Mixed Mode Cascaded Classification Models

Colin Ponce

December 17, 2010

Abstract

We present here Mixed Mode Cascaded Classification Models, an algorithmic framework that seeks to effectively solve a wide range of machine learning tasks in a "plug and play" manner. It does this by sharing predictions between machine learning tasks, thus giving each task additional high-level information that can be used to solve its specific problem. We consider here a specific implementation of this framework in which we combine the machine vision tasks of scene categorization and depth estimation. In addition, we consider the use of a Poisson distribution for depth estimation. Experimental results are provided.

1 Introduction

Much work has been done in the past to develop effective machine learning algorithms that solve specific machine learning tasks by combining information from related problems, such as in [3]. However, algorithms such as these are laborious to design, and are only effective at solving specific problems within machine learning. Recently, new techniques have been developed that can combine machine learning tasks automatically, without requiring the researcher to design specialized frameworks.

In 2008, Heitz et. al. [1] introduced the concept of a Cascaded Classification Model. A Cascaded Classification Model (CCM) is an algorithmic framework that exploits existing classifiers for the individual sub-tasks. It consists of multiple layers, each layer containing an instance of each sub-task classifier. The features used for a given task are concatenated to the output of the classifiers in a given layer and used as the input for that task's classifier in the next layer.

The intuition behind this model is that knowledge about different sub-tasks can be useful when trying to perform a particular sub-task. For example, if we know that we are looking at a city scene, then it is unlikely that the image contains any cows, and this is useful when trying to perform cow detection. A diagram of CCM can be seen in Figure 1.

Then in 2010 Li. et. al. introduced Feedback-Enabled Cascaded Classification Models (FE-CCM). An FE-CCM is a CCM consisting of only two layers, but exploiting the added intuition that the first layer classifiers don't really need to be optimizing to their stated task, as their output is never tested for accuracy. Thus, by altering the the training data labels for the first-layer classifiers, those classifiers can focus on optimizing to aspects of the image that are most useful to the second-level classifiers. In this way we obtain the feedback-enabled CCM.

In this paper, we present a "Mixed Mode CCM" (MM-CCM). A MM-CCM is also a two-layer CCM, but makes use of the intuition that a classifier is likely to be able to perform its task more accurately on a specific type of image, rather than on all types of images together. Therefore, by creating multiple instances of learners for each task in the first level, called "contributors", and having them emphasize different data during training, we can create several different learners that

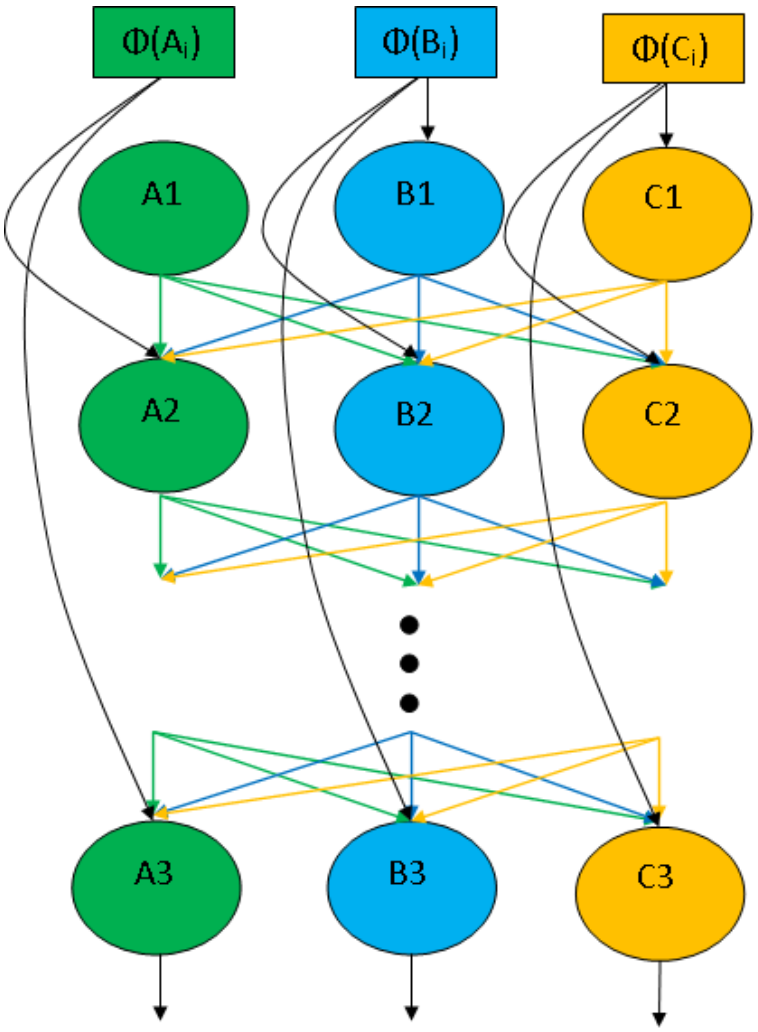

Figure 1: Cascaded Classification Models

each perform well on different subsets of the data. We can use a secondary learner, called a "mixer", to determine how best to combine these outputs to perform well on both sets of data together.

This mixing technique can be viewed as a form of extrapolated local regression. It is local because each contributor emphasizes the data points that matter most and trains based on that weighting. It is extrapolated because the weight of each data point for each contributor is determined by the mixer; because this mixer is itself a machine learner, it can effectively extrapolate how best to weigh a new, unseen data point.

In addition to this, we explore here the use of a modified Poisson distribution for depth prediction. Also, as input to the second layer depth estimator, for each pixel, we include the first layer output of the nearby pixels. This is an attempt to use MM-CCM to capture the correlation that is usually captured by Markov Random Fields.

In this paper we utilize MM-CCM to combine multiple tasks in machine vision. In particular, we perform experiments combining the two tasks of scene categorization and depth estimation. However, MM-CCM is a general algorithm that can be applied to any domain with multiple, related machine learning problems.

The rest of the paper is organized as follows. In Section 2 we describe the MM-CCM model in

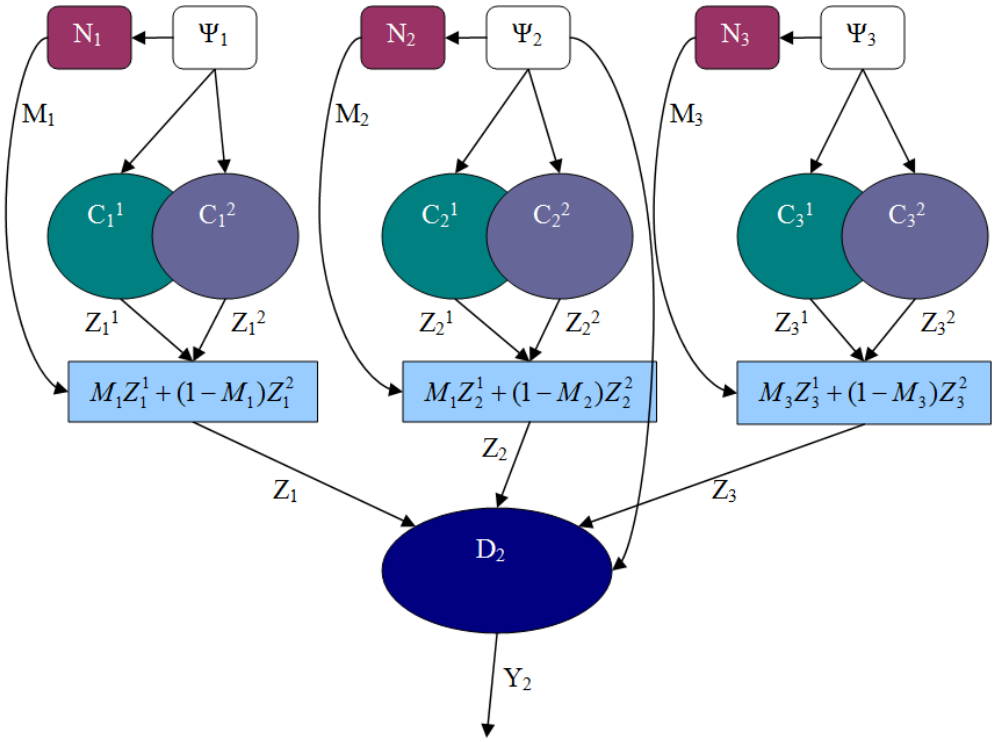

Figure 2: Mixed Mode CCM. Feedback lines are not drawn here.

detail and derive the relevant equations. In Section 3 we perform experimental tests of MM-CCM. Section 4 concludes.

2 Mixed-Mode Cascaded Classification Models

We will describe the MM-CCM model here. We will derive the equations assuming two contributors for each level-1 task, but it can be easily extended to k Suppose there are n related subtasks, and two level-1 classifiers for each subtask $i \in \{1, ..., n\}$. Then let C_i^j denote the level-1 classifier for task i, where $j \in \{1, 2\}$ denotes a specific instance of the classifier for task i. Let $\Psi_i(X)$ be the original features used for task i given an image X. Let θ_i^j represent the parameters for C_i^j, and Z_i^j the output of the classifier C_i^j. Furthermore, let D_i denote the level-2 classifier for task i, along with parameters ω_i and output Y_i.

Finally, we also have a system for determining how the individual level-1 classifier for a given task are mixed before being sent to the second layer. For each task i, let N_i denote a *mixing node* with parameters β_i that takes input $\Psi_i(X)$ and output a mixing value M_i. There are multiple methods for mixing different instance of a level-1 task; currently we perform mixing via linear interpolation:

$$Z_i = M_i Z_i^1 + (1 - M_i) Z_i^2, \quad M_i \in [0, 1].$$

This value is then sent to the level-2 classifiers as an input feature. A diagram of this process can be seen in Figure 2.

Now, we wish to determine how best to optimize our parameters, and how best to feed back

information to the level-1 classifiers. That is, we wish to optimize the log-likelihood equation

$$\ell(\theta^1, \theta^2, \omega, \beta) = \log \prod_{X \in \Gamma} P(Y_1, ..., Y_n | X; \theta, \omega, \beta), \tag{1}$$

where $\beta = \{\beta_1, ..., \beta_n\}$, $\omega = \{\omega_1, ..., \omega_n\}$, and $\theta^j = \{\theta_1^j, ..., \theta_n^j\}$ for $j \in \{1, 2\}$.

Now, the parameters that we can feed back to the level-1 classifiers are the Z_i^j's and the M_i's. Therefore, we wish to maximize the log-likelihood jointly with these latent variables:

$$\ell(\theta^1, \theta^2, \omega, \beta) = \sum_{X \in \Gamma} \log \sum_{Z^1, Z^2, M} P(Y_1, ..., Y_n, Z^1, Z^2, M, | X; \omega, \theta^1, \theta^2, \beta), \tag{2}$$

where $Z^j = \{Z_1^j, ..., Z_n^j\}$ and $M = \{M_1, ..., M_n\}$. This can be rewritten as

$$\sum_{X \in \Gamma} \log \sum_{Z^1, Z^2, M} \prod_{i \in \{1,...,n\}} P(Y_i | \Psi_i(X), Z^1, Z^2, M; \omega_i) P(Z_i^1, Z_i^2, M_i | \Psi_i(X); \theta_i^1, \theta_i^2, \beta_i) \tag{3}$$

$$= \sum_{X \in \Gamma} \log \sum_{Z^1, Z^2, M} \prod_{i \in \{1,...,n\}} P(Y_i | \Psi_i(X), Z^1, Z^2, M; \omega_i) P(Z_i^1 | \Psi_i(X); \theta_i^1)^{M_i}$$
$$P(Z_i^2 | \Psi_i(X); \theta_i^2)^{(1-M_i)} P(M_i | \Psi_i(X); \beta_i). \tag{4}$$

Note that here we exponentiate the factor $P(Z_i^1 | \Psi_i(X); \theta_i^1)$ by its mixing value M_i. This captures the intuition that when M_i is close to 0, we care little about Z_i^1 but much about Z_i^2.

2.1 Feed Forward

We would like to jointly maximize the likelihood of the parameters and variables $\theta^1, \theta^2, \beta, \omega, Z^1, Z^2, M$. However, this maximization problem is nonconvex. Therefore, as in FE-CCM, we iterate between a feedback step and a feed forward step. The feed forward stage maximizes the likelihood of ℓ with respect to the parameters $\theta^1, \theta^2, \beta, \omega$, while the feedback stage maximizes the likelihood with respect to the latent variables Z^1, Z^2, M.

Let us consider the feed forward stage here. We assume that the variables Z^1, Z^2, M are fixed. The maximization problem then becomes

$$\max_{\theta^1, \theta^2, \omega, \beta} \sum_{X \in \Gamma} \log \prod_{i \in \{1,...,n\}} P(Y_i | \Psi_i(X), Z^1, Z^2, M; \omega_i) P(Z_i^1 | \Psi_i(X); \theta_i^1)^{M_i}$$
$$P(Z_i^2 | \Psi_i(X); \theta_i^2)^{(1-M_i)} P(M_i | \Psi_i(X); \beta_i), \tag{5}$$

which is easily separable into the subproblems

$$\max_{\theta_i^1} \sum_{X \in \Gamma} \log P(Z_i^1 | \Psi_i(X); \theta_i^1)^{M_i} \tag{6}$$

$$\max_{\theta_i^2} \sum_{X \in \Gamma} \log P(Z_i^2 | \Psi_i(X); \theta_i^2)^{(1-M_i)} \tag{7}$$

$$\max_{\beta_i} \sum_{X \in \Gamma} \log P(M_i | \Psi_i(X); \beta_i) \tag{8}$$

$$\max_{\omega_i} \sum_{X \in \Gamma} P(Y_i | \Psi_i(X), M, Z^1, Z^2; \omega_i) \tag{9}$$

for every $i \in \{1, ..., n\}$. These problems can each be solved individually.

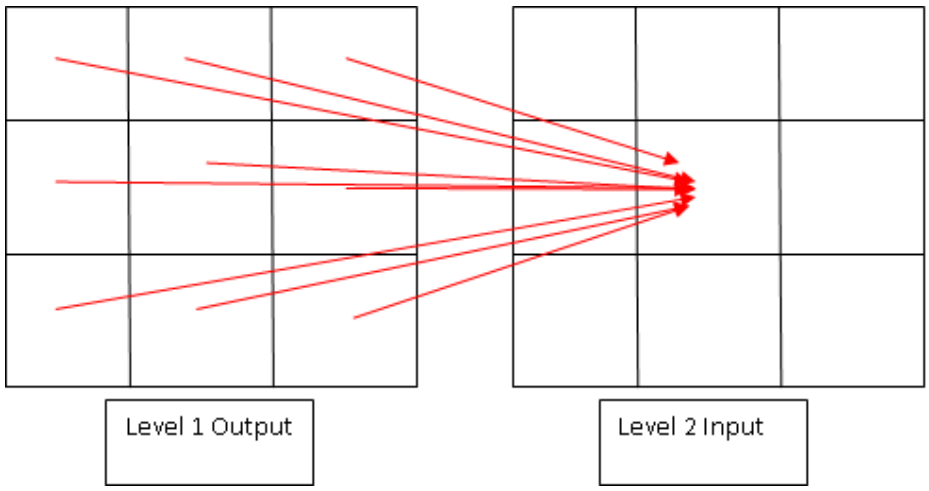

Figure 3: An illustration of sending neighboring pixels in the level 1 depth output as input to the level 2 depth estimator.

2.2 Feedback

To perform the feedback, we wish to update to latent variables Z^1, Z^2, M. For each training example, we wish to maximize the quantity

$$\log \prod_{i \in \{1,\ldots,n\}} P(Y_i|\Psi_i(X), Z^1, Z^2, M; \omega_i) P(Z_i^1|\Psi_i(X); \theta_i^1)^{M_i} \\ P(Z_i^2|\Psi_i(X); \theta_i^2)^{(1-M_i)} P(M_i|\Psi_i(X); \beta_i) \tag{10}$$

$$= \sum_i \log P(Y_i|\Psi_i(X), Z^1, Z^2, M; \omega_i) + M_i \log P(Z_i^1|\Psi_i(X); \theta_i^1) \\ +(1 - M_i) \log P(Z_i^2|\Psi_i(X); \theta_i^2) + \log P(M_i|\Psi_i(X); \beta_i). \tag{11}$$

We do this by assuming that the probability distribution of a given Z_i^j or M_i is either a Gaussian or a Poisson (see below). Unfortunately, under that assumption, this equation is not convex, and so cannot be easily optimized. To make up for this, we apply a hard Expectation Maximization approach [4]. We first assume that the variables M_i are fixed, and maximize relative to Z_i^1, Z_i^2. We then assume that the Z_i's are fixed and maximize relative to the M_i's. When this is done, the problem breaks down such that each iterative step can be solved by convex optimization.

2.3 Depth Prediction

To make depth predictions, we utilized not a Gaussian distribution, but a Poisson. The intuition behind this is that we care more about predicting depth correctly on a log scale than we do on a linear scale. For example, if the true depth of a pixel is 80 feet and we predict 79 feet, that is not nearly as bad as if the true depth is 2 feet and we predict 1. A Poisson distribution's variance is equal to its mean, and so it captures this intuition.

However, a Poisson distribution is a discrete distribution over the nonnegative integers, and depth value are continuous. So, we simply took the equation for a Poisson distribution, and took it

	Scene Accuracy (%)
Independent	79.4082
Mixed	79.2498
CCM, Poisson depth	78.2389
MM-CCM, Poisson depth	77.9385
CCM, linear depth	80.0415
MM-CMM, linear depth	**80.0574**

Table 1: Comparison of different mixing algorithms using scene accuracy. Independent is simple ridge regression. Mixed is two ridge clusters mixed. CCM and MM-CCM in this case represent zero feedback iterations.

continuouly. That is, we took the distribution as

$$P(y, \lambda) = \frac{\lambda^y e^{-\lambda}}{\Gamma(y+1)},$$

where $\Gamma(x)$ is the Gamma function, an extension of the factorial function, with it argument translated down by 1, to real numbers. This distribution can also be viewed as a special case of the Gamma distribution, though the parameters and variables in a Gamma distribution are switched.

Now, Markov Random Fields typically capture the idea that nearby pixels in an image are related to each other. We capture this same idea by including as input to the second layer depth learner not only the output of its own pixel prediction from the first layer, but also that of its eight neighboring pixels (Figure 3).

3 Testing

The project built on MATLAB code originally used to test the FE-CCM models provided by Congcong Li. Tests were performed using a 2-task construction in which scene classification is the first task and depth estimation is the second. The dataset used for scene categorization was the MIT outdoor scene dataset [5], and the depth estimation dataset used was the Make3D Range Image dataset [6].

3.1 Mixing

In our first set of experiments, we specifically tested the benefits of mixing. Thus, we trained with two tasks (scene and depth) without feedback. The results of this can be seen in Table **??**.

As can be seen from the results, the mixing technique shows a moderate improvement over the other algorithms. Somewhat surprisingly, simple ridge regression performs second best in regards to scene categorization. This may be because the second layer scene learner uses 1-norm regularization rather than 2-norm, and so may have difficulty utilizing all of its features, if many of them are useful. In addition, MM-CCM seems to show some improvement over other algorithms in depth estimation as well.

3.2 Modeling depth with Poisson distributions

In our next set of experiments, we tested the benefits of modeling depth with a Poisson distribution. As can be seen from Table **??**, using a Poisson distribution for depth is actually detrimental

	(A) RMS error	(B) MFE error	(C) Logarithmic error
Independent Linear	**16.5273**	0.85571	0.63925
Independent Poisson	18.094	0.85156	0.23236
CCM Poisson w/o Neighbors	17.049	**0.78304**	0.22795
CCM Poisson w/ Neighbors	17.7404	0.80453	**0.22162**

Table 2: Comparison of different algorithms for depth estimation. Independent linear and independent Poisson both use a regularization constant of 1. All depth predictions are capped to the maximum 80 and the minimum 1. No feedback in CCM. (A) Root-mean-square error (RMS) is root-mean-square error. (B) Mean fractional error (MFE) is $\sum_{i=1}^{n} |\hat{y}_i - y_i|/y_i$. (C) Logarithmic error is $\sum_{i=1}^{n} |\log_{10} \hat{y}_i - \log_{10} y_i|/n$.

Feedback Iterations	MM-CCM Accuracy (%)	FE-CCM Accuracy (%)
0	78.2818	**79.6931**
1	77.5973	78.195
2	78.0601	78.1378
3	78.0616	76.5464
4	78.1762	75.9669
5	78.1502	77.8228

Table 3: Scene categorization results, comparing MM-CCM and FE-CCM.

when using it as a first-layer input to a scene classifier. Table **??** shows results of using a Poisson distribution for depth estimation.

As can be seen, linear regression is still most effective at reducing RMS error. This is not surprising, as RMS error is measured in a linear fashion rather than a logarithmic fashion. Poisson regression is clearly superior in regards to logarithmic error. This was the intended effect, and the intuition behind using Poisson regression in the first place. In addition, note that including nearby pixels in the CCM feed forward does seem to have a positive effect on depth estimation.

3.3 Feedback

In our next set of experiments, we utilized mixing, Poisson distributions, and feedback all together for scene prediction. The results can be seen against the validation set in Table 3. For this experiment, we trained each of the first level classifiers initially on the original scene of depth data. However, in future iterations, we did not use the actual depth data at all, but instead trained the level 1 depth learner on images from the scene dataset using depth labels produced by the feedback mechanism.

The results are compared against the FE-CCM model. As can be seen, the best accuracy actually occurs with zero feedback iterations. The same is true with FE-CCM, and FE-CCM performs better than MM-CCM there. We believe that this is because the mixture model performs best when the data consists of several different clusters, each of which is best represented by a different separating hyperplane. It seems that the scene dataset does not behave in this way, and so the mixture does not perform its best.

Finally, we tested the effectiveness of feedback in depth estimation, making use of all modifications discussed in this paper. The results can be seen against the validation set in Table 4. It

Feedback Iterations	RMS	MFE	Logarithmic
0	17.7404	0.80453	0.22162
1	**17.2776**	**0.75242**	**0.21315**
2	17.3955	0.76604	0.2167
3	17.4628	0.77097	0.21801
4	17.4809	0.77642	0.21857

Table 4: Depth estimation results.

seems that some feedback is useful, though after a single iteration, more feedback starts to worsen the results.

4 Conclusion

We have developed here a number of interesting modifications to the CCM framework. It is interesting to note that, with the exception of including neighboring pixels as input to the level-2 depth learner, the new techniques discussed here do not have to be confined to the CCM framework. Mixture modeling can likely be effectively applied to any disjoint dataset, and Poisson regression for depth estimation can be used on its own or with other techniques. Future work can isolate these techniques from the CCM framework and explore how they can best be used.

Although not thoroughly tested here, inclusion of nearby pixels to the level-2 depth estimator also has potential to capture useful information. Future work could involve just using a two-level predictor, and compare those results to those obtained by Markov Random Fields.

In conclusion, this work is interesting, and some of the results are promising, though further testing will be useful in determining the exact value of each of the methods discussed here.

References

[1] G. Heitz, S. Gould, A. Saxena, D. Koller. Cascaded Classification Models: Combining Models for Holistic Scene Understanding. In *NIPS*, 2008.

[2] C. Li, A. Kowdle, A. Saxena, T. Chen. Feedback-Enabled Cascaded Classification Models for Scene Understanding. In *NIPS*, 2010.

[3] L. Li-Jia, R. Socher, F. Li. Towards total scene understanding: Classification, annotation, and segmentation in an automatic framework. In *CVPR*, 2009.

[4] R. Neal, G. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. *Learning in graphical models*, 89:355-368, 1998.

[5] A. Oliva and A. Torralba. Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope. In *IJCV*, 42:145-175, 2001.

[6] A. Saxena, S. H. Chung, and A. Y. Ng. 3-D Depth Reconstruction From a Single Still Image. In *IJCV*, 76, 2007.