Robust Word Similarity Estimation using Perturbation Kernels

Kevyn Collins-Thompson
Microsoft Research
1 Microsoft Way
Redmond, WA 98052
kevynct@microsoft.com

Abstract. We introduce *perturbation kernels*, a new class of similarity measure for information retrieval that casts word similarity in terms of multi-task learning. Perturbation kernels model uncertainty in the user's query by choosing a small number of variations in the relative weights of the query terms to build a more complete picture of the query context, which is then used to compute a form of *expected distance* between words. Our approach has a principled mathematical foundation, a simple analytical form, and makes few assumptions about the underlying retrieval model, making it easy to apply in a broad family of existing query expansion and model estimation algorithms.

1 Introduction

A fundamental research problem of information retrieval is how to improve search effectiveness by learning an extended representation of the user's information need, called a *query model*, that captures more about the context of an information need than is available from the few words in the query itself. For example, in performing a type of query expansion, a very simple query model might take the form of a unigram language model over words related to the user's query terms. One significant problem in performing query expansion is the *risk* of adding words that are unrelated to the query, causing the query model to 'drift' away from the user's original intent. Thus, improving the quality and reliability of the similarity measure used to find related terms is an important goal in itself.

With this problem in mind, we introduce *perturbation kernels*, which cast estimating word similarity as an type of multi-task learning problem. Informally, the key idea of perturbation-based kernels is that two input objects x and y, such as words, are considered similar in the context of a given query Q if probability distributions $p(x|Q)$ and $p(y|Q)$ that depend on Q are affected in similar ways with small variations in Q.

Our approach has several advantages. First, the use of query perturbations results in sensitivity features that give more precise word similarity relations, which in turn can improve the stability of query expansion algorithms that use them. Second, we make few assumptions about the nature of the underlying retrieval model, meaning that such similarity measures may be applied in a wide variety of existing query model estimation or expansion methods. Third, our solution has sound theoretical justification with close connections to kernel methods such as the leave-one-out kernel [13], robust approximation, and metric learning. Finally, our algorithm has a simple, efficient analytical form.

2 Mathematical formulation

Let $\mathcal{X}$ be the input domain of interest (e.g. words) from which a training example (query) is generated. For any $x \in \mathcal{X}$ we identify an m-dimensional vector called a *feature mapping*, denoted $\phi : \mathcal{X} \to \mathbb{R}^m$. With this feature mapping, we define a symmetric kernel function $k(\cdot)$ to measure the closeness of input points x and y as $k(x, y) = \phi(x) \cdot \phi(y)$ where the right side of the equation is the *inner product* of $\phi(x)$ and $\phi(y)$.

A *perturbation* to a training set of n instances $x = \{x_1 \ldots x_n\}$ can modeled by a vector of counts $\alpha = \{\alpha_1, \ldots, \alpha_n\}$ with count α_i corresponding to the weight of training example x_i. For the original training set, $\alpha_i = 1$ for all instances x_i. To leave out the instance x_i, we set $\alpha_i = 0$. To give x_i more weight, we set $\alpha_i > 1$. A *perturbation strategy* is a set $\mathcal{A} = \{\alpha_i\}$ of perturbation vectors. The set $\mathcal{A}$ may be selected with either a random or deterministic process. We use the following deterministic perturbation strategies in this study to define uncertainty sets around the initial query Q:

– The *leave-one-out* strategy (LOO) has $\mathcal{A} = \{\alpha_1, \ldots, \alpha_n\}$ where $\alpha_i[j] = 0$ for $i = j$ and 1 otherwise.
– The *term-at-a-time* strategy (TAT) is complementary to LOO and uses $\mathcal{A} = \{\alpha_1, \ldots, \alpha_n\}$ where $\alpha_i[j] = 1$ for $i = j$ and 0 otherwise.

These strategies are extremely simple to implement, widely used for query expansion and performance prediction tasks, and fast to execute in a real-time search environment. They both use $N + 1$ variants of the query, while being somewhat complementary strategies, making them ideal for comparison. The TAT and LOO methods are different extremes in a more general class of combinatorial methods that could be defined on the set of query terms. We leave exploration of more sophisticated perturbation schemes for future work.

We denote the probability distribution of $x \in \mathcal{X}$ that results from a perturbation α_i as $p^{(i)}(x)$. In the context of information retrieval, we view a query q as a training set of n instances of query terms $\{q_1, \ldots, q_n\}$ where q_i is drawn from $\mathcal{X} = \mathcal{V}$ for vocabulary $\mathcal{V}$. In the next sections, we derive the general form of the perturbation kernel and feature mapping $\phi(x)$.

2.1 Canonical similarity integrals

Our formulation of perturbation-based similarity is inspired mainly by earlier work of Baxter using auxiliary tasks for classification and function approximation, as well as a more recent followup article by Minka [11] that discussed distance measures as prior probabilities. Baxter showed that for 1-nearest-neighbor classification, there is a unique optimal similarity measure that he called the Canonical Distortion Measure (CDM) [3]. In the classification setting, this quantity $\delta(x_1, x_2)$ is the expected loss of classifying x_1 with x_2's label. The expectation is taken over a probability space of classifiers (tasks). To apply Baxter's idea to information retrieval applications, we view a task as relevance estimation with respect to a particular query q. We call a task distribution for q a *query neighborhood* of q. One way to define the query neighborhood is as a probability measure $Q(f)$ over task functions $f_q(x)$ where $f_q(x)$ gives a 'soft' label $p_q(x|\theta_R)$ with respect to the unknown 'true' Relevance Model θ_R for query q. The canonical similarity measure $\Delta_q(x, y)$ is then the expected loss over Q, given x and y in the input domain $\mathcal{X}$.

$$\Delta_q(x, y) = \int_{\mathcal{F}} \rho(f(x), f(y)) dQ(f) \tag{1}$$

This measure is uniquely determined by the task function f, the choice of query neighborhood measure Q and loss function $\rho(u, v)$. We fix the two input domain elements x and y (words) and integrate over a probability space $\mathcal{P}$ of density functions. These density functions $p^{(\alpha)}(\cdot)$ are those that result from perturbations α on the training data (query), and we assume we have a measure $Q(p)$ over $\mathcal{P}$ that describes the distribution over perturbation densities $p^{(\alpha)}(\cdot)$. We assume that the density $p^{(\alpha)}(\cdot)$ is defined for all elements of the input domain $\mathcal{X}$ (although it may be zero), so that the integral exists for all pairs $(u, v) \in \mathcal{X}^2$ with measure $G(u, v|x, y)$. The general form of the perturbation kernel is then defined to be an expected distance between x and y:

$$k_q(x, y) = \int_{\mathcal{X}^2} \int_{\mathcal{P}} \rho(p^{(\alpha)}(u), p^{(\alpha)}(v)) dG(u, v|x, y) dQ(p). \tag{2}$$

One natural choice for $\rho(x, y)$ is $\rho(x, y) = (\sqrt{x} - \sqrt{\kappa})(\sqrt{y} - \sqrt{\kappa})$ for a fixed origin κ since this converts the integral in Eq. 2 to a form of Hellinger inner product between vectors $\{p^{(\alpha)}(x)\}$ and $\{p^{(\alpha)}(y)\}$ when we set $\kappa = p(x)$. This choice is motivated by connections with Fisher kernels [13], but other choices for $\rho(x, y)$ are open to exploration (as well as other perturbation strategies).

2.2 Approximating the similarity integral

By writing the similarity measure in Eq. 2 as an integral we can bring to bear general-purpose integration methods. Recall that the basic approach to evaluate a general integral of the form $\mathcal{I} = \int_{\Theta} f(\theta) d\mu(\theta)$ on the domain Θ with measure $d\mu$ is to independently sample N points $X_1, \ldots X_N$ in Θ according to some density function $p(x)$, and then compute the random variable $F_N = \frac{1}{n} \sum_{i=1}^{n} \frac{f(X_i)}{p(X_i)}$. A simple importance sampling approximation to Eq. 2 can be derived by assigning the perturbation densities $\{p^{(\alpha)}(x)\}$ equal probability, and taking a single-sample approximation at (x, y) in the $\mathcal{X}^2$ domain, where we assume there is some sampling distribution MLE $\hat{p}(x, y)$ and that words x and y are independent (given relevance) so that $\hat{p}(x,y) = \hat{p}(x) \cdot \hat{p}(y)$ for MLE $\hat{p}(x)$ over $\mathcal{X}$, giving

$$k_q(x,y) \approx \frac{1}{\hat{p}(x,y)} \cdot \frac{1}{n} \sum_{i=1}^{n} \rho(p^{(i)}(x), p^{(i)}(y)) = \sum_{i=1}^{n} \phi(x_i)\phi(y_i) \tag{3}$$

so that the feature mapping vector $\phi(x)$ has entries

$$\phi_i(x) = \frac{1}{\sqrt{n}} \frac{\sqrt{\hat{p}^{(i)}}(x) - \sqrt{\hat{p}(x)}}{\hat{p}(x)}. \tag{4}$$

Typically, $\hat{p}(x)$ represents the distribution obtained using the results from the initial query, which can be considered a null perturbation.

2.3 Algorithm

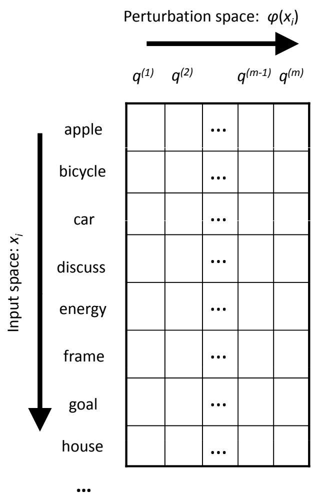
The specific steps of the perturbation kernel algorithm are given in Fig. 1. Here we show the case where we use the language modeling approach to IR, and estimate a unigram Relevance Model [10] $p_q(w|R)$ for a given query q and word $w \in \mathcal{V}$. The resulting distance matrix D, which can be viewed as a graph over words, is then typically used as one of the inputs to a specific query model estimation or expansion algorithm.

1. Given initial query q, generate N query perturbations $q^{(i)}$, $i = 1 \ldots N$.
2. Run q to generate a corresponding Relevance Model $\hat{p}(w) = p(w|R)$, and similarly run each $q^{(i)}$ to generate Relevance Model $p^{(i)}(w|R)$.
3. Compute matrix A, which has one row $\phi(w)$ for each word w in vocabulary $\mathcal{V}$, with the i-th entry $\phi_i(w)$ of the row computed using Eq. 4.
4. Compute the final word-word distance matrix D having entry $d_{ij} = g_{ii} + g_{jj} - 2g_{ij}$ where g_{ij} are the entries of the Gram matrix $G = AA^{\mathsf{T}}$.

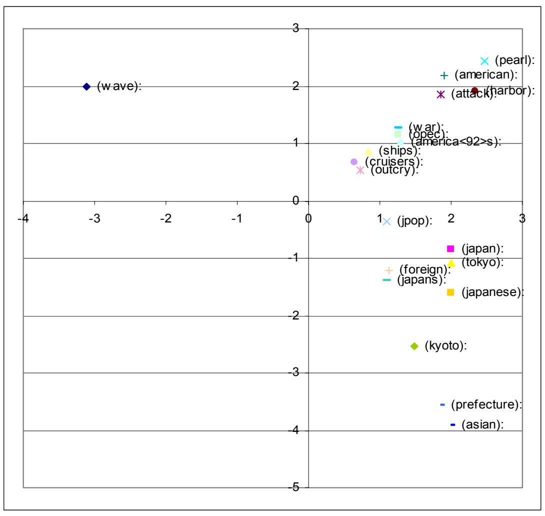
Fig. 1: The perturbation kernel algorithm (Relevance Models).

2.4 Visualizing perturbation similarity

Plotting words in feature mapping (perturbation) coordinates shows useful local and global distance properties. An example is shown in Figure 2. The x-axis plots $\log \phi_1(w)$ and the y-axis plots $\log \phi_2(w)$ where the feature mapping $\phi(w)$ is given in Eq. 4, and the perturbation strategy uses LOO query variants. Words with similar relative changes in relevance model $p(w|R)$ to the same query perturbations are close in this space. The mutual proximity of the query terms gives an indication for how phrase-like their behavior is, while the global position of clusters from the origin is related to their relevance to the query. Here, the 'japanese wave' query seeks information about tsunamis; the irrelevant 'Pearl Harbor' noise cluster has been successfully separated from the other terms and placed in the NE quadrant, while the SE quadrant brings together words much more closely related to 'japanese', such as 'asian' and 'prefecture'.


Perturbation space: $\varphi(x_i)$
$q^{(1)}$ $q^{(2)}$ $q^{(m-1)}$ $q^{(m)}$
Input space: x_i
apple
bicycle
car
discuss
energy
frame
goal
house
...

(a) Perturbation matrix A

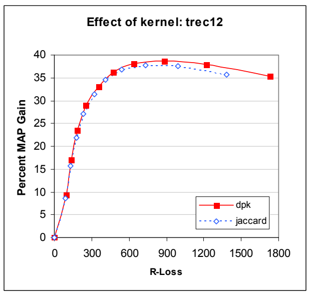


(b) Visualization example

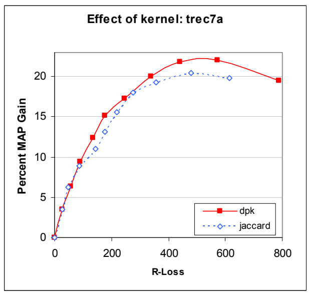
Fig. 2: (a) The matrix of probability vectors for a discrete input space (here representing a word vocabulary). Each column represents the discrete parametric or non-parametric probability distribution across all words estimated from a particular perturbation of the training data (i.e. query). The rows $\phi(x_i)$ give the probability estimates across all perturbations for a given word x_i. (b) Visualization showing how the perturbation kernel is effective at term clustering for the top 20 expansion terms for TREC topic 491 'japanese wave'. The first two co-ordinates in perturbation space are plotted as x and y axes. Terms whose probabilities respond similarly to the same query perturbations are close in this space. Close words have been jittered apart for clarity.

3 Evaluation for query expansion

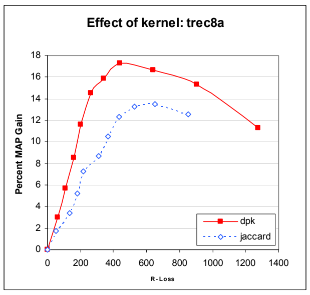
Since our focus here is on making expansion algorithms more stable, we introduce *risk-reward tradeoff curves* to visualize and compare the *risk profile* of query expansion algorithms [5]. The x-axis gives a measure of downside risk or variance of the expansion algorithm, by counting the net loss of relevant documents for queries whose initial results are made worse by using the expansion algorithm (which we call R-Loss). The y-axis gives the actual relative gain or loss from using expansion, as measured by Mean Average Precision (MAP) gain over all queries. The eleven points on each curve show how the risk-reward tradeoff changes as the query model for the initial query is interpolated using parameter α with the expansion model: from $\alpha = 0$ at the origin (initial query only) to $\alpha = 1.0$, where the query model consists entirely of the expansion model. The curves for six standard TREC topic sets are shown in Figure 3, with the number of queries in the topic set given in parentheses.



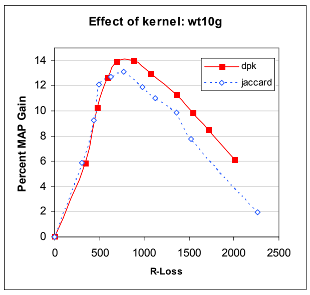
(a) TREC 1& 2 (n=150)



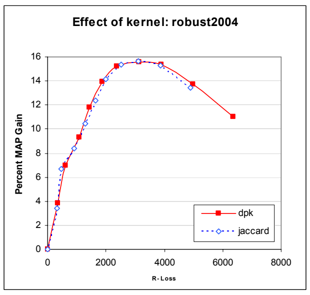
(b) TREC 7 (n=50)



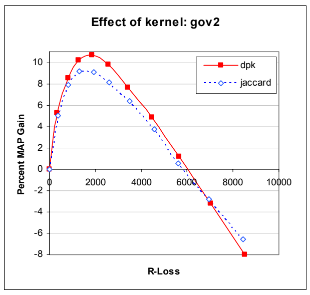
(c) TREC 8 (n=50)



(d) wt10g (n=100)



(e) Robust 2004 (n=250)



(f) gov2 (2004-2006, n=150)

Fig. 3: Risk-reward tradeoff curves for six TREC topic sets, showing how the perturbation kernel can improve the risk profile of an expansion algorithm. The solid line is the curve given by the expansion algorithm using the perturbation kernel. The dashed line uses the same expansion algorithm and parameter settings, but substitutes a Jaccard kernel that does not use sensitivity information. Tradeoff curves that are *higher and to the left* give a better risk-reward tradeoff. Curves are plotted with points at α-increments of 0.1, starting with $\alpha = 0$ at the origin and increasing to $\alpha = 1.0$.

The risk-aware query expansion framework we use is described in detail in [4]. For a baseline word similarity measure, we wanted a method that could be calculated from just the initial set of top-retrieved documents. Recall that we are deriving term association statistics from a set of documents that is already biased toward the query terms, so that the number of documents *not* containing a query term is frequently zero, or close to zero. We chose the Jaccard measure for this study since it is a simple, widely-known term association measure that ignores this non-relevant negative information.

For four of the six collections (TREC 7, TREC 8, wt10g, and gov2) the perturbation kernel improved the risk profile of the query expansion algorithm, particularly in the typical operational zone from $\alpha = 0$ to $\alpha = 0.5$. At a setting of $\alpha = 0.5$, the improvements are largest for TREC 8 and gov2. For TREC 8, the perturbation kernel gives a MAP gain of 14.5% with R-Loss of 262, while the Jaccard kernel gives a MAP gain of 8.68% with R-Loss of 307. For the gov2 collection, the perturbation kernel MAP gain is 9.78% with R-Loss of 2555, while the Jaccard kernel has MAP gain of 8.13% with R-Loss of 2605. For two of the collections (TREC 1&2 and Robust 2004), the performance of the two kernels is almost identical: TREC 1&2 shows only a tiny advantage for the perturbation kernel for $\alpha \geq 0.6$. We found that LOO perturbation had a small but consistently dominance in performance over TAT for all experiments and so we report only LOO experiments here. The results suggest that the perturbation kernel gives the potential for useful gains on some collections, with little downside risk.

4 Related Work

Recent research on kernels has developed a broad family based on inner products over probability distributions. When we assign each input point x_i a probability distribution over input space, we can integrate over input space – in the discrete case, the *columns* of A shown in Figure 2a, instead of the *rows* of A. This type of similarity measure includes probability product kernels [8]; the leave-one-out (LOO) kernel [13]; and marginalized kernels [14]. Fisher kernels [7], a special case of marginalized kernel, compare the sufficient statistics of generative models and are well-suited to query model problems, because they can exploit unlabeled data: the similarity of two data items is not only a function of the items themselves but also their context. There are also interesting connections to information diffusion kernels [9] and statistical translation models such as those developed by Dillon *et al.* [6], where word similarity is defined in terms of the probability that two words have the same context.

In multi-task learning, in addition to Baxter [3], Ando *et al.* used a multi-task learning framework [2] in a preliminary TREC genomics study [1] where they noted the connection between multi-task learning and using auxiliary queries. For Web retrieval, Sahami and Heilman [12] proposed a kernel for comparing text snippets using the inner product of the query expansions that result by considering each text snippet as a Web query. None of these methods, however, explored the use of sensitivity information or framed the similarity problem in terms of multi-task learning to improve stability of the 'client' algorithm.

5 Conclusions

The perturbation kernel is a useful tool for comparing the similarity of elements in a domain $\mathcal{X}$, such as words, when we have a probability distribution over $\mathcal{X}$ whose functional form may be unknown and/or highly complex and which is estimated based on a very small training set over elements from $\mathcal{X}$. Similarity between elements is induced with respect to small perturbations in the training data, so that each input point is identified with multiple probability densities evaluated at that point that are integrated over probability density space. We showed how casting word similarity estimation as a multi-task learning problem of this type can exploit knowledge from multiple 'tasks' in the form of query perturbations. Our initial evaluation suggests that the perturbation kernel is a more stable replacement for similar baseline measures that ignore sensitivity. More generally, it represents a step in a fruitful research direction: exploring how information retrieval algorithms can exploit higher-level risk or variance information to improve their performance [5]. Further improvements may be possible with more sophisticated perturbation strategies for the query, such as those learned from Web query logs or user profiles.

References

1. R. K. Ando, M. Dredze, and T. Zhang. TREC 2005 genomics track experiments at IBM Watson. In *Proceedings of TREC 2005.* NIST Special Publication, 2006.
2. R. K. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *J. Mach. Learning Research*, 6:1817–1853, 2005.
3. J. Baxter. The canonical distortion measure for vector quantization and function approximation. In *ICML 1997*, pages 39–47.
4. K. Collins-Thompson. Estimating robust query models using convex optimization. In *NIPS 2008*, 2008.
5. K. Collins-Thompson. *Robust Model Estimation Methods for Information Retrieval.* PhD thesis, Carnegie Mellon University, 2008.
6. J. Dillon, Y. Mao, G. Lebanon, and J. Zhang. Statistical translation, heat kernels, and expected distances. In *UAI 2007*, pages 93–100.
7. T. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. In *NIPS 11*, 1999.
8. T. Jebara, R. Kondor, and A. Howard. Probability product kernels. *J. Machine Learning Research*, 5:819–844, 2004.
9. J. D. Lafferty and G. Lebanon. Information diffusion kernels. In *Advances in Neural Information Processing Systems (NIPS) 15*, pages 375–382. MIT Press, 2002.
10. V. Lavrenko. *A Generative Theory of Relevance.* PhD thesis, Univ. of Massachusetts, Amherst, 2004.
11. T. Minka. Distance measures as prior probabilities. Technical report, 2000.
12. M. Sahami and T. Heilman. A web-based kernel function for measuring the similarity of short text snippets. In *Proc. of WWW 2006*, pages 377–386.
13. K. Tsuda and M. Kawanabe. The leave-one-out kernel. In *ICANN 2002*, pages 727–732, London, UK, 2002. Springer-Verlag.
14. K. Tsuda, T. Kin, and K. Asai. Marginalized kernels for biological sequences. *Bioinformatics*, 18(1):268–275, 2002.