# Science Driven Innovations for Mobile Data Science: Theory, Practices and Lessons Learned

Deguang Kong *

**Abstract**

Recent years have witnessed the increasing popularity of mobile device due to the convenience that it brings to human lives. On mobile devices, rich user profiling data (including inter-app, intra-app and supply data) make it possible to provide much better recommendation services and further drive revenues from understanding users' behaviors in different segmentations. Intelligent personal assistant on device is highly desirable to provide accurate recommendation, ads targeting, and real-time image recognition, voice translations, etc. This paper presents the research efforts we have conducted on mobile device which aim to provide much smarter and more convenient services by leveraging data science, machine learning, optimization, deep learning and user profiling techniques. From different case studies, one can easily understand how science driven innovations help to improve the current services and provides business leadership in driving revenues. In the meantime, these research efforts have clear scientific contributions and are very promising in driving new growth in product.

**Keywords**: User engagement; Data Science; Recommendation; Targeting Model; Forecasting; Campaign Analysis; Deep Learning; CNN; Optimization

## 1 Introduction

New challenges come with the exponentially growing markets of mobile apps. We need to address many new problems, for example, diversified app markets, heterogeneous user behaviors and limited computational resources, in order to provide better service and improve user engagement on mobile devices. In the paper next, we will show our research efforts towards building smarter and more convenient mobile systems in the following aspects:

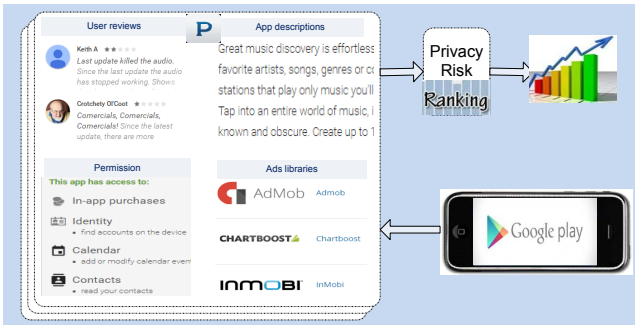


Figure 1: Ranking the risk of mobile apps using multi-modal features such as descriptions, user review, permission access and ads library.

- Mobile App Risk Assessment
- Mobile App Recommendation and Targeting
- Image Privacy on Mobile Devices
- Deep Learning on Mobile Devices
- User Profiling and Statistic Campaign Analysis for Samsung Apps

We provide scientific solutions and leaderships to solving these challenging problems because pure engineering can not work in practice. These techniques will be very helpful for solving mobile data science and AI problems in real world if (a big "if") the technology can be accurate and robust enough. We need research breakthrough to improve the state-of-the-art and we have diligently worked on them.

## 2 Mobile App Risk Assessment

Comparing with traditional software markets, markets like Google Play and Apple Store have lower entry threshold for developers and faster financial payback, hence greatly encouraging more and more developers to invest in this thriving business. Therefore controlling the quality of apps, especially the security risk of them across the whole markets, becomes an important issue to all that involved. On the other hand, public

*The author currently works at YAHOO! Research (701 First Avenue, Sunnyvale, 94089), and email is: doogkong@gmail.com. This work was based on the authors' scientific works published/submitted in public conferences during working at Samsung. Any opinions, findings or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of Samsung.

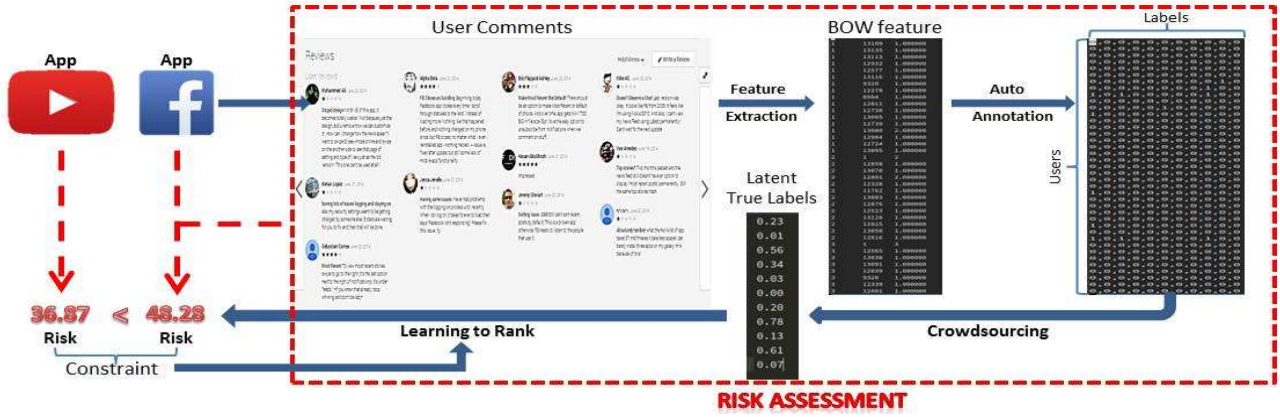


Figure 2: Flowchart of risk assessment from user comments. Given an app, the user comments are used to evaluate the risk of apps in two steps, (a) "crowdsourcing" is used to accumulate user comments into app-level features (shown as "feature extraction", "auto annotation" and "crowdsourcing"; (b) "learning to rank" model is used to predict the risk scores by utilizing the latent features, where pairwise contraints are enforced between pairwise apps (shown as relative scores of two apps).

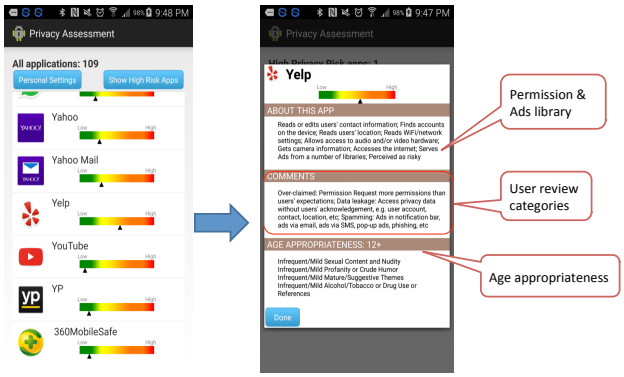


Figure 3: Demo of risk assessment.

concerns about privacy issues with online activity and mobile phones are also elevating, demanding a mobile environment with more respect to users' privacy.

In mobile apps, permissions indicate the resources that the apps can access, and thus can be viewed as a privacy indicator. From users' perspective, the meta data such as users' reviews and developers' descriptions reflect users' perceptions and developers' expectations for the apps, and thus are also correlated with risks of apps.

Our idea is to explore heterogeneous privacy indicators [8], [11] for app risk ranking, which, we believe, is very important to internet company to improve user engagement in mobile platforms (as shown in Fig. 1). The risk ranking problem is formulated as a mutli-view feature learning problem by exploring group LASSO and exclusive group LASSO techniques [10], [9], which can automatically select the most discriminant features by considering both inter-view feature competitions, and also intra-view feature correlations. In particular, we solve the following problem, given feature $x_i$ for each app $i$, $Y_{Ki}$ for label of app $i$ with category $k$, we aim to find the feature weight $w_k^v$ for class $k$ regarding $k$-th view feature, *i.e.*,

$$\min_{W \in R^{n \times K}} \quad \sum_{ik} (Y_{ki} log \sum_k e^{w_k^T x_i} - Y_{ki} w_k^T x_i) + \alpha \sum_k \sum_v \|w_k^v\|_2 + \beta \sum_k \sum_v \|w_k^v\|_1^2$$

Correspondingly, we derive an efficient iteratively re-weighted algorithm to tackle the resultant optimization problem, which can handle any group structure, regardless of coherent or exclusive group structures. It demonstrates very good performance in real-world datasets (totally 13, 174 apps, 34, 514 descriptions, 9, 986, 568 user reviews and 100 ads libraries).

We also derive a crowdsourcing ranking approach [2], [7] (see Fig.2) to rank risk of apps from user comments by combining feature learning and ranking SVM methods, which also provides good solutions in practice. The problem we solve is formalized as:

$$\min_{w \in R^d, \theta, Y_\ell} \quad -logPr(D_n|\theta, Y_\ell) - logPr(\theta) \\ +\lambda\|w\|^2 + C\|(e - BY_\ell w)\|_2^2.$$

where $w$ is the feature weight, $Y_\ell$ is the labels learned from the feature learning step and $\theta$ is the prior distribution of parameters in crowdsourcing process, and $lnPr(D_{n|})$ gives the likelihood of objective given current parameters while $\|(e - BY_\ell w)\|_2^2$ is the hinge-loss function in SVM ranking.

**Lessons Learned** We do need multi heterogenous models to find the most discriminant features. User reviews and ads libraries play import roles in understanding the risk of apps except the permissions. A demo system is shown in Fig.4.

## 3 Mobile App Recommendation and Targeting

As of July 2013, Google Play had over 1 million Apps with over 50 billion cumulative downloads, and the number of Apps has reached over 1.2 million in June 2014; as the beginning of June 2014, App Store had 1.2 million Apps and a cumulative of 75 billion downloads. Therefore, it is urgent to develop effective personalized App recommendation systems. In this section, we provide our works regarding app recommendation. Recommendation is useful since it strongly connects with target ads.

### 3.1 Privacy aware app recommendation

Recent years have witnessed a rapid adoption of mobile devices and a dramatic proliferation of mobile applications (Apps for brevity). However, the large number of mobile Apps makes it difficult for users to locate relevant Apps. Therefore, recommending Apps becomes an urgent task. Traditional recommendation approaches focus on learning the interest of a user and the functionality of an item (e.g., an App) from a set of user-item ratings, and they recommend an item to a user if the item's functionality well matches the user interest. However, Apps could have privileges to access a user's sensitive resources (e.g., contact, message, and location). As a result, a user chooses an App not only because of its functionality, but also because it respects the user's privacy preference. To the best of our knowledge, this work presents the first systematic study on incorporating both interest-functionality interactions and users' privacy preferences to perform personalized App recommendations [14]. Specifically, we first construct a new model to capture the trade-off between functionality and user privacy preference. In particular, in this work, it leverages the state-of-the-art Poisson factorization technique and optimizes the objective:

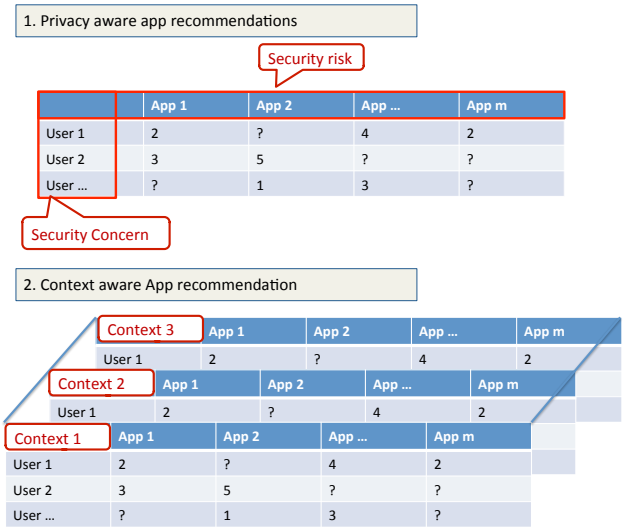


Figure 4: upper panel: privacy aware app recommendation; lower panel: context aware app recommendation.

$$\max_{u,v} Pr(y_{ij}|u_i, v_j, p_s) \\ = Poisson\Big(y_{ij}, u_i^T(v_j + \lambda \sum_{s \in \Sigma_j} p_s)\Big),$$

where $y_{ij}$ is the rating score for a particular user $i$ for app $j$, $u_i$ is the user $i$ latent factor, $v_j$ is the app $j$ latent factor and $p_s$ is app privacy latent factor w.r.t app $j$.

Then we crawled a real-world dataset (16, 344 users, 6, 157 Apps, and 263, 054 ratings) from Google Play and use it to comprehensively evaluate our model and previous methods. We find that our method consistently and substantially outperforms the state-of-the-art approaches, which implies the importance of user privacy preference on personalized App recommendations. Moreover, we explore the impact of different levels of privacy information on the performances of our method, which gives us insights on what resources are more likely to be treated as private by users and influence users' behaviors at selecting Apps.

### 3.2 Context aware app recommendation

In many practical applications, in practice, we require the recommendation depend on context. Here"context" is a very generic concept that can denote location, gender, age or other different segments. In other words, recommendation is performed on different buckets based on an attribute or a combination of a group of attributes. Similar to app recommendation, we solve this problem using tensor bilinear factorization technique [6]. In particular, we solve the following problem:

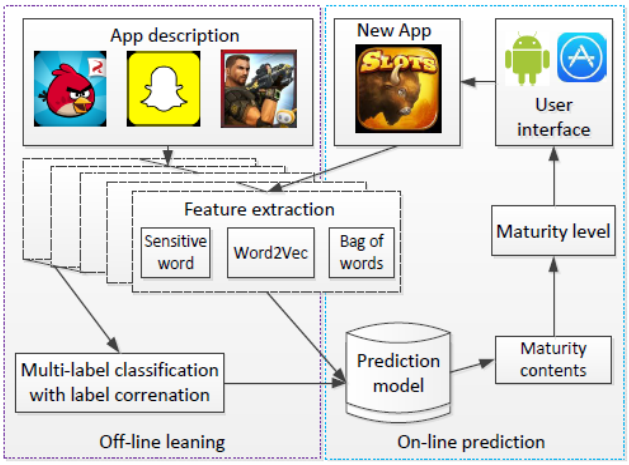


Figure 5: App maturity evaluation framework using deep learning (word2vec) and machine learning.

$$\max_{U,V,P} Pr(X_{ijk}|U_{ir}, V_{js}, P_{kt}) = Poisson\Big(X_{ijk}, U_{ir}V_{jr} + U_{it}P_{kt} + V_{js}P_{ks}\Big),$$

where $X_{ijk}$ is the rating score for a particular user $i$ for app $j$ in context $k$, $U_{i:}$ is the user $i$ latent factor, $V_{j:}$ is the app $j$ latent factor and $P_{k:}$ is app context latent factor for context $k$.

Similarly this framework can be easily extended for generating the context aware service recommendations. Here we give the use case:

| User | Location Semantics | recommended service |
|---|---|---|
| John | Safeway | Use Apple Pay |
| Damao | Bank | Have coffee |
| Amy | Mall | Consume coupons |

**3.3 App maturity rating framework** We also have other works about how to protect children from inappropriate content in mobile apps. Apps may contain sexual, violence and drug usage in their content. Therefore, mobile platforms provide rating policies to label the maturity levels of Apps and the reasons why an App has a given maturity level, which enables parents to select maturity-appropriate Apps for their children. However, existing approaches to implement these maturity rating policies are either costly (because of expensive manual labeling) or inaccurate (because of no centralized controls). In this work [4], we aim to design and build a machine learning framework to automatically predict maturity levels for mobile Apps and the associated reasons with a high accuracy and a low cost.

Specifically, we extract novel features from App descriptions by leveraging *word2vec* to automatically capture the semantic similarity between words and adapt Support Vector Machine to capture label correlations with pearson correlation in a multi-label classification setting. In particular, in *word2vec* step, given a sequence of training words $w_1, w_2, w_3, \cdots, w_T$, the skip-gram model is used to maximize the average log probability given representation $v$ using:

$$\max_{v} \frac{1}{T}\sum_{t=1}^{T}\sum_{-c\le j\le c, j\neq 0} logPr(w_{t+j}|w_t), \tag{3.1}$$

where $c$ is the size of training context. and $Pr(w_{t+j}|w_t)$ is usually defined using softmax function, *i.e.,*

$$Pr(w_O|w_I) = \frac{\exp\left((v'_{w_O})^T v_{w_I}\right)}{\sum_{w=1}^{W}\exp\left((v'_w)^T v_{w_I}\right)}, \tag{3.2}$$

where $v_w$ and $v'_w$ are the "input" and "output" vector representations of $w$ and $W$ is the number of words in vocalbulary.

Moreover, we evaluate our approach and various baseline methods using datasets that we collected from both App Store and Google Play. We demonstrate that, with only App descriptions, our approach already achieves 85% Precision for predicting mature contents and 79% Precision for predicting maturity levels, which substantially outperforms baseline methods.

**Lessons Learned**. In this section, we use "app" as demonstrated examples for recommendation purpose. Our approach can be easily adapted for recommendations on purchase and others.

## 4 Image Privacy on Mobile Devices

In this section, we present two works that help to protect image privacy on mobile devices.

**4.1 Image privacy protection via image perturbation** Every second, nearly 4,000 photos uploaded to Facebook, around 4,600 photos exchanged through Snapchat Photos are uploaded, saved and shared on cloud, *e.g.*, centralized photo sharing platforms (PSPs) In cloud side (PSP), sensitive regions are exposed to public, etc. What is the security and privacy risk? Photo owners worry about privacy leakage on cloud/PSPs, also Cloud/PSPs may access and process user photos without explicitly asking for users' agreement and share the unprotected photos. In this work we propose image perturbation technique to protect the image privacy. Ideally, given encryption function $E(.)$, transformation function $T(.)$, the goal is to find decryp-

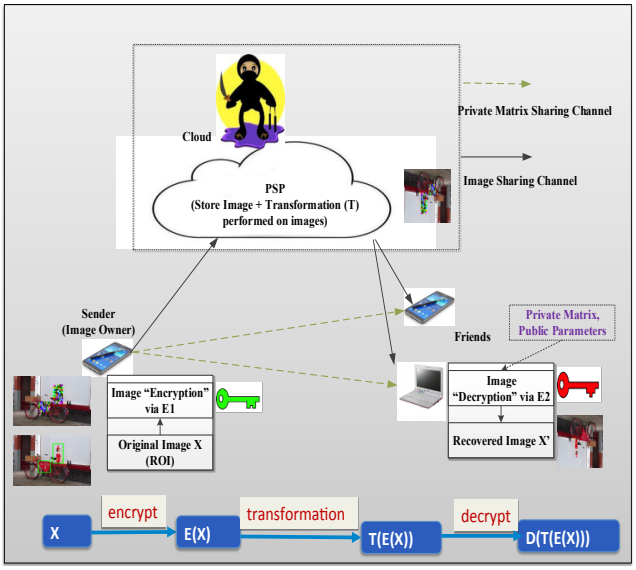


Figure 6: System framework of image privacy protection.

tion function $D(.)$, such that for an image $X$, it exists:

$$D\Big[T(E(X))\Big] = T(X). \tag{4.3}$$

Our system design is guided by the following theorem [3]. Let $P(.)$ be the image perturbation technique, we have:

THEOREM 4.1. *Using Image perturbation technique $E = P(.)$ for "encryption", it can exactly "decrypt"*

$$D\Big[T(E(X))\Big] = T(X),$$

*where $D = f(T, E)$ can be easily calculated given $E(.) = P(.)$ and $T(.)$.*

The system framework is shown in Fig.6. Note that "crypto" based technique (including symmetric and public key encryption) may not work since it is not compatible with transformation $T(.)$ and also $D$ is impossible to be computed given $E(.)$ and $T(.)$, and therefore $X$ can not be recovered. For exactly the same reason, differential privacy added laplacian noises to the image, which is, in fact, irreversible although privacy preserving. Finally we cannot recover anything given image transformation $T(.)$.

In our approach, it supports different linear transformations (e.g., Rotation, Cropping, Scaling) and also non-linear transformation such as compression. The key idea of our approach is to perturb DC and AC components discriminant in FFT domain, which achieves the same purpose as crypto but is compatible with transforms. Also, our approach has advantages due to its

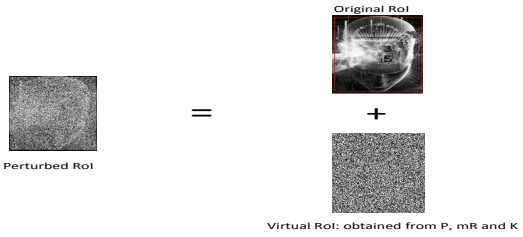


Figure 7: Image perturbation on "Google brain" image.


Figure 8: Private photos

simple, fast and effective implementation. In our solution, applying a transformation on perturbed image is equal to applying transformation on original image plus applying transformation on virtual image (generated from perturbation). An example is shown in Fig.7.

**4.2 Deep learning based techniques for protecting image privacy** This section provides a deep learning based technique to protect image privacy. Several examples are shown in Fig.8. Photo privacy is a very important problem in the digital age where photos are commonly shared on social networking sites and mobile devices. The main challenge in photo privacy detection is how to generate discriminant features to accurately detect privacy at risk photos. Existing photo privacy detection works, which rely on low-level vision features, are non-informative to the users regarding what privacy information is leaked from their photos. In this section, we propose a new framework called PrivacyCNH [15] that utilizes hierarchical features which include both object and convolutional features in a deep learning model to detect privacy at risk photos. In particular, given the joint deep learning structures $\mathcal{V} = \{V^1, V^2, V^3, V^4\}$ and $\mathcal{W} = \{W^1, W^2, \cdots, W^5\}$, the posterior probability of privacy risk for an image $i$ is:

$$Pr(y_i = 1|X_i; \mathcal{V}, \mathcal{W}) = \frac{1}{1 + exp(-z)}, \tag{4.4}$$

where

$$z = (V_k^4)^T h_4(X_i) + (W_\ell^6)^T \ell_5(X_i) + \beta, \tag{4.5}$$

where $V^i$ and $W^j$ are the CNN network structure parameters with script $i$ and $j$ indicating the layer number, $k$ indexes the hidden unit in layer $i$, $\ell$ indexes the hidden unit in layer $j$, $h_i$ and $\ell_j$ are the activation functions for object CNN and low-level CNN respectively and $\beta$ is the biased scalar term.

The generation of object features enables our model to better inform the users about the reason why a photo has privacy risk. The combination of convolutional and object features provide a richer model to understand photo privacy from different aspects, thus improving photo privacy detection accuracy. Experimental results demonstrate that the proposed model outperforms the state-of-the-art work and the standard convolutional neural network (CNN) with convolutional features on photo privacy detection tasks. Fig. 9 demonstrates the pipeline of our method.

**Lessons Learned**. It is good to use CNN to identify the privacy risks of images. However, we did not consider too much information regarding the object regions. As the next step work, we will leverage LSTM to further improve the performance of our method.

## 5 Deep Learning on Mobile Devices

CNN model has been widely and successfully used in many computer vision tasks, such as object detection, fine-grained image classification, age estimation, etc. The popularity of mobile phone brings the great convenience to people's life due to the existence of many practical and excellent apps. However, to run CNN models (even in testing phrase) for a typical vision task is a luxury for most devices due to the high computational cost and limited memory space and power resources. To accelerate CNN models is highly desirable to facilitate mobile vision applications that highly depends on the performance of CNN models.

Our investigation on AlexNet indicates that not only full-connected layers and convolution layers consume a lot of time, but also some non-tensor layers (such as Pooling layer and LRN layers) that do not contain any high-order tensor-type weight parameter are also time-consuming. However, current researches focus on achieving fast speed and/or less storage by making low rank approximation or parameter compression in full-connected and convolution layers. Although helpful, the acceleration and compression of non-tensor layers are totally ignored.

To address this limitation, this paper [13] proposes a unified framework to compress CNN models by dismembering non-tensor layers, to simultaneously accelerate the CNN model testing performance with neglect performance degradation. With re-trained new network parameters in "re-birth" layers, the functionality of non-tensor layers are equivalently implemented in the new merged layers with significant efficiently improvement. The standard least square error is used to minimize the error function in re-training process where the new parameters are essentially the "quantized" old parameters (in some sense). The framework includes both "streaminig merge" and "branch merge" that is able to conduct fast computations easily adapted for current mainstream CNN models and potential new CNN pipelines. In the meantime, in order to run deep learning on mobile devices, we provide an "elastic" approach to run deep learning in a distributed fashion (shown in Fig.10).

**Theoretical Analysis** The convolution layer transforms the input feature map $X \in \mathbf{R}^{M\times N\times K} \to Y \in \mathbf{R}^{M'\times N'\times K'}$, *i.e.*

$$\begin{aligned} & f_{\text{conv}} : X \mapsto Y, \\ & Y_{i'j'k'} = \sum_{i=1}^{d_k}\sum_{j=1}^{d_k}\sum_{k=1}^{K} W_{ijkk'} X_{i+i',j+j',k} \\ & (1 \le k' \le K') \end{aligned} \tag{5.6}$$

where $K, K'$ are the number of feature map channels, and $M, N; M', N'$ are the size of the images, which is actually regular linear convolution by a filter bank, $d_k \times d_k$ is the kernel size and feature map $Y$ is essentially the sum of inner product by traversing along different locations with $d_k \times d_k$ kernel (e.g., $d_k = 3$) and the output response $Y$is obtained by enforcing linear transformation $W$ on feature map $X$.

The local response normalization (LRN) layer performs "lateral inhibition" based on the fact that the activated neurons will have impact on those neurons in its local input regions. Therefore, it usually performs normalizing over local input regions from $\mathbf{R}^{M\times N\times K} \to \mathbf{R}^{M\times N\times K'}$, *i.e.,*

$$\begin{aligned} & f_{\text{LRN}} : X \mapsto Y, \\ & Y_{ijk'} = \frac{X_{ijk}}{\left(\kappa + \alpha \sum_{k\in G(k')} X_{ijk}^2\right)^\beta}, \end{aligned} \tag{5.7}$$

where $G(k) = \left[k - \lfloor \frac{\rho}{2} \rfloor, k + \lceil \frac{\rho}{2} \rceil\right] \cap \{1, 2, \ldots, K\}$ is a group of $\rho$ consecutive feature channels in the input map. Clearly, if $\kappa = 0, \alpha = 1, \beta = 1$, this gives $\ell_2$

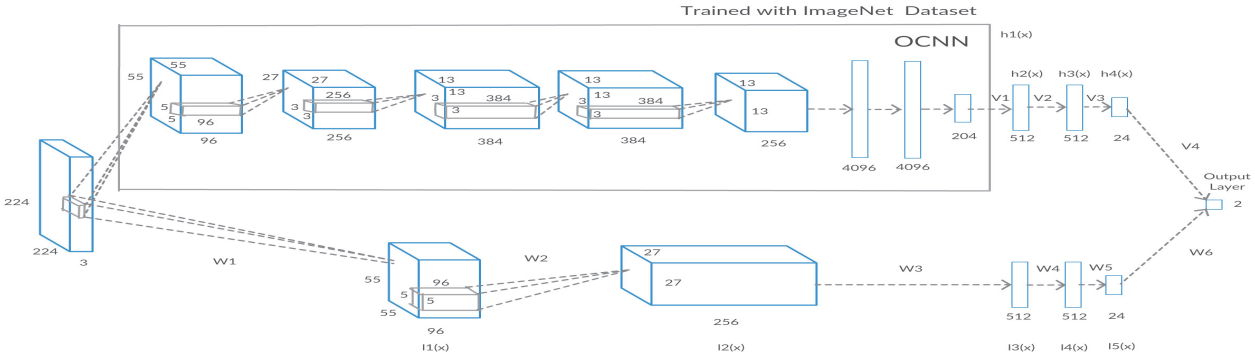


Figure 9: CNN pipeline for privacy detection that consists of two pipelines: (a) object feature learning pipeline (upper panel); (b) convolution feature learning pipeline (lower panel). $h_i(x), \ell_j(x)$ are activation functions.

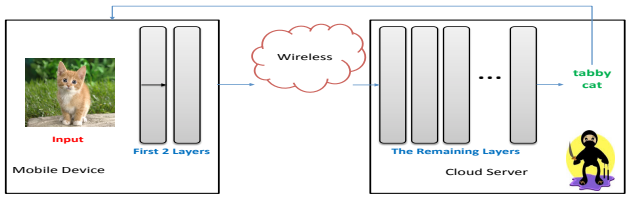


Figure 10: Distributed deep learning on mobile devices and cloud.

normalization. A batch normalization operation [5] is usually applied to change the distributions of activations to avoid "Internal covariate shift".

To achieve the desired functionality with acceleration, the idea is to find a mapping function $\mathcal{F} : X \in \mathbf{R}^{M\times N\times K} \rightarrow Y \in \mathbf{R}^{M''\times N''\times K'}$ such that it can get the same feature map value $Y^i$ given the same input feature map $X^i$ for any image $i$. Recall that convolution operation can be viewed as enforcing linear transformation $W$ on the input feature maps in the fully connected layers, and therefore we aim to build a single convolution operation $(*)$ that replaces several non-tensor layers by setting a new optimization goal, *i.e.,*

$$(5.8)\forall i: \quad Y^i = Y^i_{\text{COM}}; \quad Y^i_{\text{COM}} \simeq \hat{W} * X^i + \hat{b};$$

While the type and sequence of functions is usually handcrafted, the parameters $W$ and bias $b$ can be learned from our experiments for solving a least square problem using SGD, *i.e.,*

$$(\hat{W}^*, \hat{b}^*) = argmin_{\hat{W},\hat{b}} \sum_i \|Y^i_{\text{COM}} - (\hat{W} * X^i + \hat{b})\|^2,$$

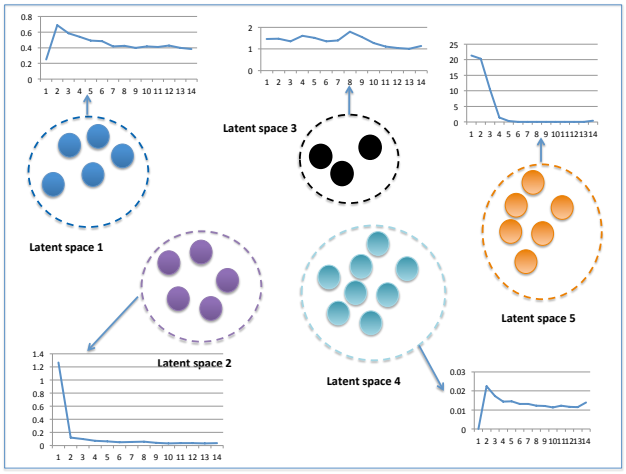


Figure 11: User segmentation examples based on time-varying features.

## 6 User Profiling and Statistic Campaign Analysis for Samsung Apps

In this section, we show our efforts towards improving user engagement on Samsung Apps. Due to management issues, this will be illustrated in detail somewhere else. In this section I give a big picture of what have done from *research* perspective.

Q1: How to measure different users similarity given temporal data?

*A1: We leverage pearson correlation, feature correlation [12] and hashing to compute the similarity (for big data). Fig.11 gives a brief illustration on segmentation [1] of temporal user behavior data.*

Q2: How to measure market campaign effectiveness?

*A2: We leverage chi-square test, propensity-*

*adjusted regression model for casual inference.*

Q3: How to build user profiling?

*A3: We achieve this in this following steps:*

- *1: User segmentation and targeting based on multiple attributes: age, gender, device, geo, search, etc.*
- *2: Audience look-alike model*
- *3: Audience volume and performance forecasting*
- *4: Hierarchical models and large scale optimization in different landscapes.*

For more details, please refer to the forthcoming work.

## 7 Conclusion

This paper presents our research efforts on mobile data science, which provides a scientific approach to drive innovations on different mobile applications. This work also shows how to apply machine learning and optimization techniques to solve the real-world challenging problem on mobile devices. The future works include building a more robust and intelligent mobile device echosystem and delivery of more products driven from scientific innovations.

**Acknowledgement**. All of the paper contents are based on the published papers (or technical report) during working at Samsung. Thanks for all co-authors, Xiaolong Wang, Dawei Li, Bin Liu (USC), Xuan Bao, Bing Hu, Shiva Kasiviswanathan, Huijun Xiong, Na Wang, Lei Cen, Bin Liu (Rutgers), Neil Gong, Jing Wang, Lam Tran, Wei Yang, Jianping He, Jian Huang, Pengfei Hu and Hongxia Jin.

## References

[1] X. Bao, B. Liu, B. Tang, B. Hu, D. Kong, and H. Jin. Pinplace: associate semantic meanings with indoor locations without active fingerprinting. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing, UbiComp 2015, Osaka, Japan, September 7-11, 2015*, pages 921–925, 2015.

[2] L. Cen, D. Kong, H. Jin, and L. Si. Mobile app security risk assessment: A crowdsourcing ranking approach from user comments. In *Proceedings of the 2015 SIAM International Conference on Data Mining, Vancouver, BC, Canada, April 30 - May 2, 2015*, pages 658–666, 2015.

[3] J. He, B. Liu, D. Kong, X. Bao, N. Wang, H. Jin, and G. Kesidis. PUPPIES: transformation-supported personalized privacy preserving partial image sharing. In *46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2016, Toulouse, France, June 28 - July 1, 2016*, pages 359–370, 2016.

[4] B. Hu, B. Liu, N. Z. Gong, D. Kong, and H. Jin. Protecting your children from inappropriate content in mobile apps: An automatic maturity rating framework. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19 - 23, 2015*, pages 1111–1120, 2015.

[5] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 448–456, 2015.

[6] D. Kong. Context aware recommendation via tensor factorization. In *Samsung Technical Report 2015*, 2015.

[7] D. Kong, L. Cen, and H. Jin. AUTOREB: automatically understanding the review-to-behavior fidelity in android applications. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-6, 2015*, pages 530–541, 2015.

[8] D. Kong, L. Cen, and H. Jin. Mobile app security risk assessment using heterogeneous learning. In *Samsung Technical Report 2015*, 2015.

[9] D. Kong and C. H. Q. Ding. Efficient algorithms for selecting features with arbitrary group constraints via group lasso. In *2013 IEEE 13th International Conference on Data Mining, Dallas, TX, USA, December 7-10, 2013*, pages 379–388, 2013.

[10] D. Kong, R. Fujimaki, J. Liu, F. Nie, and C. H. Q. Ding. Exclusive feature learning on arbitrary structures via \ell_{1, 2}-norm. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 1655–1663, 2014.

[11] D. Kong and H. Jin. Towards permission request prediction on mobile apps via structure feature learning. In *Proceedings of the 2015 SIAM International Conference on Data Mining, Vancouver, BC, Canada, April 30 - May 2, 2015*, pages 604–612, 2015.

[12] D. Kong, J. Liu, B. Liu, and X. Bao. Uncorrelated group LASSO. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 1765–1771, 2016.

[13] D. Li, X. Wang, and D. Kong. Deep learning on mobile devices. In *Samsung Technical Report 2016*, 2016.

[14] B. Liu, D. Kong, L. Cen, N. Z. Gong, H. Jin, and H. Xiong. Personalized mobile app recommendation: Reconciling app functionality and user privacy preference. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, WSDM 2015, Shanghai, China, February 2-6, 2015*, pages 315–324, 2015.

[15] L. Tran, D. Kong, H. Jin, and J. Liu. Privacy-cnh: A framework to detect photo privacy with convolutional neural network using hierarchical features. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 1317–1323, 2016.