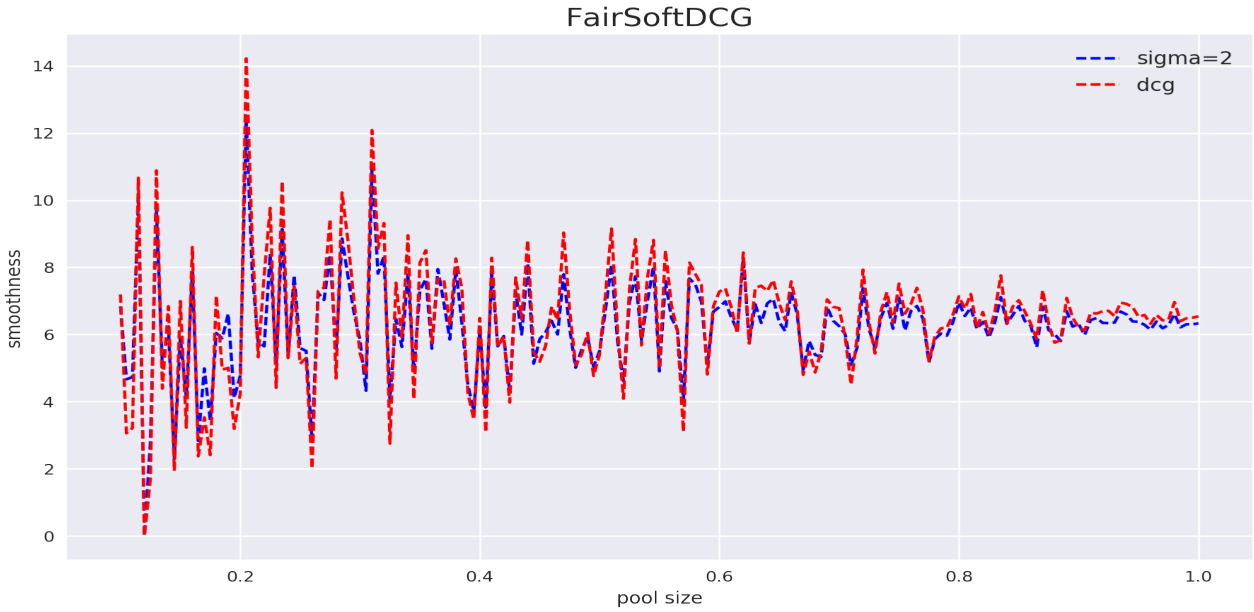
Зависимость гладкости от размера пула запросов

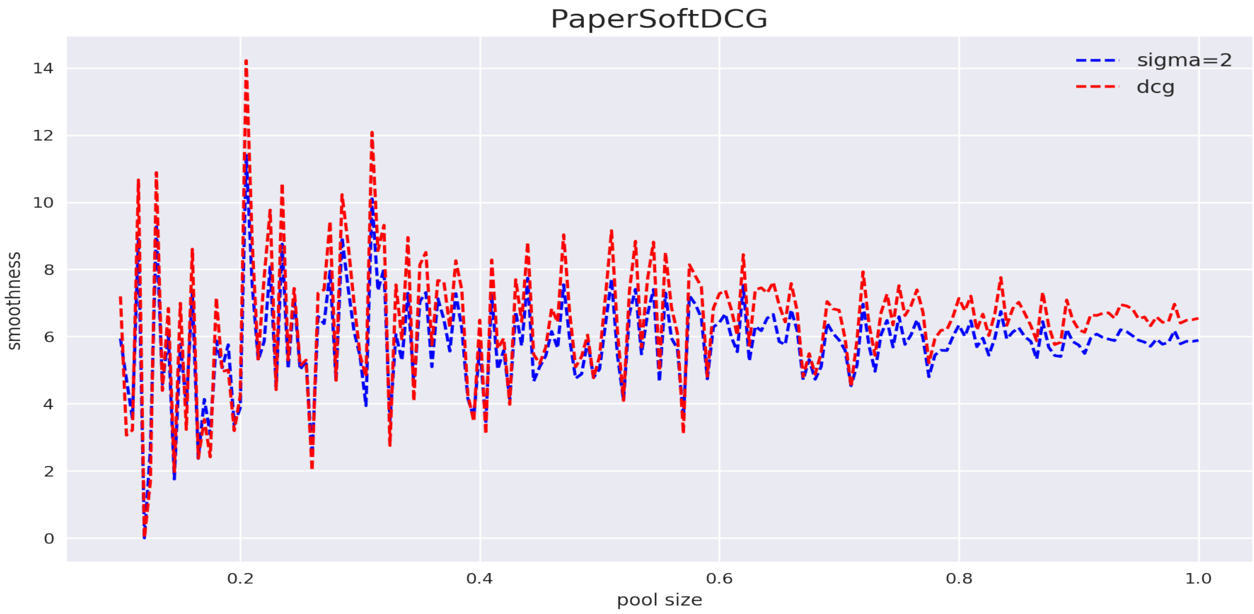
Первый эксперимент является одновременно способом проверить наш метод измерения гладкости на адекватность и способом сравнения между собой расмотренных ранее метрик ранжирования. Здесь и далее рассуждения будут вестись для метрики DCG, но они точно так же верны для любой метрики из списка SoftDCG, NoisedSoftDCG, FairSoftDCG.

Для начала вспомним, что такое в нашей задаче "размер пула". Размер пула - это количество обрабатываемых поисковых запросов. Для каждого запроса мы считаем значение DCG, а потом производим усреднение данного значения по всем запросам. Все это происходит при каком-то зафиксированном наборе гиперпараметров.

Таким образом для каждого значения набора мы получаем какое-то значение метрики. Например, можно смотреть на значение метрики для ранжирующиго алгоритма, который является выпуклой комбинацие двух других алгоритмов, в зависимости от коэффициента, с которым алгоритмы смешиваются. Если рассматривать данный график, на множесте коэффициентов $[0;1]$, можно посчитать его гладкость. При этом, рост размера пула - т.е. увеличение количества запросов, по которому мы усредняем ответ, должно играть роль сглаживающего фактора.

Данный эксперимент состоял в том, чтобы посмотреть на изменение гладкости с увеличением роста пула. Хочется заметить, что т.к. увеличивать пул мы возможности не имели, делалось эквивалентное действие - его


FairSoftDCG
sigma=2
dcg
smoothness
pool size
0
2
4
6
8
10
12
14
0.2
0.4
0.6
0.8
1.0


PaperSoftDCG
sigma=2
dcg
smoothness
pool size
0
2
4
6
8
10
12
14
0.2
0.4
0.6
0.8
1.0

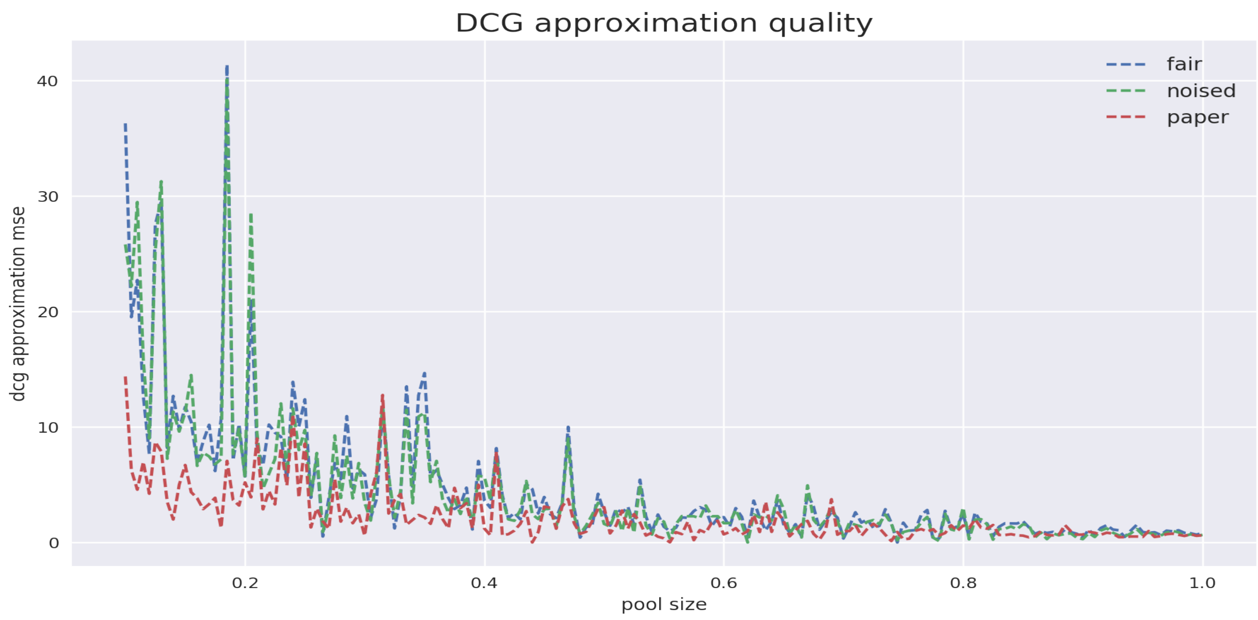
уменьшение. Ожидалось, что для каждой метрики будет получен график, имеющий обратную пропорциональность - чем больше размер пула, тем меньше гладкость. Следует также напомнить, что в наших терминах гладкость кривой - тот параметр который хотелось бы уменьшить.

На графиках для сравнения приведена гладкость DCG и гладкость соответствующих метрик. Для наглядности значения метрик брались при одном и том же значении гипарпараметра $\sigma = 2$. Можно заметить, что самой гладкой в нашей терминологии являлась метрика NoisedSoftDCG. При совсем маленьких размерах пула (0.6 от изначального размера и меньше) графики, как и ожидалось, являются слишком шумными.

Зависимость качества аппроксимации от размера пула запросов

Аналогично предыдущему эксперименту логично было бы ожидать графики, на которых с увеличением размера пула аппроксимация DCG уменьшается.

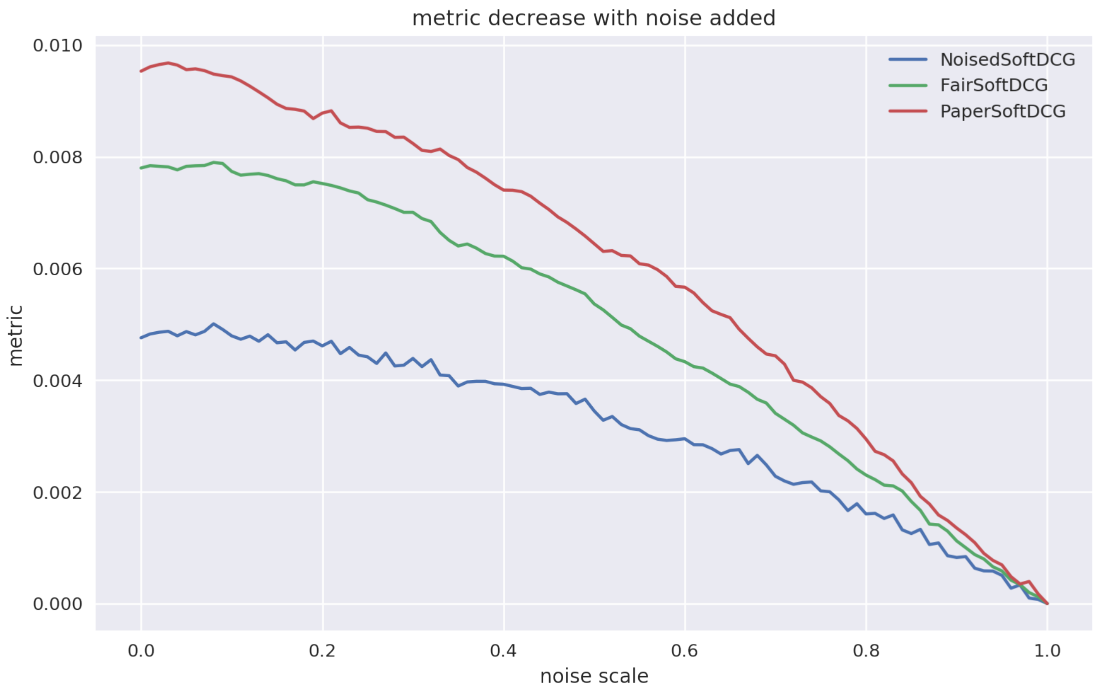
Аппроксимация в нашей нотации имеет тот же смысл, что и гладкость: меньшие значения соответствуют более близкой к DCG метрике.



Что же касается аппроксимации DCG, можно заметить, что график для NoisedSofDCG является наиболее шумным, а FairSoftDCG - наименее.

Изменение метрики при добавлении шума

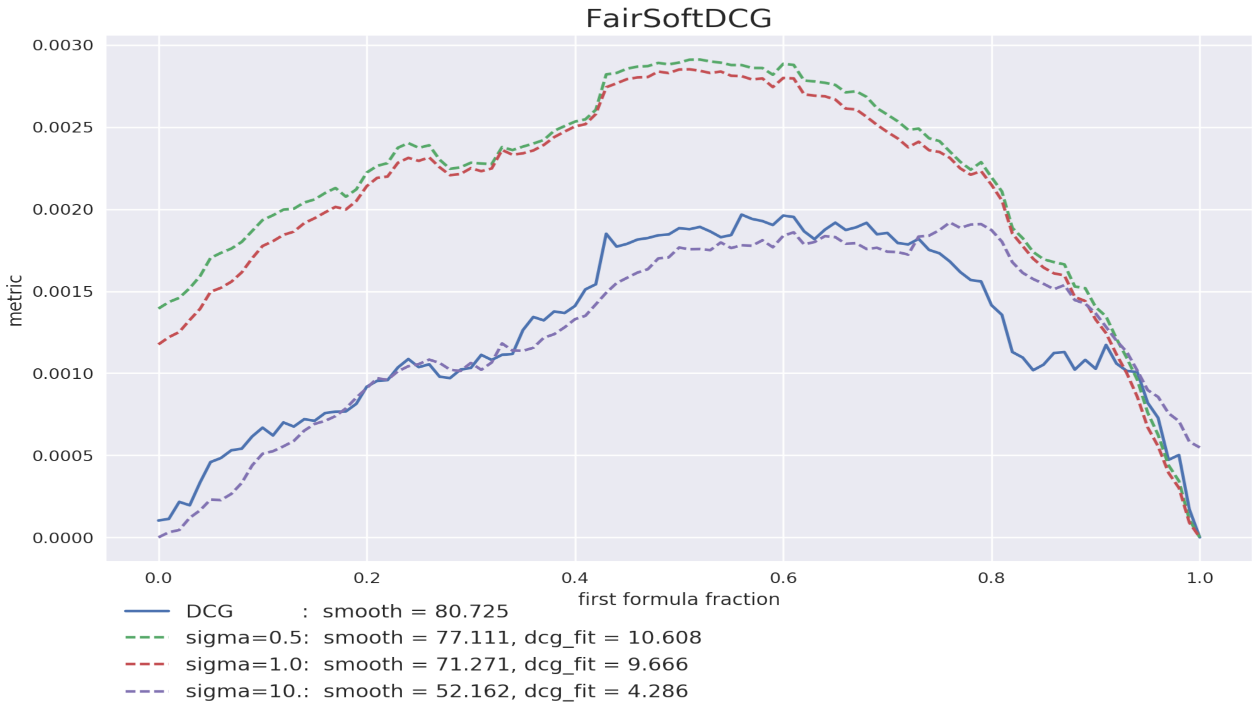
Как упоминалось ранее, первоочередным действием в работе было желание убедиться в том, что метрика гладкости действительно несет ту смысловую нагрузку, которая на нее была возложена. Для этого проводился эксперимент по смешиванию двух ранжирующих формул, одна из которых была настроящей, а вторая - просто сгенерированным случайным шумом.



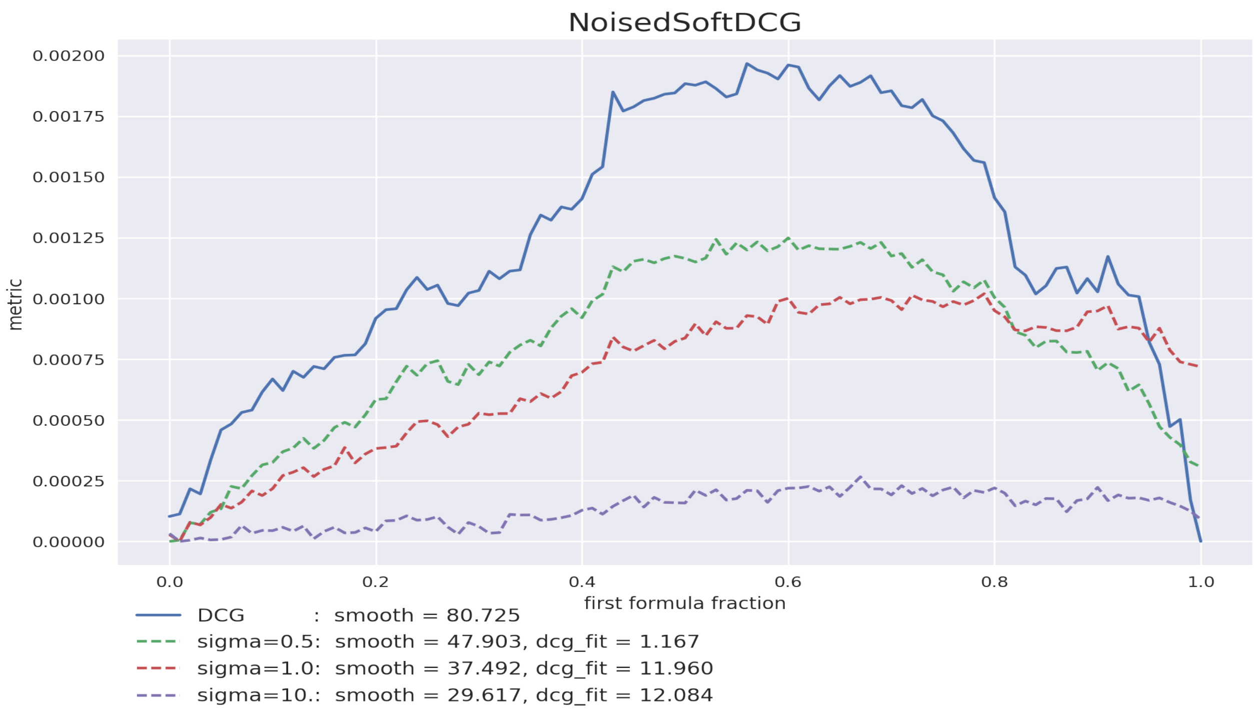
По оси x отложен коэффициент, с которым прибавляется случайный шум, по оси y - значения метрики. Можно заметить, что наши метрики убывают (т.е. они логичны) причем визуально видно, что они делают это почти монотонно.

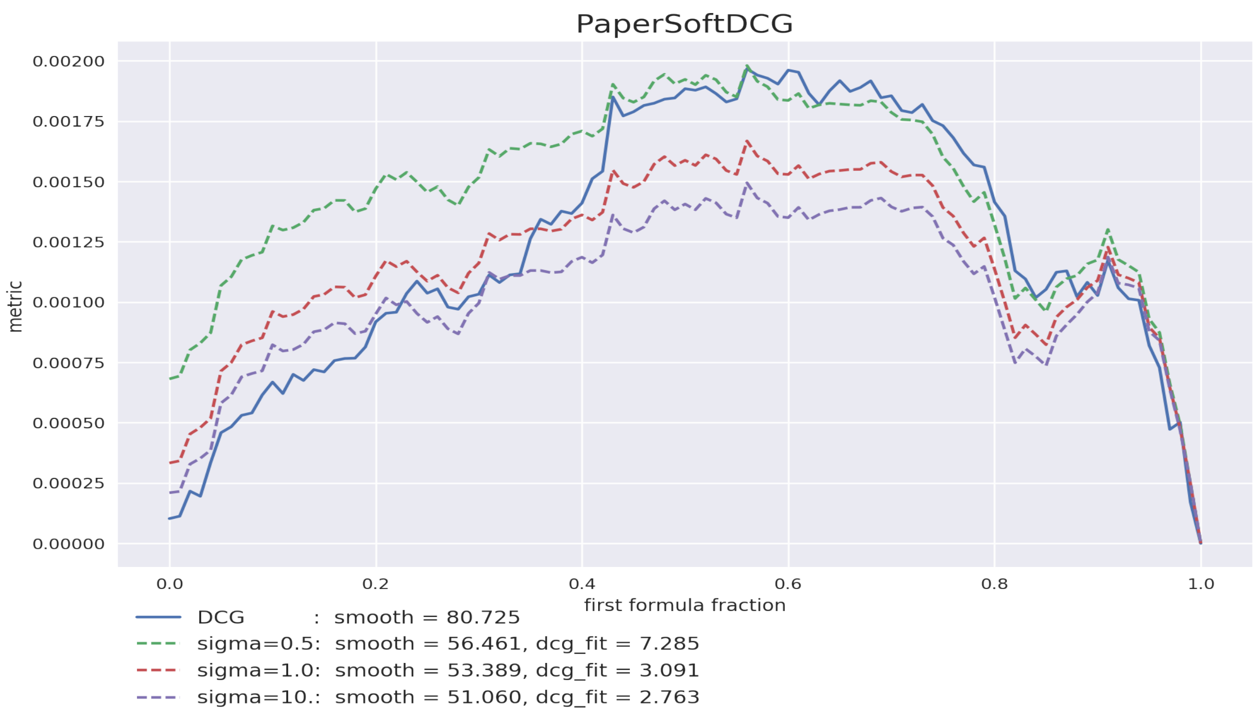
Подобное поведение в данном эксперименте является необходимым условием для класса искомых метрик.

Изменение метрики для комбинации двух формул



Самый важный с точки зрения пользы эксперимент - эксперимент по смешиванию двух формул ранжирования. Цель эксперимента - проверить, какая из наших метрик лучше отображает реальный оптимум качества при использовании в качестве ранжирующей формулы выпуклую комбинацию двух формул.

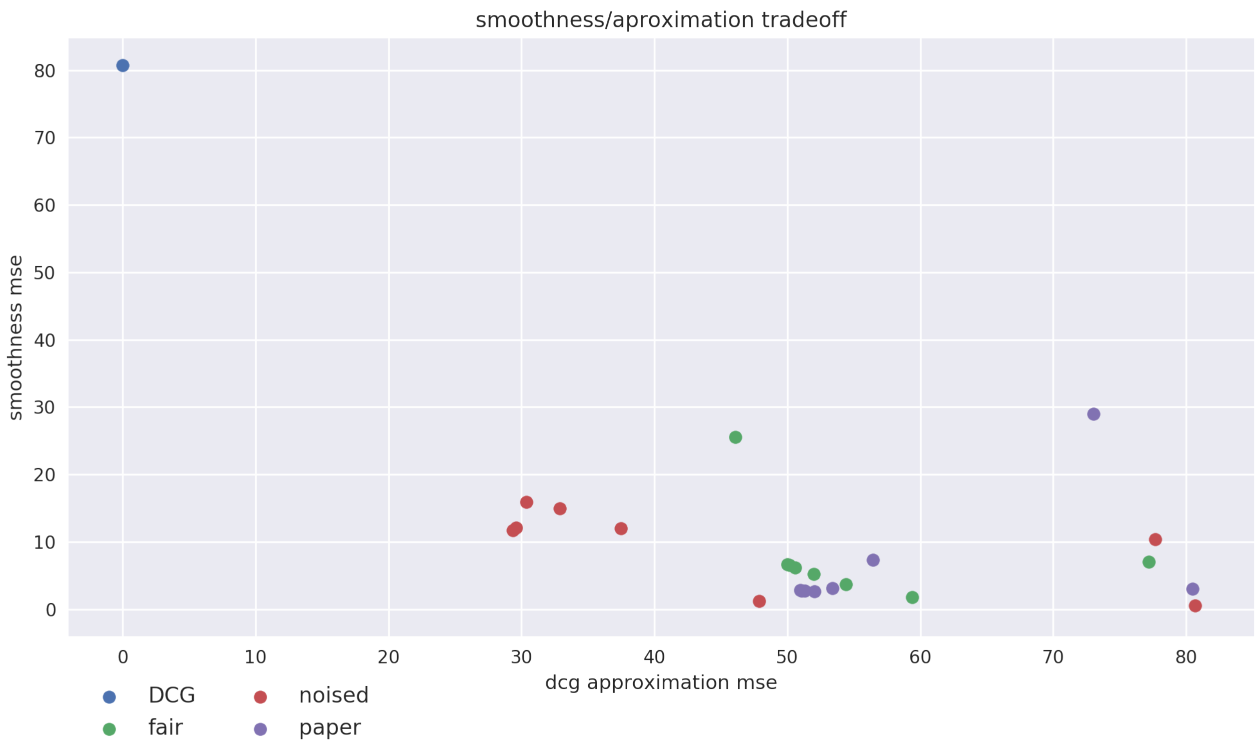

NoisedSoftDCG
metric
0.00000
0.00025
0.00050
0.00075
0.00100
0.00125
0.00150
0.00175
0.00200
0.0
0.2
0.4
0.6
0.8
1.0
first formula fraction
DCG : smooth = 80.725
sigma=0.5: smooth = 47.903, dcg_fit = 1.167
sigma=1.0: smooth = 37.492, dcg_fit = 11.960
sigma=10.: smooth = 29.617, dcg_fit = 12.084


PaperSoftDCG
metric
0.00000
0.00025
0.00050
0.00075
0.00100
0.00125
0.00150
0.00175
0.00200
0.0
0.2
0.4
0.6
0.8
1.0
first formula fraction
DCG : smooth = 80.725
sigma=0.5: smooth = 56.461, dcg_fit = 7.285
sigma=1.0: smooth = 53.389, dcg_fit = 3.091
sigma=10.: smooth = 51.060, dcg_fit = 2.763

Проблема выбора гиперпараметра

Так как главной целью работы было получение гладкой метрики, с помощью которой можно будет удобно и правильно подбирать коэффициенты в смеси ранжирующих формул, остановимся на этом эксперименте.

Как можно было заметить из графиков, приведенных выше, при разных значениях гиперпараметра σ получаются разные качества аппроксимации/гладкости. При этом, логично предположить следующее: чем лучше аппроксимация (т.е. чем ближе σ к 0), тем хуже гладкость кривой. Ниже приведен график в осях, соответствующих данным двум показателям, метрик с разными гиперпараметрами.



Как видно из графика, для каждой метрики в можно варьировать этот гиперпараметр, получая тем самым желаемый компромисс между гладкостью метрики и хорошей аппроксимацией DCG.

Следует отметить, что выбор параметра - неочевидная задача. Т.к. для конкретной смеси формул получается свое значение параметра. В итоге, возникает проблема определения σ. Она заключается в том, что необходимо выбирать данный параметр, опираясь, лишь на скоры, выданные каждой из формул. Однако скоры в каждом запросе свои. В добавок функция, которую мы при этом пытаемся минимизировать: $\alpha \cdot approx + \beta \cdot smoothness$, не является гладкой относительно σ.

Есть некоторые эвристики по подбору σ, но хорошего решения в ходе исследования данного вопроса найдено не было.

Выводы

В ходе исследования была рассмотрена реализация метрики SoftDCG, идея которой описана в другой работе. Также были рассмотрены свои метрики NoisedSoftDCG и FairSoftDCG.

При изучении этих метрик были введены такие параметры, как "гладкость"и "аппроксимация DCG". Было проведено исследование касательно изменения этих показателей при изменении различных гиперпараметров. При этом не возникало противоречий с высказаными заранее предположениями, основанными на логике. Это дало возможность понять, что представленные метрики в целом являются неким продвижением в нужном направлении - но окончательного решения не дают.

В конце была освещена проблема выбора гиперпараметра σ, который соответствует тому, насколько сильно наша метрика стремится раздвинуть между собой документы (фактически - порядок разницы между скорами близких документов).

Список литературы

[1] M. Taylor, J. Guiver, S. Robertson and T. Minka. SoftRank: Optimising Non-Smooth Rank Metrics. Microsoft Research Cambridge, 2016

[2] Christopher J.C. Burges. From RankNet to LambdaRank to LambdaMART. Microsoft Research Technical Report, 2010