# Background for Our Novel Framework for Vertex Nomination

Alex Clemmer
Machine Learning (CS 5350)

October 18, 2011

**Abstract**

Possibly the most common graph problem found in the real world is the case where there is strong verified knowledge of a useful set of attributes for only some fraction of the total vertices. Among such cases, it is frequently true that we care particularly about one feature (or set of features). The process of using this verified knowledge to "nominate" vertices that are likely to have this attribute that we care about is called vertex nomination [Coppersmith and Priebe, 2011].

The end goal for the final project is to produce a useful framework the task of vertex nomination. In this document I hope to present a detailed background on the approach I suggest, as well as a rough program for the suggested solution.

## 1 The State of the Art

The traditional approach to vertex nomination so far is to return a list of the top $k$ vertices that are most likely to have the specific attribute that we're looking for. The problem, often enough, is that the thing we're looking for is often not immediately verifiable. If, for example, the attribute we're looking for is that this person is an evil executive, or a spammer, it is particularly difficult to evaluate how successful the process has been.

The usual approach is to simply look at the mean average precision — the more people that we return who are known to have such an attribute, the "better" the result is, up to a point. This presents a problem of empiricism: if we return a list with *no* vertices that are known to have this attribute, we are not necessarily wrong, since we don't actually know who does and does not have this attribute. And there are plenty of cases where MAP simply paints the wrong picture (the most obvious case being if *every* vertex has this attribute).

One significant consequence of this observation is that pretty much any vertex nomination scheme cast from this mold is not going to be very generalizable: researchers will cast their solution to fit their specific context, and every solution will be $ad-hoc$ fit to that data. Results will become less comparable as the context changes.

We would like to propose a framework that steps away from these problems. First, vertex nominationis fundamentally a clustering task, and we would thus like to decouple the (significant) problems of clustering correctly with the problems of using those clusters effectively. One major advantage of this is that researchers can then take advantage of the troves of research dedicated to these two very well-defined problems.

Second, we would like to develop the idea of vertex nomination into a rich platform for describing a swath of problems that are expressable with this graphical abstraction. We want to step away from the inherent limitations of viewing vertices as people and nomination as role application to those vertices.

## 2 Dirichlet Processes (aka DPs)

A probablility measure (with some simplification) can be thought of as any probability function $G$ for which $G(\Omega) = 1$, $G(0) = 0$, and the disjoint union $G(A_0 \dot{\cup} \dots \dot{\cup} A_m)$ where $A_i \subset \Omega$.

If we think of $(\Theta, B)$ as a measureable space, with $G$ as a random probability measure on that space. For this random variable $G$, a **Dirichlet Process** is basically the distribution of probability measures that $G$ can possibly be over the space $(\Theta, B)$, such that any possible finite partition $(A_0, \dots, A_m)$ of $\Theta$ has a corresponding random vector $(G(A_0), \dots, G(A_m))$ which is distributed as a finite-dimensional Dirichlet. It is said thus of this vector, for some positive real $\alpha$ (which we can basically ignore for now):

$$(G(A_0), \dots, G(A_m)) \sim \text{Dirichlet}(\alpha_0 G_0(A_0, \dots \alpha_0 G_0(A_m)) \tag{1}$$

Note here that $G_0$ is the *base measure* of $G$; $\alpha_0$ is known as the concentration parameter (and, again, ignored for now).

## 3 DP Mixture Models

Let $G \sim DP(\alpha_0, G_0)$ be a random probability measure distributed according to the formulation of Dirichlet Processes noted above. If we have an $n$-element vector of (exchangeable) data $\vec{x} = (x_0, \ldots, x_n)$, then we can draw $n$ mixture weights $\phi_i \sim G, \forall i \in [0..n]$. If this is confusing, remember that the mixture weight $\phi_i$ is really the prior probability of the component $x_i$.

From here, according to the formulation of the standard mixture model, $x_i \sim F(\phi_i)$ for some distribution $F$. When this is drawn for every $i \in [0..n]$, we have formed what is known as the **Dirichlet Process Mixture Model**.

Note that if each of these weights $\phi_i$ were all completely distinct, the model would effectively be a useless $n$-component vector; for reasons involving exchangeability of random variables, this is simply not the case; the distinct values actually only grow in $O(\log n)$, but I won't derive the solution here. You can ask me if you really care.

## 4 Hierarchical DP Mixture Models

HDP more or less allows for a