Chapter 8

Conclusions

8.1 Introduction

In this chapter I conclude this thesis by summarising the main content of the dissertation in Section 8.2 and discussing my novel contributions and experimental findings in Section 8.3. I present in Section 8.4 several suggestions as to how the research presented in this thesis could be extended in the future, with a focus on directions that I consider to be potentially most fruitful.

8.2 Thesis Summary

In this thesis I explored the benefits of *learning* similarity preserving binary hashcodes for hashing-based approximate nearest neighbour (ANN) search. In Chapters 1-2, I described and motivated the problem of nearest neighbour search which involves finding the most similar data-point(s) to a query in a large database. We saw how this operation is truly fundamental in many diverse fields of study, from constructing noun similarity lists from web-scale textual datasets (Ravichandran et al. (2005)) to the automatic detection of thousands of object classes on a single machine (Dean et al. (2013)). The simplicity of this definition belies the considerable complexity of solving this search problem in a manner that does not require exhaustively comparing the query to every single data-point in the database. I specifically focused on the sub-field of approximate nearest neighbour search in this dissertation which encompasses a class of algorithms that seek to generate *similarity preserving* binary hashcodes for the query and database data-points. These binary hashcodes have the critical property of being more similar - that is sharing more bits in common - for more similar data-points. In Chapter 2 we

have seen how we can then obtain an attractive *constant query time* by using these binary hashcodes as the indices into the buckets of a set of hashtables and only returning colliding data-points as candidate nearest neighbours. This vast reduction in the search space is the primary advantage of hashing-based ANN search algorithms. However as I also discussed in Chapter 2 the faster query time comes at the cost of both a non-zero probability of failing to retrieve the closest neighbour in all cases and typically the requirement of relatively long hashcodes and multiple hashtables for an adequate level of precision and recall.

The original research presented in Chapters 4-7 introduced a suite of novel data-driven algorithms for improving the retrieval effectiveness of existing models for hashing-based ANN search at the cost of a one time offline training phase. My premise throughout this thesis was that learning the binary hashcodes by explicitly taking the *distribution of the input data* into consideration could yield much more compact and discriminative hashcodes that would improve search effectiveness over and above the state-of-the-art. In Chapter 2 I identified two key operations used by existing hashing-based ANN search algorithms to generate hashcodes both of which are amenable to data-driven learning: *projection* followed by *quantisation*. The projection step involves fracturing the input feature space into a set of disjoint polytope-shaped regions using a set of randomly drawn hyperplanes, with each region so-formed constituting a hashtable bucket. The hashcode for a data-point is therefore a geometric identifier specifying the unique region within which that data-point resides and can be computed by simply determining the position of the data-point with respect to each of the hyperplanes, with a '0' appended to the hashcode if the data-point is on one side of a hyperplane and a '1' otherwise. Computationally this operation involves two fundamental steps: a dot product (projection) onto the normal vector to each hyperplane followed by a thresholding (quantisation) operation on the resulting projected values.

I argued in Chapter 2 that these two operations - projection and quantisation - are responsible for the overall locality preserving quality of the hashcodes. For example, if a hyperplane happens to partition two true nearest neighbours or if a quantisation threshold separates the two related data-points then the hashcodes for those data-points will ultimately be more dissimilar, sharing less bits in common. In Chapter 2 it was further highlighted how Locality Sensitive Hashing (LSH) - a popular family of randomised algorithms for solving the ANN search problem - positioned the hashing hyperplanes randomly and the quantisation thresholds statically (at zero), *independent of the data distribution*. I hypothesised that the data-independent setting of these two crit-

ical parameters negatively impacted the retrieval effectiveness of hashing-based ANN search and I set out in Chapters 4-7 to investigate this claim.

Concretely, in Chapter 4, I introduced a novel algorithm for optimising one or more quantisation thresholds using an objective function that explicitly targeted the number of true nearest neighbours that are assigned the same bits. Rather than a single threshold placed statically at zero along a projected dimension I instead advocated a data-adaptive *optimisation* of *multiple* thresholds. In Chapter 5 this quantisation model was extended to automatically learn the *optimal quantity* of thresholds for each hyperplane based on a novel measure of *hyperplane informativeness*. Hyperplanes that better preserved the neighbourhood relationships between the data-points in the input feature space were rewarded with a greater allocation of thresholds. Locality preserving hyperplanes typically result in projections with more related data-points clumped together, a structure that can be better exploited with a finer quantisation granularity consisting of many thresholds. Having instilled a degree of data-dependence into the quantisation operation I turned my attention to learning the hashing hyperplanes in Chapter 6. I introduced a *three-step iterative algorithm* that utilised a small amount of pairwise supervision to guide the placement of the hashing hyperplanes in a way that attempted to avoid dividing regions of the space dense in true nearest neighbours. This model was further adapted to learn hyperplanes that generated similar hashcodes for similar data-points in *two different modalities*, such as images and text. Chapter 7 consolidated the research in this thesis by showing how the hashing hyperplanes and quantisation thresholds could be learnt as part of the same hashing model, thereby unifying the contributions put forward in Chapters 4-6. In all cases experimental analysis suggested my data-driven algorithms significantly improved retrieval effectiveness over incumbent models that set the hyperplanes or quantisation thresholds independently of the data distribution. I will outline the specific contributions made in each of these four chapters in Section 8.3.

8.3 Contributions and Experimental Findings

In this dissertation I have demonstrated that learning the hashing hyperplanes and quantisation thresholds in a task-specific manner can yield statistically significant improvements in hashing-based approximate nearest neighbour search effectiveness. To investigate this claim I set out to relax the previously ingrained assumptions of existing work regarding how the hashing hyperplanes and quantisation thresholds should be

constructed. The following three limiting assumptions were first identified in Chapter 1:

- A_1: Single static threshold placed at zero (for mean centered data)
- A_2: Uniform allocation of thresholds across each dimension
- A_3: Linear hypersurfaces (hyperplanes) positioned randomly

Each of these assumptions led to a new data-driven model that specifically sought to relax that assumption. Furthermore, with each model I empirically set out to test its retrieval effectiveness with respect to the best of prior-art in the field on the primary task of query-by-example image retrieval. I now summarise each model and the findings that arose from their experimental evaluation.

8.3.1 Learning Multiple Quantisation Thresholds

In Chapter 4 I contributed the first known semi-supervised multiple threshold learning algorithm for scalar quantisation in the context of hashing-based ANN search. In most existing work the projections are binarised by placing a single static threshold at zero along each projected dimension. The resulting binary bits are then used to construct the hashcodes for the data-points. My model permitted one or more thresholds to be optimised per projected dimension in addition to the application of any binary codebook to index the quantised regions. Furthermore, my quantisation model was unique in both its objective function and the manner in which this objective function was maximised. I proposed an objective function consisting of an F_1-measure supervised term that was interpolated with an unsupervised term that computed the compactness of the projections along a given dimension. The F_1-measure was computed using an adjacency graph that dictated the pairwise relationships between the data-points in the original feature space. True positives in this case were true nearest neighbour pairs falling into the same quantised regions, false negatives were true nearest neighbours pairs that fell into different regions and false positives were non nearest neighbours that fell into the same quantised regions. Given the non-differentiable nature of this objective function I advocated maximisation by stochastic search (simulated annealing and evolutionary algorithms). My experimental results arising from this research were many and varied and I will only attempt to outline the main findings here. Specifically, I demonstrated that:

- Retrieval effectiveness can always be increased by optimising the quantisation threshold(s) rather than statically placing the threshold at zero along a projected dimension (Chapter 4, Section 4.3.3.5).
- The optimum number of thresholds per projected dimension is projection function specific. For example, Locality Sensitive Hashing (LSH) generally preferred a single threshold while Principal Components Analysis (PCA) hashing benefited from two or more (Chapter 4, Sections 4.3.3.5-4.3.3.6).
- My semi-supervised objective function yielded the best retrieval effectiveness compared to state-of-the-art multi-threshold scalar quantisation models from the literature (Chapter 4, Section 4.3.3.8).

I confirmed that relaxing assumption A_1 is beneficial for retrieval effectiveness in the context of hashing-based approximate nearest neighbour search.

8.3.2 Learning Variable Quantisation Thresholds

In the second contribution of this thesis presented in Chapter 5, I highlighted the importance of learning the appropriate allocation of thresholds per projected dimension. The existing strategy of assigning the same number of thresholds to all projected dimension assumes that the corresponding hyperplanes are of equal locality preserving quality, yet this is frequently not true in real datasets. For example, a PCA hyperplane that captures a large proportion of the variance in the input feature space will tend to be much more discriminative than a hyperplane that captures a much lower proportion of the variance. I argued that the additional structure in the projected dimensions resulting from the more informative hyperplanes should be exploited with a quantisation of a finer granularity using multiple thresholds. This thesis presented, to the best of my knowledge, the first known research that identified and solved this problem in the context of hashing-based ANN search. I advocated the F_β-measure as an original way of quantifying the quality of a hyperplane. This F_β-measure was computed from a data-point adjacency graph with hyperplanes better preserving the neighbourhood structure between the data-points generally obtaining a higher F_β-measure. I introduced two new threshold allocation algorithms that used the computed F_β-measure scores to allocate thresholds to hyperplanes. Both algorithms sought to maximise the cumulative F_β-measure across hyperplanes but did so in two very different ways: one algorithm solved the binary integer linear programme using branch and bound, while the other

algorithm used a greedy approach that redistributed thresholds from the least informative to the most informative hyperplanes. The experimental evaluation demonstrated the following main findings:

- F_β-measure is a useful quantity for grading the locality preserving power of a hyperplane (Chapter 5, Section 5.3.3.2).
- Retrieval effectiveness can be increased significantly by learning a variable allocation of quantisation thresholds compared to a uniform allocation of thresholds across projected dimensions (Chapter 5, Section 5.3.3.2).

I demonstrated that relaxing assumption A_2 of existing work leads to significantly higher retrieval effectiveness for hashing-based approximate nearest neighbour search.

8.3.3 Learning the Hashing Hypersurfaces

My third and fourth contributions in Chapter 6 centered around the relaxation of assumption A_3. Existing models for hashing-based ANN search draw the hashing hyperplanes randomly within the input feature space. I contributed a new supervised projection function that instilled a degree of supervision into the placement of the hashing hypersurfaces. My contention was that a small amount of supervision would enable a better positioning of the hashing hypersurfaces in a way that encouraged more true nearest neighbours to fall within the same partitioned regions of the space. The projection function was an iterative three step algorithm reminiscent of the Expectation Maximisation (EM) algorithm. In the first step hashcodes of training data-points were smoothed using a data-point adjacency graph, which had the effect of setting the hashcode for each data-point to be the average of the hashcodes of its nearest neighbours as defined by the adjacency graph. This was my novel method for integrating supervision into the hypersurface learning procedure. In the next step a set of binary classifiers were learnt to predict the regularised bits with maximum margin. This step effectively positioned the hashing hypersurfaces within the space in a way that was consistent with the regularised bits: if two data-points shared a bit in common they were more likely to end up on the same side of the corresponding hypersurface. In the third step of the iterative algorithm the training data-points were re-labelled using the learnt hypersurfaces, which corrected the bits of any data-points that ended up on the wrong side of the hypersurfaces in the previous step. Iterating these three steps for a fixed number of iterations enabled the hypersurfaces to evolve into positions that

fractured the input feature space in a manner consistent with the supervisory signal. In my unimodal image retrieval evaluation, I made the following main experimental findings:

- Learnt hashing hyperplanes lead to significantly higher nearest neighbour retrieval effectiveness compared to hyperplanes that are placed randomly in the input feature space (Chapter 6, Section 6.3.3.4).
- Regularising hashcodes over a data-point adjacency graph is a more effective method of integrating supervision into the process of hyperplane learning than a Laplacian Eigenmap dimensionality reduction (Chapter 6, Section 6.3.3.6).
- Non-linear hypersurfaces induced by the radial basis function (RBF) kernel provide a more effective partitioning of the input feature space compared to linear hypersurfaces (hyperplanes) (Chapter 6, Section 6.3.3.8).
- The training and prediction (hashcode generation) time of the linear variant of my projection function was a fraction of the training time of competitive baseline models (Chapter 6, Section 6.5).
- My supervised projection function attained state-of-the-art retrieval effectiveness on standard image datasets, outperforming a large number of competitive data-dependent and independent hashing models from the literature (Chapter 6, Section 6.3.3.9).

The benefit of relaxing assumption A_3 was confirmed in the context of unimodal image retrieval in which the query and database are of the same feature type (e.g. SIFT features). I further extended this supervised projection function to learn hyperplanes that assigned similar hashcodes to similar data-points in two different modalities, such as text (e.g. TF-IDF vectors) and images. The extension to cross-modal hypersurface learning was surprisingly straightforward: I simply learnt another set of binary classifiers in the image feature space using the hashcodes of the textual data-points as targets. This had the desired effect of making the hypersurfaces in the visual feature space consistent, that is capable of assigning the same bits to similar data-points, with those in the textual feature space. Despite this simplicity I found the following encouraging results:

- Extending the three-step iterative hypersurface learning algorithm to cross-modal hashing yielded state-of-the-art retrieval effectiveness on standard cross-modal

datasets, outperforming a large selection of existing models in the field (Chapter 6, Section 6.6).

- Regularising hashcodes over a data-point adjacency graph is more effective for learning cross-modal hypersurfaces than solving an eigenvalue problem to obtain the hypersurfaces (Chapter 6, Section 6.6).

- The training time of my cross-modal projection function was a fraction of the time required by competitive baselines while having a similar prediction (hashcode generation) time (Chapter 6, Table 6.27).

Relaxing assumption A_3 was therefore also found to be beneficial for cross-modal retrieval effectiveness.

8.3.4 Learning Hypersurfaces and Quantisation Thresholds

In the final contribution of this thesis I conducted a preliminary exploration into the effect on retrieval effectiveness of learning both the hashing hypersurfaces and multiple quantisation thresholds jointly. This research presented in Chapter 7 combined the multiple threshold quantisation algorithms introduced in Chapters 4-5 with the iterative hypersurface learning algorithm presented in Chapter 6. In doing so I created a fully data-adaptive hashing pipeline of projection followed by quantisation. To connect both models I binarised the low-dimensional projections computed by the hypersurface learning algorithm using the multiple threshold quantisation algorithm. On the standard task of query-by-example image retrieval I made the following encouraging finding:

- Learning the hashing hypersurfaces and the quantisation thresholds as part of the same hashing model gives a retrieval effectiveness significantly greater than learning either parameter individually (Chapter 7, Section 7.3.3.1).

8.4 Avenues for Future Work

The novel contributions presented in this thesis have but only scratched the surface of this important and flourishing field of research and the potential scope for future research is both many and varied. I will attempt to highlight several potential future directions that I consider particularly promising in this last section.

8.4.1 Groundtruth and Evaluation Metric Correlation with Human Judgments

There has been little previous work that examines the extent to which the evaluation metrics and groundtruth used in the learning to hash field are sensible for learning hashcodes that correlate well with user search satisfaction. For example, ideally it should be the case that a significant increase in the area under the precision recall curve (AUPRC) should also lead to a significant increase in user satisfaction with the retrieved images or documents. Furthermore, in Chapter 3 I introduced the class-based and ε-NN based groundtruth definitions that were subsequently used to evaluate my models in Chapter 4-7. Many datasets of interest do not have manually assigned class labels, and so it would be useful to conduct a user-study as to how metric definitions of nearest neighbour groundtruth, such as the ε-NN groundtruth paradigm outlined in Chapter 3 Section 3.3.1, align with human judgements of item-item similarity. Ideally we would want many related data-points to a given query, as judged by a user, to be contained within the same ε-ball. For the class-based groundtruth used in Chapter 6 and outlined in Chapter 3 Section 3.3.2, this is less of an issue because those labels have been specifically assigned to the images by humans. The outcome of this user study would be expected to inform future developments in the evaluation procedures for hashing-based ANN search algorithms, and would be a valuable contribution to the community.

8.4.2 Online Learning of the Hashing Hypersurfaces

In Chapter 6 the hashing hypersurfaces were constructed in a batch fashion that assumed the entire training dataset would be immediately available for learning. As soon as the hypersurfaces were learnt they were never updated. This batch learning assumption is flawed when we consider many modern data sources of prime interest such as social media streams (e.g. Twitter). Twitter posts, for example, can be modelled as a never-ending, effectively infinite stream of data that could never be inspected in its entirety in a batch fashion (Petrović et al. (2010)). Furthermore streaming data sources are highly likely to exhibit a drift in the distribution of the data over time as, for example, new topics are discussed and the vocabulary changes. Simply learning a set of hypersurfaces once with no possibility of further updates would be an entirely suboptimal approach in this situation. It would be particularly interesting to adapt the hypersurface learning algorithm presented in Chapter 6 to the streaming data scenario

in which the hypersurfaces are capable of being updated in an online manner after each labelled pair of data-points are encountered in the stream. To achieve this goal one could potentially investigate the effectiveness of using passive aggressive (PA) classifiers (Crammer et al. (2006)) in place of the support vector machines (SVMs) used in this thesis. The PA classifier is particularly amenable to online learning and would make an ideal starting point for future research on this topic. To the best of my knowledge no online supervised projection function has so far been proposed for application to large-scale streaming data sources. I believe such a model would have significant potential impact in the field. An interesting challenge in this context would be how to efficiently update the hashcodes of existing data-points in the face of changing data. Furthermore, implementation of this model would address a second criticism of the work presented in this thesis, namely the application of the algorithms to datasets of medium size (1 million data-points or less) and of relatively low dimensionality ($D \leq 512$).

8.4.3 Hashing Documents Written in Two Different Languages

The cross-modal extension to my supervised projection function was only tested on images and textual data in this thesis, both of which were represented as low dimensional feature vectors. A particularly interesting extension to the work would involve exploring how the model could be adapted to hash *cross-lingual documents*, for example English and Spanish Wikipedia articles. In this task my goal would be to cluster related cross-lingual documents in the same hashtable buckets, without using any form of machine translation. In contrast to the image and text features used in this thesis multi-lingual document data sources are likely to be very high dimensional when encoded as TF-IDF vectors. The large freely available *parallel and comparable corpora*[1] consisting of similar documents written in different languages would provide the needed pairwise supervision for learning the hashing hypersurfaces, negating any tedious manual effort to obtain the required labels. The cross-lingual projection function could be directly compared and evaluated against Ture et al. (2011), a solution based on machine translation and traditional unimodal Locality Sensitive Hashing (LSH). Given the significant gains in retrieval effectiveness for the cross-modal experiments conducted in this thesis I have strong reason to suspect that cross-lingual hashing with a suitable adaptation of my graph regularised projection function would attract similar

[1] http://www.statmt.org/europarl/

gains in performance.

Given that more and more data on the Web is written in different languages I also foresee an online version of this cross-lingual projection function being particularly exciting future work. For example, a fast steaming algorithm for clustering similar tweets written in many different languages into the same hashtable buckets could prove useful to analysts in the financial industry or to linguists interested in studying the linguistic properties of Twitter and other related micro-blogs (Zanzotto et al. (2011)).

8.4.4 Dependent Hypersurfaces and Quantisation Thresholds

The multiple threshold quantisation models introduced in Chapters 4-5 positioned the quantisation thresholds *independently* across each projected dimension. In other words, the learning of the quantisation thresholds for one projected dimension was independent of the learning of the quantisation thresholds for another projected dimension. Inspired by the body of research into *multivariate discretisation* (Bay (2001)) a potential future avenue of research could examine the benefits of inducing a degree of *dependence* between the quantisation thresholds across projected dimensions. A particularly simple, albeit contrived example of a dataset that would not be quantised correctly by independently optimised thresholds is the two dimensional XOR dataset (Bay (2000)). In this case the quantisation algorithm would need to account for the correlation between the different feature dimensions in order to find the optimal positioning of the thresholds.

In a similar vein of research, the supervised projection function introduced in Chapter 6 constructed each hypersurface independently in a simple sequential fashion. Inducing a degree of dependence between the learning of the hypersurfaces might contribute to a reduced redundancy between bits while also permitting hypersurfaces learnt later in the sequence to focus on data-point pairs incorrectly classified by hypersurfaces learnt earlier in the sequence. A straightforward starting point would be to assign a weight to each pair in the adjacency matrix in a similar manner to the Adaboost algorithm (Schapire and Freund (2012)). True nearest neighbours assigned the same bits by earlier hypersurfaces could have their weight decreased while non-nearest neighbours assigned the same bits could have their weights increased. In this way the learning of the hashing hypersurfaces could be gradually biased to focus on data-points pairs that are more difficult to classify, potentially resulting in enhanced retrieval effectiveness.

8.4.5 Closer Integration of the Projection and Quantisation Operations

In Chapter 7, I demonstrated that combining projection function and quantisation threshold learning as part of the same hashing model can lead to significantly better retrieval effectiveness as compared to learning either in isolation. My approach involved a simple concatenation of my quantisation and projection models developed in Chapters 4-6. In effect, the hyperplanes were optimised first and then the quantisation of the projections were optimised during the second step of the two-stage pipeline, with both steps being performed independently, and without knowledge of the other. In future work it would be interesting to explore a combined objective function that both learns the optimal positioning of the hyperplanes while also simultaneously minimising the quantisation loss. This objective could be optimised in a single procedure, for example, by using existing gradient-based optimisers, to exploit synergies between the projection and quantisation operations to mutually influence and reinforce each other. Indeed, there has been recent evidence provided by Zhu et al. (2016) that such a tight coupling of projection and quantisation can lead to significantly better retrieval effectiveness over the standard image datasets considered in this dissertation.

8.5 Concluding Remarks

This thesis has explored the benefits of learning binary hashcodes for fast nearest neighbour search over large-scale datasets, with a particular focus on images. The experimental results have overwhelmingly indicated that significant increases in retrieval effectiveness can be obtained through data-aware hashcodes compared to their data-oblivious counterparts frequently employed in both industry and academia. I hope that the research presented in this dissertation contributes in some small way to the development of increasingly more effective and efficient algorithms for nearest neighbour search.