AI-503 Advanced Machine Learning

Up until now…

- Classification
  - Predict discrete classes
- Regression
  - Predict continuous targets
- Feature Extraction/ Selection
  - Dimensionality reduction
- Clustering
  - Group data into clusters

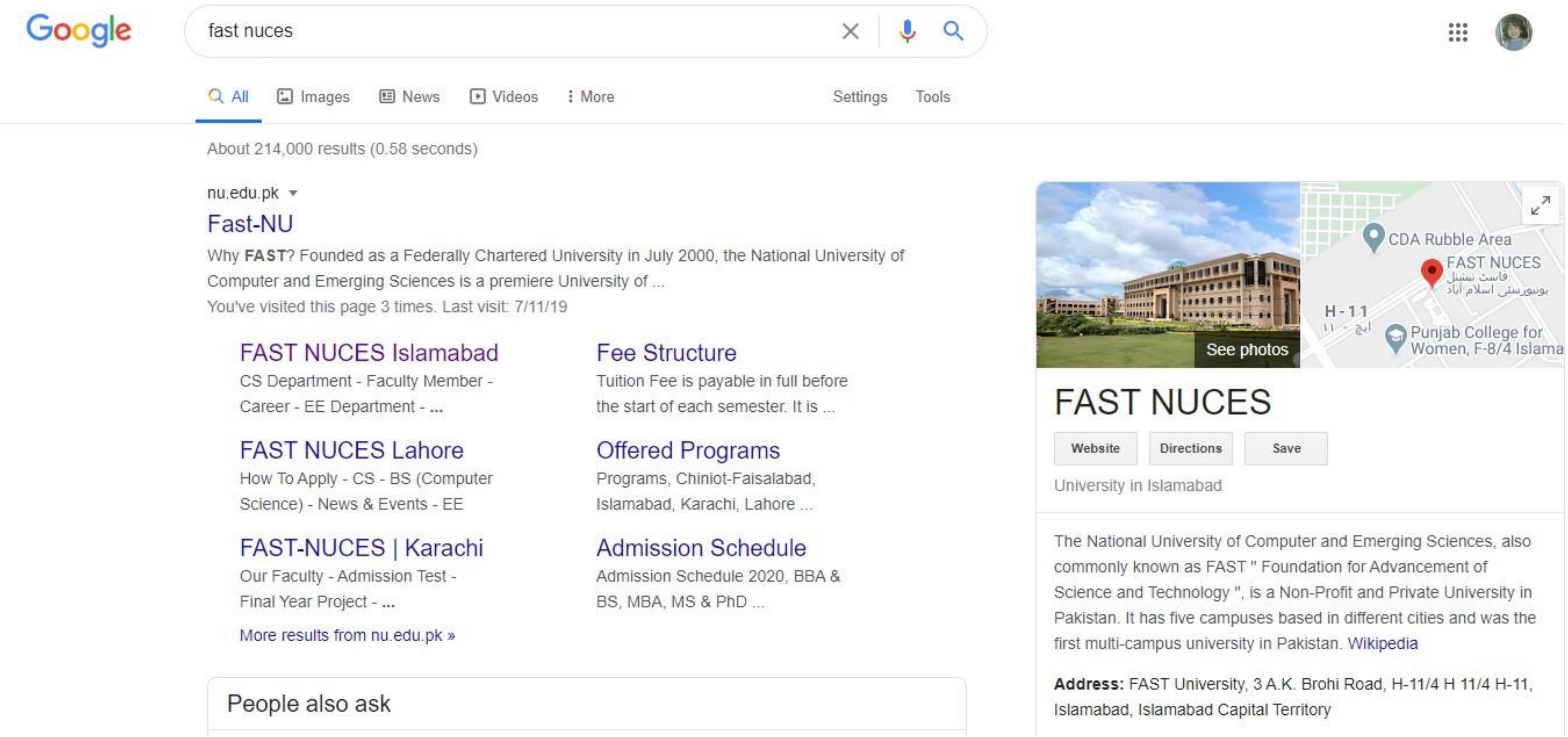
Ranking

- Rank a set of examples
  - The example with a higher true rank produces a higher score as compared to the other examples
- Examples…

Recommendation systems

- Netflix
- YouTube
- Amazon
- Facebook
- Twitter
- Journal recommendations
- …

Information retrieval



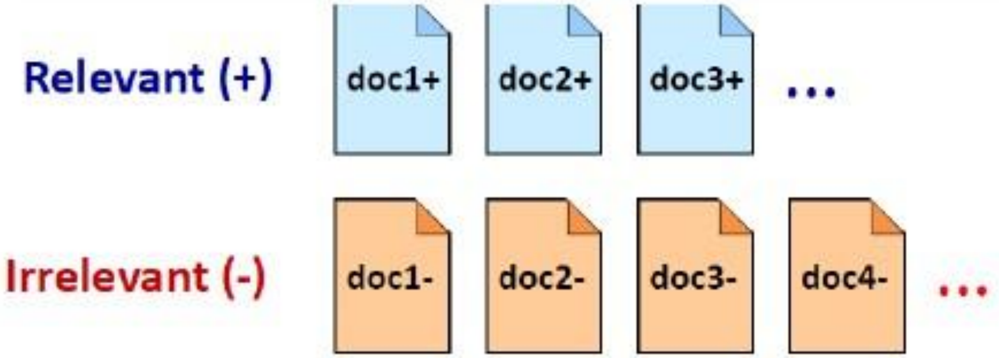
Drug discovery

- Rank the given set of drugs in the decreasing order of their potency against a disease
- Test the most promising ones in the lab

So how to develop models than can rank?

- SVMs
- Neural Networks
- Tree based methods
- …

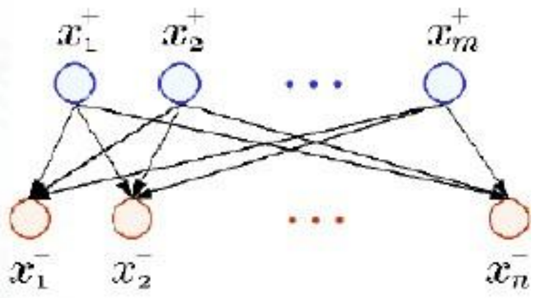
Bipartite ranking



- Instance space: X
- Input: $S = \{S_+, S_-\}$
  - $S_+ = \{x_1^+, \ldots, x_m^+\}$ (positive examples)
  - $S_- = \{x_1^-, \ldots, x_n^-\}$ (negative examples)
- Output: Ranking function $f: X \to \mathbb{R}$

Bipartite ranking vs classification & regression

- In classification we require
  - $f(x) > 0$ for positive examples
  - $f(x) < 0$ for negative examples



- In regression
  - $f(x)$ should be as close as possible to the corresponding output
- The ranking function should be such that
  - $f(x) > f(x')$ if instance x should rank higher than x'
    - Any given positive instance should rank higher than any negative instance

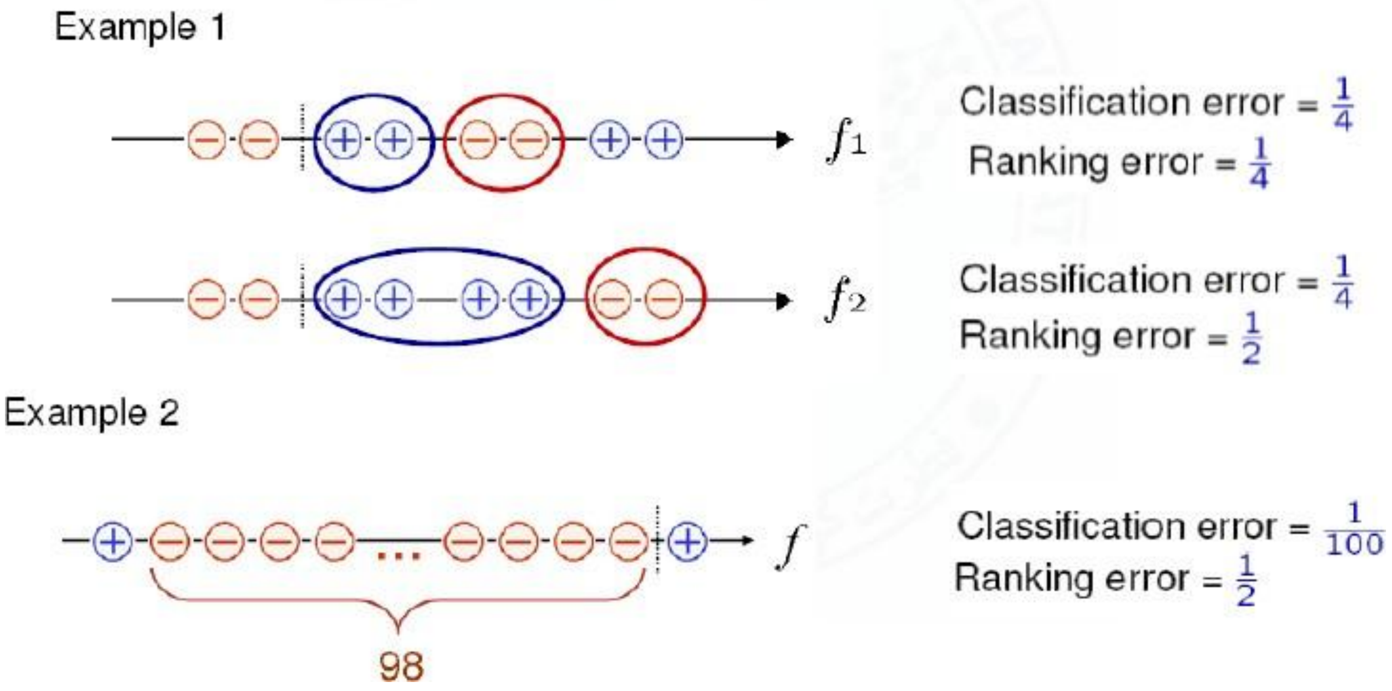
Definition of error in ranking

- If we have m positive instances and n negative instances, the number of possible 'pairings' becomes mn
  - Thus, the error for a given ranking function will be given by:

$$\widehat{\mathbf{er}}_S(f) = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} \mathbf{1}(f(x_i^+) < f(x_j^-))$$

Ranking error

- What will be the classification and ranking errors of the following ranking functions?



Ranking with SRM

- SRM requires

$$f^* = argmin_f \{\underbrace{L(X,Y;f) + \lambda g(\|f\|)}_{\text{Structural Risk}}\}$$

Regularization
Classifier Complexity
(smoothing) term

Empirical Loss (or risk)
term

X, Y is the training data
f is the learning function

- For ranking, we know the empirical error can be given by

$$\widehat{\mathbf{er}}_S(f) = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} \mathbf{1}(f(x_i^+) < f(x_j^-))$$

Ranking with SRM

- Find me a ranking function that minimizes the empirical error, possibly with some regularization, over some class of ranking functions
  - Examples: $f(x) = < w, x >$
- Mathematically

$$\min_{f \in \mathcal{F}} \left[\frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} \ell(f, x_i^+, x_j^-) + \lambda N(f) \right]$$

$\ell(f, x_i^+, x_j^-)$: convex upper bound on $\mathbf{1}(f(x_i^+) < f(x_j^-))$
$N(f)$: regularizer
$\lambda > 0$: regularization parameter
$\mathcal{F}$: class of ranking functions

Ranking with SRM

- What should be loss function?
  - The ideal loss function would be:
    - Zero-one loss $\mathbf{1}(f(x_i^+) < f(x_j^-))$
      - Incurs a loss of 1 only if the ranking is not proper
      - Otherwise the loss is zero
      - Problems?
        - » Non-convex
        - » Discontinous
    - Hinge-Loss

$$\ell_{\text{hinge}}(f, x_i^+, x_j^-) = \left(1 - \left(f(x_i^+) - f(x_j^-)\right)\right)_+ \quad \left[u_+ = \max(u, 0)\right]$$

Bipartite Rank-SVM

$$min_{w,\xi>0}\ \lambda\|w\|^2 + \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\xi_{ij}$$

Such that, for all i= 1…m and j = 1…n

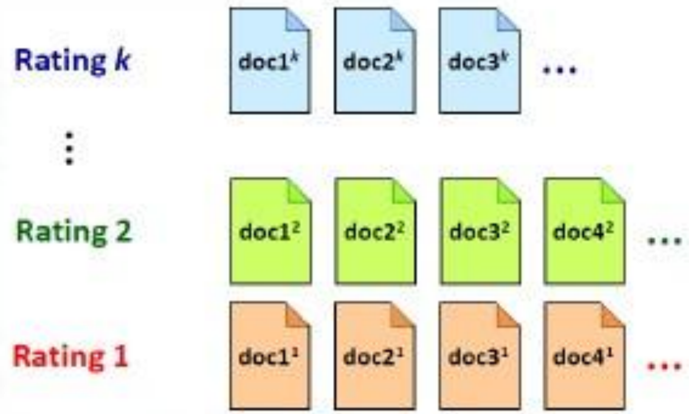
$$f(x_i) \geq f(x_j) + 1 - \xi_{ij}$$

K-partite ranking

Input: Training sample $S = (S_1, S_2, \ldots, S_k)$:

$$S_k = (x_1^k, \ldots, x_{n_k}^k) \in X^{n_k} \quad \text{(examples of rating } k\text{)}$$
$$\vdots$$
$$S_2 = (x_1^2, \ldots, x_{n_2}^2) \in X^{n_2} \quad \text{(examples of rating 2)}$$
$$S_1 = (x_1^1, \ldots, x_{n_1}^1) \in X^{n_1} \quad \text{(examples of rating 1)}$$

Output: Ranking function $f : X \to \mathbb{R}$



Empirical error:

$$\widehat{\mathbf{er}}_S(f) = \left(\frac{1}{\sum_{1 \le a < b \le k} n_a n_b} \right) \sum_{1 \le a < b \le k} \sum_{i=1}^{n_b} \sum_{j=1}^{n_a} (b-a) \, \mathbf{1}(f(x_i^b) < f(x_j^a))$$

K-partite Ranking with SRM

Minimize a convex upper bound on the empirical ranking error, possibly with some regularization, over some class of ranking functions:

$$\min_{f \in \mathcal{F}} \left[\left(\frac{1}{\sum_{1 \le a < b \le k} n_a n_b} \right) \sum_{1 \le a < b \le k} \sum_{i=1}^{n_b} \sum_{j=1}^{n_a} \ell(f, x_i^b, x_j^a, (b-a)) + \lambda N(f) \right]$$

where

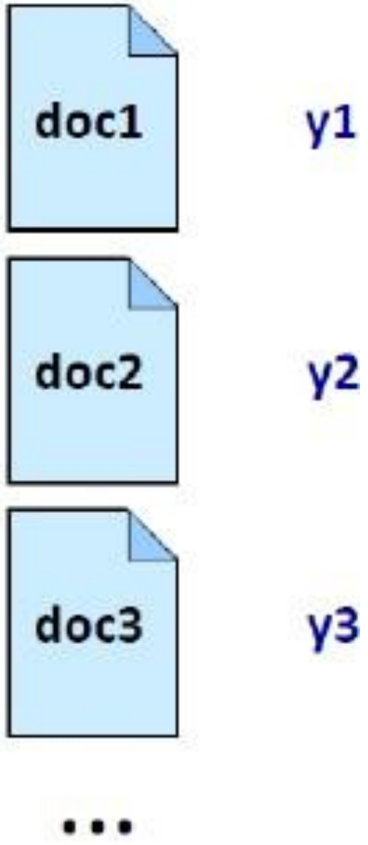
$\ell(f, x_i^b, x_j^a, (b-a))$: convex upper bound on $(b-a)\,\mathbf{1}(f(x_i^b) < f(x_j^a))$

$N(f)$: regularizer

$\lambda > 0$: regularization parameter

$\mathcal{F}$: class of ranking functions

Ranking with Real-Valued Labels



- Instance space X
- Real-valued labels $Y = \mathbb{R}$
- Input: Training sample $S = ((x_1, y_1), \ldots, (x_m, y_m)) \in (X \times \mathbb{R})^m$
- Output: Ranking function $f : X \rightarrow \mathbb{R}$

- Empirical error:

$$\widehat{\mathbf{er}}_S(f) = \frac{1}{\binom{m}{2}} \sum_{1 \leq i < j \leq m} |y_i - y_j| \, \mathbf{1}\left((y_i - y_j)(f(x_i) - f(x_j)) < 0\right)$$

Real-Valued Ranking with SRM

Minimize a convex upper bound on the empirical ranking error, possibly with some regularization, over some class of ranking functions:

$$\min_{f \in \mathcal{F}} \left[\frac{1}{\binom{m}{2}} \sum_{1 \leq i < j \leq m} \ell(f, (x_i, y_i), (x_j, y_j)) + \lambda N(f) \right]$$

where

$\ell(f, (x_i, y_i), (x_j, y_j))$: convex upper bound on $|y_i - y_j| \mathbf{1}\left((y_i - y_j)(f(x_i) - f(x_j)) < 0\right)$

$N(f)$: regularizer

$\lambda > 0$: regularization parameter

$\mathcal{F}$: class of ranking functions

Example

$k = 3$ relevance levels (1 = low, 2 = medium, 3 = high)

Each class has **2 documents**:

Data

S_1 (Rating 1): x_1^1, x_2^1

S_2 (Rating 2): x_1^2, x_2^2

S_3 (Rating 3): x_1^3, x_2^3

Document	Rating	Score $f(x)$
x_1^1	1	1.0
x_2^1	1	2.0
x_1^2	2	2.5
x_2^2	2	2.2
x_1^3	3	2.4
x_2^3	3	1.5

- Ranking SVM maximizes the AUC score
  - Because it ensures that positive examples rank higher than negative examples
    - Ulf Brefeld, Tobias Scheffer. n.d. “AUC Maximizing Support Vector Learning.”

SVMRank Tools

SVM^{rank}

Support Vector Machine for Ranking

Author: Thorsten Joachims <thorsten@joachims.org>
Cornell University
Department of Computer Science

Version: 1.00
Date: 21.03.2009

Overview

SVM^{rank} is an instance of *SVM^{struct}* for efficiently training Ranking SVMs as defined in [Joachims, 2002c]. *SVM^{rank}* solves the same optimization problem as SVM^{light} with the '-z p' option, but it is much faster. On the LETOR 3.0 dataset it takes about a second to train on any of the folds and datasets. The algorithm for solving the quadratic program is a straightforward extension of the ROC-area optimization algorithm described in [Joachims, 2006] for multiple rankings using the one-slack formulation of *SVM^{struct}*. However, since I did not want to spend more than an afternoon on coding *SVM^{rank}*, I only implemented a simple separation oracle that is quadratic in the number of items in each ranking (not the O[k*log k] separation oracle described in [Joachims, 2006]). While this makes the implementation not very suitable for the special case of ordinal regression [Herbrich et al, 1999], it means that it is nevertheless fast for small rankings (i.e. k<1000) and scales linearly in the number of rankings (i.e. queries).

Source Code

The program is free for scientific use. Please contact me, if you are planning to use the software for commercial purposes. The software must not be further distributed without prior permission of the author. If you use SVM^{rank} in your scientific work, please cite as

- T. Joachims, *Training Linear SVMs in Linear Time*, Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD), 2006. [Postscript (gz)] [PDF]

The implementation was developed on Linux with gcc, but compiles also on Solaris, Cygwin, Windows (using MinGW) and Mac (after small modifications, see FAQ). The source code is available at the following location:

http://download.joachims.org/svm_rank/current/svm_rank.tar.gz

https://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html
https://sourceforge.net/p/lemur/wiki/RankLib/ (Java)

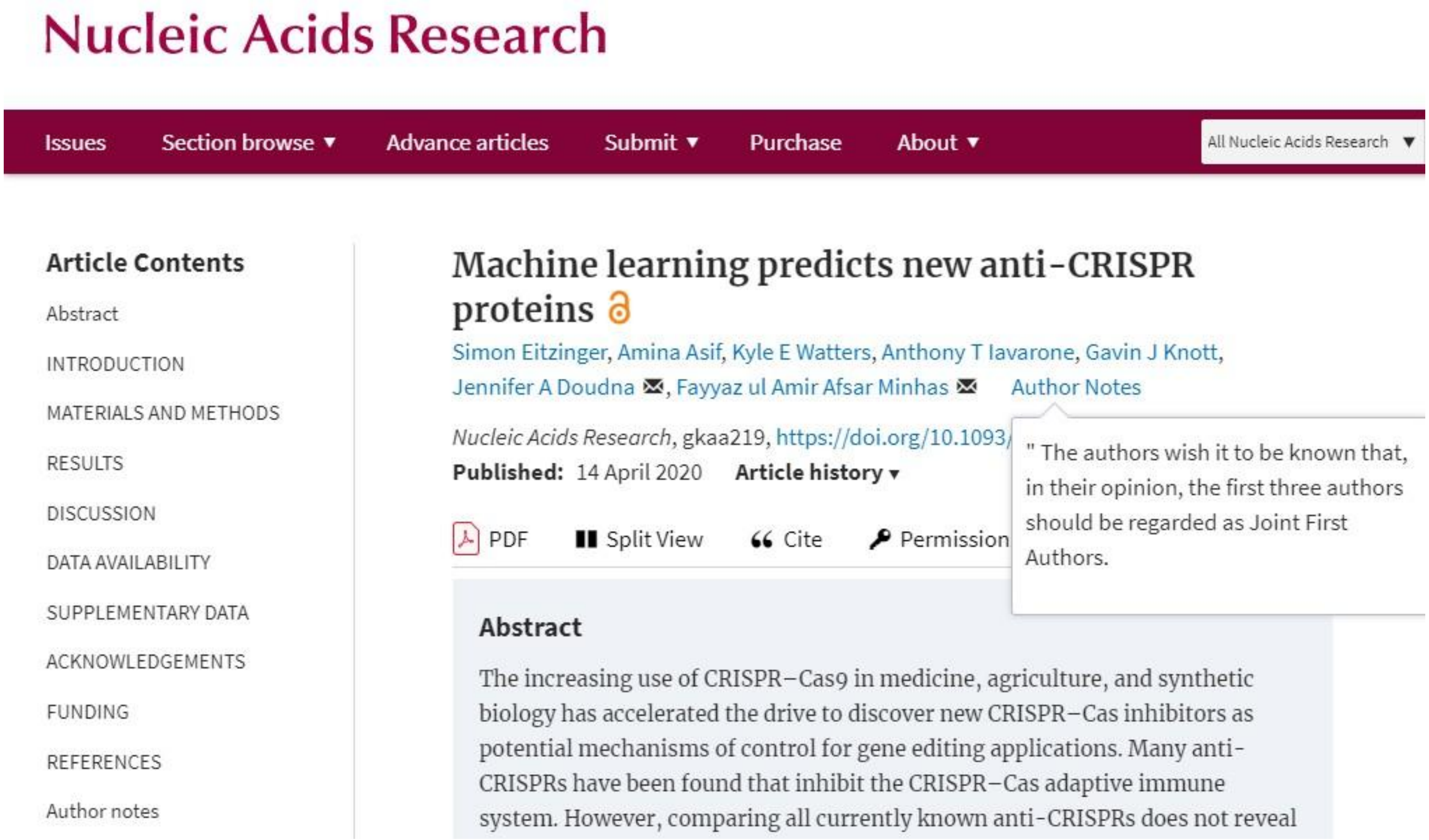
Other methods for ranking

- https://en.wikipedia.org/wiki/Learning_to_rank#List_of_methods

Issues

- The number of constraints becomes quadratic
- Efficient Algorithms Required

Ranking ML case study


Nucleic Acids Research

Issues | Section browse ▾ | Advance articles | Submit ▾ | Purchase | About ▾ | All Nucleic Acids Research ▾

Article Contents

Abstract
INTRODUCTION
MATERIALS AND METHODS
RESULTS
DISCUSSION
DATA AVAILABILITY
SUPPLEMENTARY DATA
ACKNOWLEDGEMENTS
FUNDING
REFERENCES
Author notes

Machine learning predicts new anti-CRISPR proteins

Simon Eitzinger, Amina Asif, Kyle E Watters, Anthony T Iavarone, Gavin J Knott, Jennifer A Doudna, Fayyaz ul Amir Afsar Minhas Author Notes

" The authors wish it to be known that, in their opinion, the first three authors should be regarded as Joint First Authors.

Nucleic Acids Research, gkaa219, https://doi.org/10.1093/

Published: 14 April 2020 **Article history** ▾

PDF Split View Cite Permission

Abstract

The increasing use of CRISPR–Cas9 in medicine, agriculture, and synthetic biology has accelerated the drive to discover new CRISPR–Cas inhibitors as potential mechanisms of control for gene editing applications. Many anti-CRISPRs have been found that inhibit the CRISPR–Cas adaptive immune system. However, comparing all currently known anti-CRISPRs does not reveal

References