Wiki là một chương trình máy chủ cho phép người dùng hợp tác trong việc hình thành nội dung của một trang web. "wiki" trong tiếng Hawaii có nghĩa là "nhanh". Hiện nay, wiki phát triển rất nhanh với nền tảng là website. Chúng ta có thể có wikileaks, wikipedia, wikigames...

**Dataset:** Dữ liệu từ Wikipedia => với mục đích đánh giá khả năng dự đoán của mô hình đánh giá giữa người với người trên các bài viết không chính thức trên web nhưng liên quan đến các kết quả xã hội quan trọng.

**Mô tả dataset:**

- Một thành viên của cộng đồng Wikipedia muốn trở thành quản trị viên thì 1 yêu cầu cho adminship (RFA) là phải nộp đơn hoặc tham gia ứng cử để những thành viên khác của cộng đồng wikipedia bầu chọn theo 3 quan điểm: ủng hộ (supporting), đối lập (neutral), trung lập (oppose).

- Bộ dữ liệu được thu thập và phân tích trong các quá trình RFA từ năm 2003 - 05/2013 trên tất cả các phiếu bầu.

- Bộ dữ liệu gồm 11.402 người dùng (tính cả cử tri và vootes) được tạo thành 189.004 cặp cử tri/voote riêng biệt với 198.275 phiếu (do 1 ứng cử viên có thể chạy chương trình tranh cử nhiều lần và 1 cặp ứng cử viên/vootes có quyền đóng góp ý kiến nhiều lần).

- Tạo thành một network với các node (đại diện cho các thành viên Wikipedia)

và cạnh (đại diện cho phiếu bầu) với điều kiện loại bỏ các phiếu trung lập.

- Nodes: 10.835; cạnh: 159.388 (loại bỏ phiếu trung lập trong tính toán số liệu thống kê nhưng giữ nguyên trong dữ liệu); tạo thành 956.428 hình tam giác.

**Cấu trúc dataset:**

* SRC: tên cử tri
* TGT: tên người tham gia tranh cử
* VOT: nguồn vote chủ yếu(-1 = phản đối; 0 = trung tính; 1 = hỗ trợ)
* RES: kết quả cuộc bầu cử (-1 = bị từ chối; 1 = được chấp nhận)
* YEA: năm cuộc bầu cử diễn ra
* TXT: nhận xét bằng văn bản của các nguồn, trong wiki

**Thiết lập thử nghiệm:**

* Nhóm tác giả thiết lập thử nghiệm theo mô hình “BFS sampling”: chọn ngẫu nhiên 10 node, từ mỗi node sau đó áp dụng thuật toán BFS tìm kiếm theo chiều rộng cho đến khi liên kết được tới 350 node khác => tạo thành 10 subgraph với mỗi subgraph là 350 node. Sau đó, đào tạo một mô hình cho mỗi subgraph i thử nghiệm nó trên subgraph i+1 (mod 10), đảm bảo các cạnh từ biểu đồ đào tạo sẽ được gỡ bỏ khỏi biểu đồ kiểm tra.
* Đánh giá trên 3 mô hình:

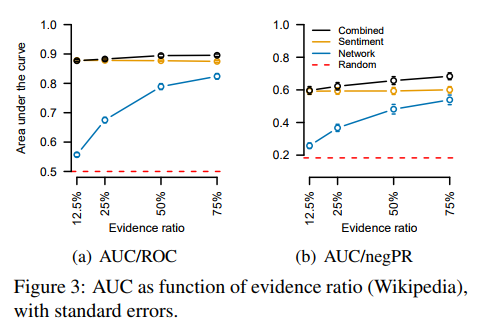
1. ***Sentiment:*** Mô hình chuẩn, dựa trên những đoạn text thể hiện quan điểm, đánh giá và xử lý các cạnh như các điểm dữ liệu độc lập.

Mô hình:

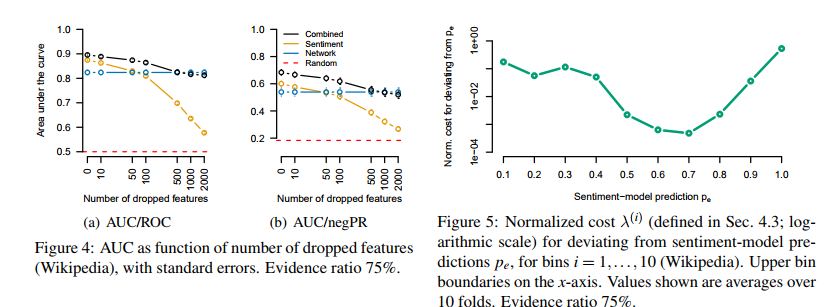
* L2-regularized: phân loại logistic hồi quy có các tính năng có tần số từ 10.000 từ thường gặp nhất.
* L2-penalty: Xác nhận chéo trên tập huấn luyện. Lấy từ các ý kiến rõ ràng chứa nhãn “support” hoặc “oppose”, loại bỏ tất cả các từ chứa “pre-fixes”, “support” hoặc “oppos”. Đào tạo 1 lần trên 1 mẫu ngẫu nhiên gồm 1000 ý kiến/160.000 ý kiến (đại đa số không xuất hiện trong 10 subgraph được tạo ra).

1. ***Combine:*** Mô hình đầy đủ kết hợp chi phí cạnh dựa trên dự đoán của mô hình thứ nhất với tam giác chi phí nắm giữ bối cảnh mạng.
2. ***Network:*** Mô hình chỉ xem xét chi phí tạo trian-gle, trong khi bỏ qua dự đoán của mô hình thứ nhất.

* **Evidence ratio:** khám phá tác động của tỷ số evidence trong suốt thiết lập thử nghiệm. Hiệu suất càng cao thì tỉ lệ chính xác càng lớn.
* **Metricts** (hình dưới): sử dụng các vùng dưới đường cong (AUC) của người nhận;
  + Đường cong đặc trưng hoạt động (ROC): thước đo chuẩn phân loại hiệu suất trên dữ liệu không cân bằng; nắm bắt xác suất 1 bài kiểm tra ngẫu nhiên. Ví dụ: random cho ra kết quả tỉ lệ ủng hộ cao hơn phản đối thì AUC/ROC là 0.5.
  + Đường cong thu hồi chính xác (PR): có 2 loại “positive” và “negative”. Kết quả thu được: 76% positive (UAC = 0.76), 24% negative (UAC = 0.24). Cạnh tiêu cực tuy chỉ chiếm tỷ lệ nhỏ nhưng là quan trọng hơn vì chỉ ra những căng thẳng trong mạng => quan tâm, phát hiện, giải quyết. Biểu đồ bên dưới thể hiện đường cong âm (AUC/negPR)



**Kết quả:**



**Thư viện hỗ trợ: SNAP**

SNAP (Stanford Network Analysis Platform) là thư viện hỗ trợ hiệu suất cao trong phân tích và thao tác với các mạng lớn lên tới hàng triệu nút và hàng tỉ cạnh.

SNAP sử dụng một STL (Standard Template Library) giống như thư viện [GLib](http://kt.ijs.si/dunja/TextGarden/). SNAP và GLib đang được tích cực phát triển và được sử dụng trong rất nhiều dự án học tập và công nghiệp.

SNAP được viết bằng C++ và tối ưu hóa cho hiệu suất tối đa và biểu đồ nhỏ gọn, hỗ trợ tính toán tính chất cấu trúc, tạo ra các đồ thị thường xuyên và ngẫu nhiên, hỗ trợ thuộc tính trên các nút và cạnh và khả năng mở rộng đồ thị.

Hỗ trợ 2 ngôn ngữ là: C++ và Python.

* Snap.py là giao diện Python cho SNAP, cung cấp lợi ích hiệu suất của SNAP, kết hợp với sự linh hoạt của Python, có hầu hết chức năng của SNAP.