Dns2Vec: Exploring Internet Domain Names with Deep Learning

Amit Arora
Georgetown University

Abstract

The concept of vector space embeddings was first applied in the area of NLP but has since been applied to several domains wherever there is an aspect of semantic similarity. Here we apply vector space embeddings to Internet Domain Names. We call this ***Dns2Vec***.

A corpus of Domain Name Server (DNS) queries was created from traffic from a large Internet Service Provider (ISP). A skipgram word2vec model was used to create embeddings for domain names. The objective was to find similar domains and examine if domains in the same category (news, shopping etc.) cluster together. The embeddings could then be used for several traffic engineering application such as shaping, content filtering, prioritization and also for predicting browsing sequence and anomaly detection.

The results were confirmed by manually examining similar domains returned by the model, visualizing clusters using t-SNE and also using a 3rd party web categorization service (Symantec K9).

Introduction

An ISP is always interested in understanding how to better prioritize, shape or even filter traffic if needed. However, with the ubiquitous use of HTTPS and other encrypted protocols this has become much harder. Vector embeddings for domain names would encode similarity between web pages and once available for a large corpus of domain names it can very quickly be used map web traffic to a category (shopping, entertainment, news/media etc.) and a whole host of other traffic engineering applications.

There are several COTS solutions for finding similar domains such as SimilarWeb, Web Shrinker and Alexa Sites. A 2017 paper by *Waldemar L´opez et.al* [1], discusses the idea of word embeddings for domain names. *Dns2Vec* while having the same core idea uses different approaches for pruning the dataset, a different set of hyperparameters that work better for our dataset and evaluating the effectiveness of embeddings by using web site categorization rather than direct similarity as that is better suited for traffic engineering applications.

Dns2Vec is Word2Vec applied to DNS traffic by considering DNS queries from a particular source IP address in a time interval as "words" in a single "document". The Word2Vec model uses a single hidden layer neural net to learn word embeddings. The skipgram variant of the Word2Vec model is used here as we want to predict the target word (similar domain name) given another word (input domain name).

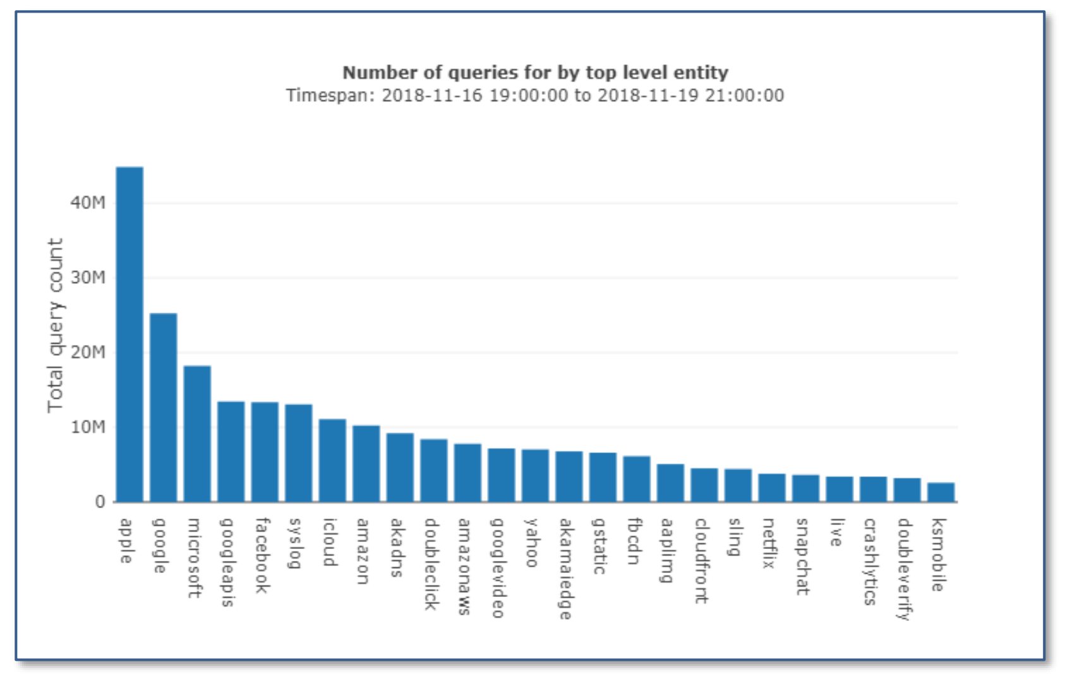

Figure 1. DNS Query count by top level entity

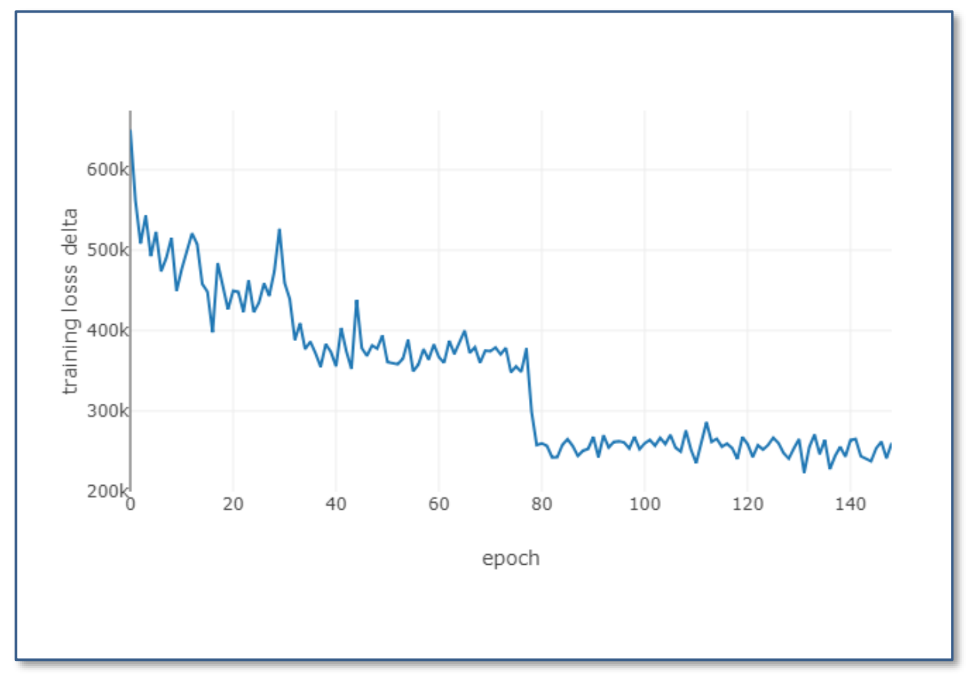

Figure 2. Delta training loss per iteration while training Word2Vec on DNS data

Methods and Materials

DNS traffic was collected from a large ISP for a 3 day period. The dataset consisted of CSV files (one for each day) in the format *<timestamp>,<IP version>, <Source IP address>,<Query Name>., <Query Type>*. The files were combined into a single file which had a total of 589 million rows (one DNS query per row) and provided to a data cleaning and wrangling module.

To remove the heterogeneity from domain names belonging to the same entity the domains were shortened to 2 or 3 significant levels (for domains ending in country specific suffixes) resulting in 1.05 million unique domains.

The file was parsed, each line tokenized and a new timestamp aggregation field was created to group timestamps into 20 minute intervals. <$T_{interval}$, Source IP address> tuple was used as a grouping key to create a *Documents* of DNS queries <$T_{interval}$, Source IP address, [dn1, dn2, ..,dnN]> and finally the entire dataset was converted into a *Corpus* of DNS *Documents*.

Stop-domains akin to stop-words were removed from each Document. Tf-Idf scores were determined for each domain name and the 100 domain names with the lowest Tf-Idf scores were treated as stop-domains, these included google.com, apple.com etc.

The corpus thus created was fed to the Gensim package's Word2Vec implementation (**Figure 3**). Hyperparameters of the model: vocabulary size 40,000, sample 10^{-5}, negative exponent -1, embedding size 128, window 7, negative samples 5, training iterations 150. **Figure 2**.

Results

All 1.05 million domains were grouped into categories such as hopping, news/media, sports, adult etc. using a website categorization service available from the Symantec K9.

The results were evaluated in the following three ways:

1. **Manual inspection of similar domains** retuned by the Dns2Vec model given an input domain name. See **Table 1** for similar domain name, **Table 3** and **Figure 4** for vector arithmetic on domain name vectors to find **analogies in domain name space.**
2. **Finding top 3** most similar domains corresponding to a given name and then considering the result to be valid if at least one of the top 3 domains is in the same category as the input domain. The **success metric** was then evaluated over the entire Dns2Vec vocabulary. This number was found to be **55.4%**.
3. **Plotting the domain name vectors using t-SNE (Figure 4)** and color coding the markers by category. Domains belonging to the same category tend to cluster together, this is especially true for search uncategorized/suspicious, adult content, news/media.

Table 1. Top 3 Similar Domains found by *Dns2Vec*

Domain Name	Similar Domains
theatlantic.com	nytimes.com, getpocket.com, npr.org
food.com	geniuskitchen.com, whisk.com, snidigital.com
glassdoor.com	ziprecruiter.com, indeed.com, glassdoor.de

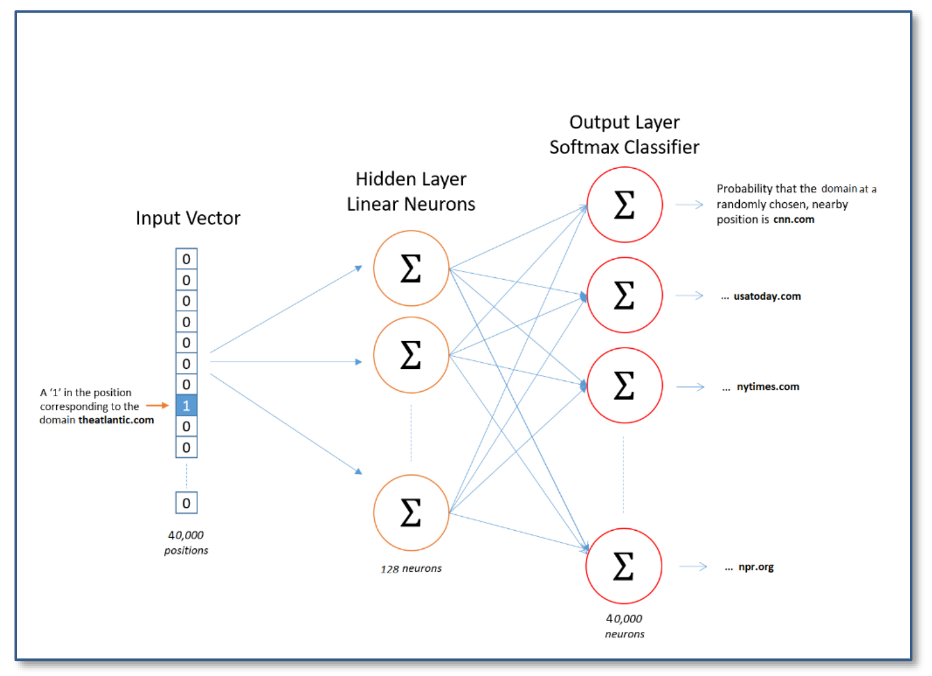

Figure 3. Dns2Vec Neural Network architecture

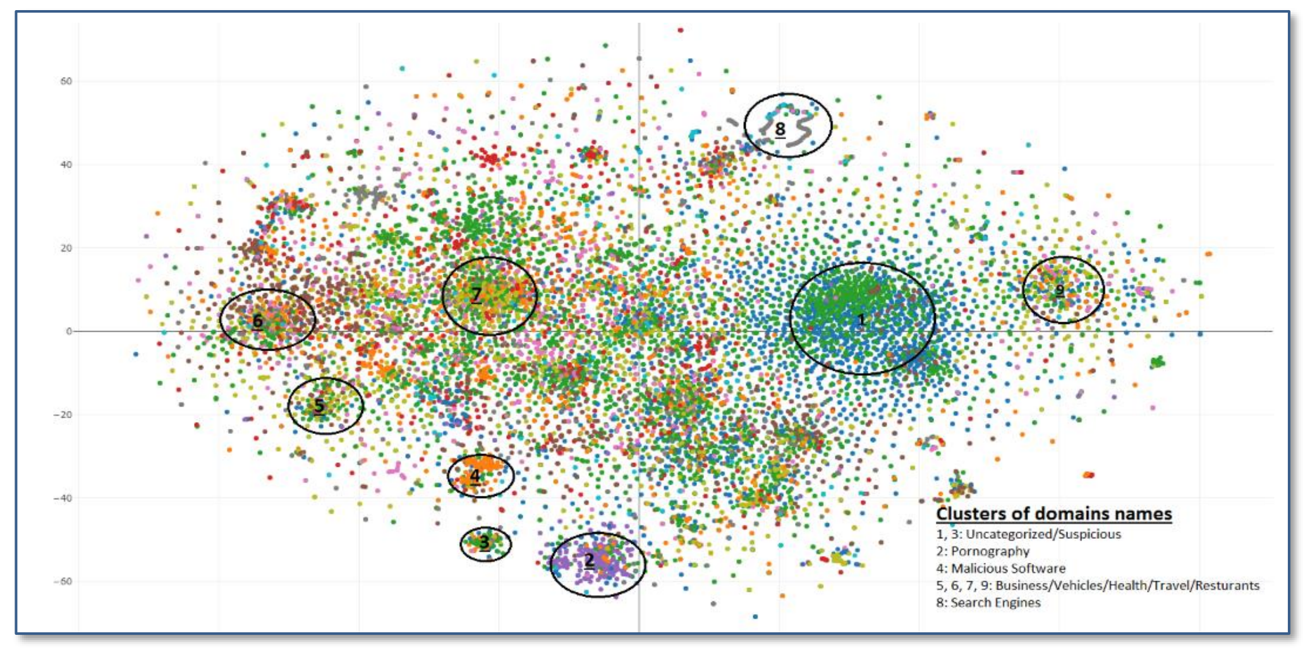

Figure 4. t-SNE on Vector Embeddings for domain names.

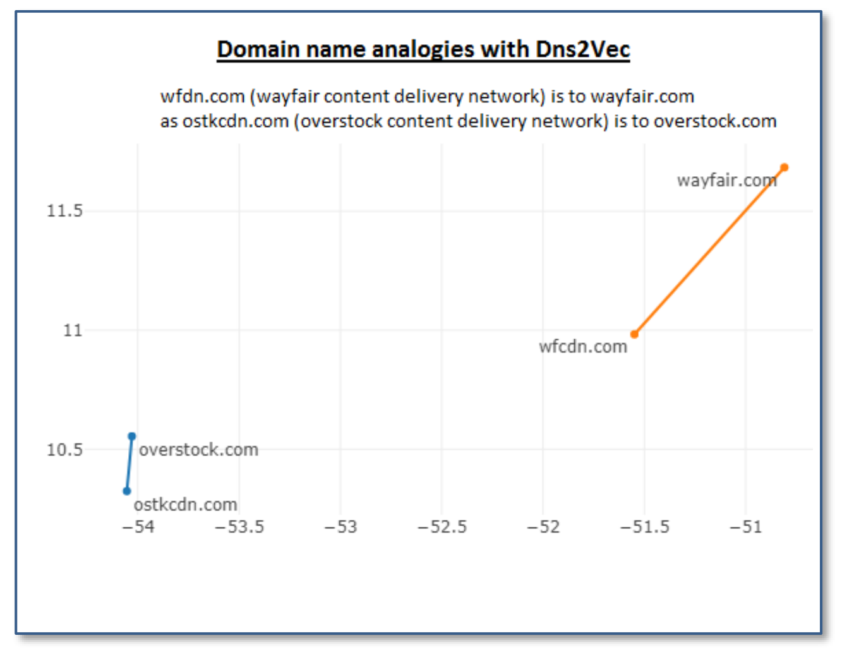

Figure 5. Vector arithmetic to find analogies

Table 2. Analogy illustration using *Dns2Vec*

Input	Analogy discovered by *Dns2Vec*
wfcdn.com is to wayfair.com	ovstkcdn.com is to overstock.com (**Relation**: find cdn (content delivery network) to a website)
tripadvisor.com is to delta.com	nps.gov is to lyft.com (**Relation**: reviews about a place and means to get there)

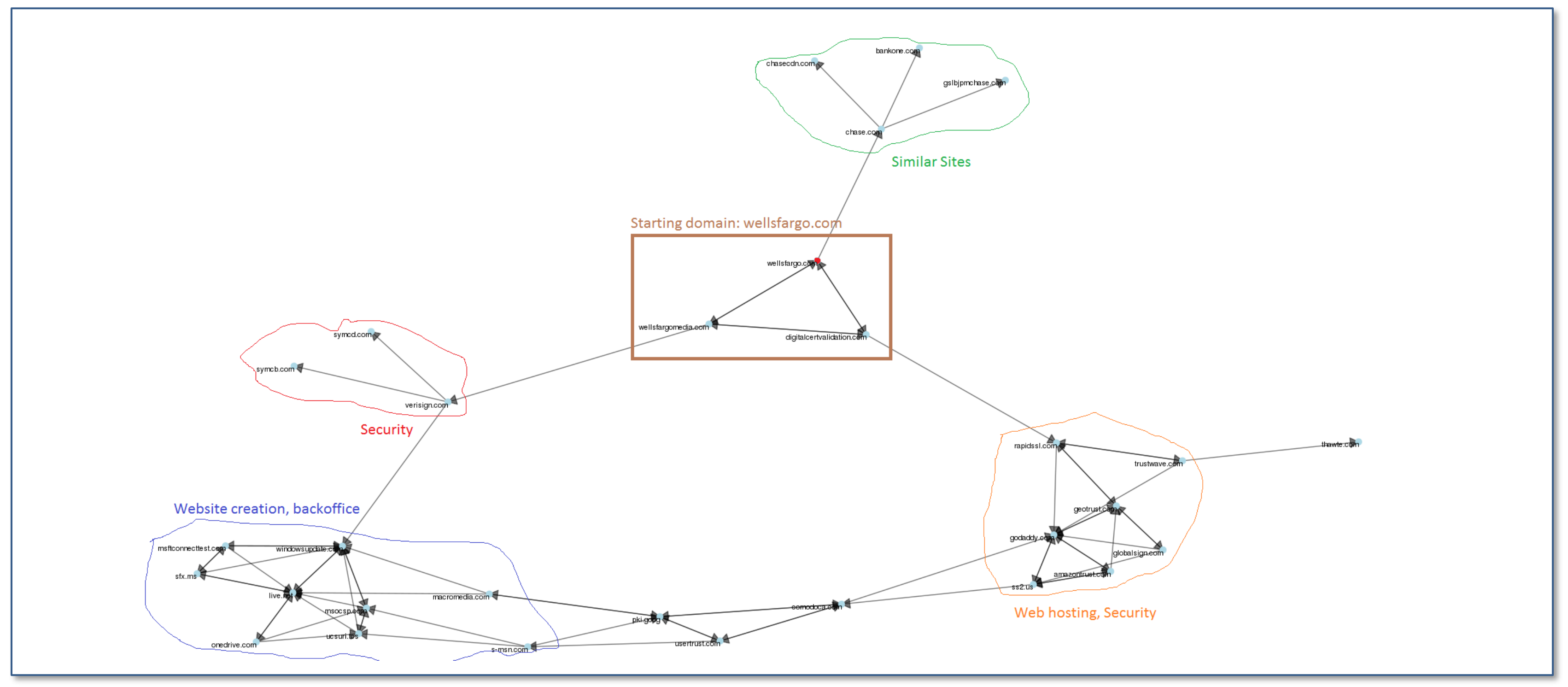

Figure 6. Exploring DNS vector space using a seed domain

Discussion and Conclusions

We show that vector embeddings can be used for representing domain names and the DNS vectors do encode similarity between domains (**Table 1, Figure 4**). We also show that vector arithmetic is able to find complex relations between domains that corresponds to real world linkages (**Table 2, Figure 5**). Furthermore, the domain name vector space can be traversed to build network topologies (**Figure 6**). All this has tremendous values to various traffic engineering functions.

Future work in this area would be to explore further hyperparameter optimization to get better embedding efficiency and separation in t-SNE representation. Also, collecting more data would certainly produce even better quality vectors. An alternative to using DNS logs could be to use IP flow logs corresponding to actual traffic flows, this would present yelp.com and yelp.de from appearing as similar domains which would help with traffic prediction applications.

Contact

Amit Arora
Georgetown University
3700 O St NW, Washington, DC 20057
aa1603@georgetown.edu
301-525-6705

References

1. Waldemar L´opez, Jorge Merlino and Pablo Rodr´ıguez-Bocca. 2017. Vector representation of Internet Domain Names using a Word Embedding technique.
2. Hugo Caselles-Dupré, Florian Lesaint, Jimena Royo-Letelier. 2018. Word2vec applied to Recommendation: Hyperparameters Matter. arXiv:1804.04212v3 [cs.IR].
3. Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. CoRR, vol. abs/1405.4053.
4. Ben Eisner, Tim Rockt¨aschel, Isabelle Augenstein, Matko Bosnjak, and Sebastian Riedel. 2016. emoji2vec: Learning emoji representations from their description, CoRR, ol. abs/1609.08359.
5. Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space," CoRR, vol. abs/1301.3781.
6. Ledell Wu, Adam Fisch, Sumit Chopra, Keith Adams, Antoine Bordes, Jason Weston. 2017. StarSpace: Embed All The Things!. arXiv:1709.03856 [cs.CL]
7. Andrej Karpathy. 2015. The Unreasonable Effectiveness of Recurrent Neural Networks, http://karpathy.github.io/2015/05/21/rnn-effectiveness/.

© Poster Template by Genigraphics® 1.800.790.4001 www.genigraphics.com