菜品预测问题中的数据增广方法探究

郑耀威

2021 年 12 月 31 日

摘要

菜品预测问题旨在通过菜品的原料预测该菜品属于哪一种菜系，其输入是英文词组序列，输出是菜系类别标签。基于词袋模型的支持向量机（SVM）方法泛化性能较差，难以处理词表外的单词，因此深度神经网络更适用于此类复杂任务。训练深度神经网络需要大量标注数据，但该问题的标注数据较少。因此在本文中，我们基于 TextCNN 架构使用了多种数据增广方法和正则化方法解决标注数据缺乏的问题。具体地说，我们采用了随机重排、随机删除策略和表征混合方法对模型施加先验约束。最后，我们使用集成学习方法综合多个模型的预测结果，在测试集上达到了 82.01% 的准确率。模型实现代码公布在：https://github.com/hiyouga/Cuisine_Prediction。

1 问题介绍

菜品预测问题是一个多分类任务，给定制作菜品的原料，该任务旨在预测菜品的菜系类别，包括中国菜、印度菜、巴西菜等等。例如以“香草”、“牛奶”、“蛋黄”、“白糖”、“玉米淀粉”为原料制作的菜品属于法国菜，而以“鸡汤”、“鸡肉”、“照烧汁”、“红糖”、“蒜瓣”为原料制作的菜品属于中国菜，该任务最终从 20 种菜系类别中选择预测概率最大的类别作为结果。

菜品预测问题的输入是英文词组的序列，因此我们可以将其看作一般的文本分类任务。传统方法采用特征工程方法构造数据特征，并使用支持向量机（SVM）模型进行分类 [1]。随着深度神经网络方法的兴起，越来越多的工作采用结合词向量 [2] 的卷积神经网络（CNN）[3] 或循环神经网络（RNN）[4] 解决文本分类任务。实验证明，经过预训练的词向量可以提供丰富的语义和语法信息 [2]，从而进一步提高深度学习方法的表现。

本文采用卷积神经网络架构，以菜品原料为输入，使用给定的标注数据训练能够预测菜系类别的模型。我们首先将菜品原料视为连续的英文单词，将其按顺序拼接为一个句子。接着利用 GloVe 预训练词向量 [5] 将单词序列映射到向量空间，使用不同窗口大小的一维卷积操作提取输入样本的特征，最终通过 Softmax 层得到不同菜系类别的预测概率。

在菜品预测问题中，原料的先后顺序通常不影响最终的菜系类别。因此与传统的文本分类不同，一个好的菜品分类模型应当与输入的原料顺序无关。使用词袋表示的 SVM 等传统方法自然具备输入顺序无关的特性，而深度神经网络却不完全具备该特性。因此在模型训练阶段，我们运用了多种数据增广技术。具体地说，我们将输入的原料序列随机重排以构造新的输入。此外，为了提高模型的鲁棒性，我们类比 [6] 中的数据增广方法，随机删去输入序列中的一种原料。我们同时使用两种数据增广方式构造额外的训练样本，以提升模型的泛化性能。

除了数据增广技术，我们还利用了表征混合方法（MixUp）[7] 进一步提高模型表现。表征混合方法以一定的比例随机混合样本的输入特征和标签，并使用混合后的表征训练模型，该方法可以对模型的输入空间施加线性约束以提升其泛化性能。

最后，我们借鉴集成学习的思想 [8]，将多次训练得到的深度神经网络模型的输出进行集成，使用集成模型预测菜品的类别。综合上述方法，我们使用给定的数据集训练模型并调整超参数，最终在菜品预测问题的测试集上达到了 82.01% 的准确率。

该工作的主要贡献总结如下：

- 本文提出了两种不同的数据增广方式：随机重排和随机删除，以解决标注数据不足的问题。
- 本文运用了表征混合方法（MixUp）对模型进行正则化，以提高模型的泛化性能。
- 本文使用了深度集成学习方法，集成模型在测试集上达到了 82.01% 的准确率。

2 技术调研

2.1 文本分类

对于文本数据的分类问题，早期方法大多采用支持向量机（SVM）[1]、主题模型（LDA）[9] 等等。随着深度学习方法的兴起，多种神经网络架构也被应用于文本分类问题上。鉴于卷积神经网络（CNN）在视觉图像分类中的良好表现，TextCNN[3] 使用多通道的一维卷积操作提取文本特征，取得了较优的分类效果。相比卷积神经网络仅能提取局部特征的限制，循环神经网络（RNN）[4] 在提取全局特征上有着更大的优势，然而循环神经网络无法并行计算，因此其效率往往比较差。基于文档中信息含量不均匀的假设，分层注意力网络（HAN）[10] 使用注意力机制有效地捕捉文本中的重要信息。为了克服循环神经网络计算效率低的限制，Google 基于注意力机制提出了 Transformer 模型 [11]，利用自注意力机制并行地计算文档表示。同时随着近几年大规模预训练方法的推广，大型自然语言预训练 Transformer 模型（例如 BERT[12] 等）在经过微调后可在许多下游任务中取得惊人的表现。由于菜品预测问题所提供的数据集较小，本文仅采用 TextCNN 架构来构建分类模型。

2.2 数据增广和正则化

近年来，许多研究工作提出了多种数据增广方法以解决标注数据不足导致模型泛化性能较差的问题。对于图像数据，一种很自然的想法是将其翻转、裁剪在保持语义不变的情况下生成新的图像，这类方法被广泛应用于视觉模型的训练中 [13]。而由于文本数据的离散性，数据增广成为一个尚未完全解决的难题。简单数据增广（EDA）[6] 提供了一种对文本数据进行增广的思路，它采用简单的插入、替换、交换和删除策略，提升了少样本数据上模型的表现。类似地，我们在菜品预测问题中使用数据增广策略，以提升深度神经网络模型的表现。

除了数据增广以外，正则化方法也能通过对模型施加先验约束来提高模型表现。例如广为人知的权重衰减 [14]、随机丢弃（Dropout）[15] 和归一化技术 [16] 都属于正则化方法的范畴之内。标签混合技术 [17] 通过将独热编码的标签向量与标签空间的均匀分布向量进行混合来避免置信度过高的问题。表征混合方法（MixUp）[7] 以一定的比例随机混合输入特征和标签，使

用混合后的表征训练模型，对模型的输入空间施加额外的线性约束以提升泛化性能。浮动方法（Flooding）[18] 通过避免训练损失下降到零来避免模型过拟合。对抗训练技术 [19] 不仅能提升模型的鲁棒性，也能通过引入对抗样本对模型进行正则化。参数对抗扰动算法（AMP）[20] 在每次迭代时向模型参数中注入对抗噪声，以寻找光滑的局部最优解，提高模型的泛化性能。我们选取了权重衰减和表征混合方法作为实验中实际采取的正则化方法，提高了模型在测试集上的表现。

2.3 集成学习

在许多学习场景中，我们会面临标注数据匮乏的问题。在训练集较小的情况下，我们往往只能学习到表现比较差的模型，而将多个表现较差的模型进行集成，通常可以得到一个稳定且表现较好的模型，这也就是集成学习 [21] 的思想。实验证明，将多次训练得到的深度神经网络的输出进行平均后，可以得到一个较稳定的模型 [8]。在实验中，我们随机训练 10 个深度神经网络分类模型，取最后一层输出概率的平均值作为每个菜系类别的预测概率，集成后的模型进一步提升了测试集上的准确率。

3 方法描述

3.1 数据增广

3.1.1 随机重排

在菜品预测问题中，原料的先后顺序应当与最终的菜系类别无关。因此我们对于每个输入样本，将其原料进行随机重排得到新的增广样本，如表 1 所示。例如 “香草”、“牛奶”、“蛋黄”、“白糖”、“玉米淀粉” 进行随机重排后，可能为 “白糖”、“牛奶”、“蛋黄”、“玉米淀粉”、“香草”。

原始数据	p_1	p_2	p_3	p_4	p_5
增广数据	p_4	p_2	p_3	p_5	p_1

表 1: 随机重排示例

3.1.2 随机删除

制作菜品时通常会用到多种原料，缺少一种原料往往不会影响最终的菜系类别。因此在模型训练阶段，我们随机删去每个训练样本中的一种原料，如表 2 所示。例如 “香草”、“牛奶”、“蛋黄”、“白糖”、“玉米淀粉” 进行随机删除后，可能为 “香草”、“牛奶”、“蛋黄”、“玉米淀粉”。

原始数据	p_1	p_2	p_3	p_4	p_5
增广数据	p_1	p_2	p_3	p_5	

表 2: 随机删除示例

3.2 模型架构

我们选用 TextCNN 架构 [3] 构造用于菜品预测的神经网络模型，将原料词组按顺序拼接为一个单词序列作为模型输入。假设输入序列 x 包含 n 个单词，首先将序列中每个单词 w_i 映射为 k 维的词向量 $e_i \in \mathbb{R}^k$ 作为卷积层的输入。卷积层共包含 h 个卷积核 $f \in \mathbb{R}^{mk}$，每个卷积核以 m 个连续单词作为输入，通过以下函数计算输出 c_i：

$$c_i = \text{ReLU}(f \cdot e_{i:i+m-1} + b_f) \tag{1}$$

其中 $b_f \in \mathbb{R}$ 是偏置项，我们选用修正线性单元 ReLU 作为非线性激活函数。

每个卷积核在整段输入序列上进行卷积，遍历所有可能的窗口 $\{e_{1:m}, e_{2:m+1}, \ldots, e_{n-m+1:n}\}$ 产生特征图：

$$c = [c_1, c_2, \ldots, c_{n-m+1}] \tag{2}$$

其中 $c \in \mathbb{R}^{n-m+1}$。

我们使用最大池化操作根据特征图计算出当前卷积核的输出：

$$\hat{c} = \max_i c_i \tag{3}$$

假设整个卷积层共有 h 个卷积核，我们将所有卷积核的输出拼接作为池化层的输出 $q \in \mathbb{R}^h$：

$$q = [\hat{c}_1, \hat{c}_2, \ldots, \hat{c}_h] \tag{4}$$

最后我们使用 Softmax 层计算每个菜系类别的预测概率 $o \in \mathbb{R}^C$：

$$o = \text{Softmax}(W_o q + b_o) \tag{5}$$

其中 $C = 20$，W_o 和 b_o 分别为 Softmax 层的权重和偏置。

我们将 TextCNN 模型的整体架构绘制在图 1 中。

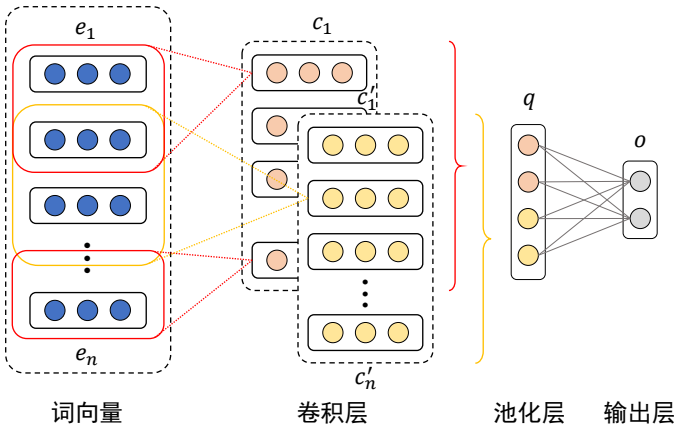

图 1: TextCNN 模型架构图

3.3 损失函数

假设训练集中包含 T 个训练样本 (x_t, y_t)。对于多分类问题，通常我们使用交叉熵损失函数来优化模型参数，同时附加权重衰减正则项：

$$\mathcal{L}(\hat{y}, y) = -\sum_{i=1}^{T}\sum_{j=1}^{C} y_i^j \log(\hat{y}_i^j) + \lambda \sum_{\theta \in \Theta} \|\theta\|_2^2 \tag{6}$$

其中 y_i^j 代表真实类别，$\hat{y}_i^j$ 代表预测类别，$C = 20$，Θ 对应全部的可训练参数，λ 控制权重衰减正则项对训练过程的影响程度。

4 模型评估和分析

4.1 数据集

我们在给定的训练集和测试集上进行实验，其中训练集包含菜品 ID、原料和烹饪风格（即菜系类别），测试集仅包含菜品 ID 和原料。菜品 ID 是唯一的数字序列，而原料由多个词组组成，每个词组中包含一个或多个小写英文单词。烹饪风格是模型需要预测的标签，由一个小写英文单词表示。我们将数据集的统计信息整理在表 3 中，表 4 列出了数据集中的所有 20 种菜系类别。

	样本数	最大原料个数	最大单词个数	类别数
训练集	31819	65	136	20
测试集	7955	38	73	-

表 3: 数据集统计信息

brazilian	巴西菜	french	法国菜	jamaican	牙买加菜	russian	俄国菜
british	英国菜	greek	希腊菜	japanese	日本菜	southern_us	美国南部菜
cajun_creole	克里奥尔菜	indian	印度菜	korean	韩国菜	spanish	西班牙菜
chinese	中国菜	irish	爱尔兰菜	mexican	墨西哥菜	thai	泰国菜
filipino	菲律宾菜	italian	意大利菜	moroccan	摩洛哥菜	vietnamese	越南菜

表 4: 全部菜系类别

4.2 实现细节

我们将菜品中所有原料对应的英文词组以空格分词，再将所有分词后的词语序列按顺序拼接为一个句子（严格地说，这里的句子不存在语法结构，仅由连续的单词构成），将其作为 TextCNN 模型的输入。在实验中，我们采用 300 维的 GloVe 预训练词向量 [5] 作为词向量嵌入表征。我们还随机初始化了 10 维的位置编码，将其拼接在词向量后面构成单词的表征。对于没有出现在预训练词向量词表中的单词，我们使用均匀分布 $U(-0.25, 0.25)$ 对词向量进行随机

初始化。在训练中，我们只更新位置编码，冻结 GloVe 预训练词向量。TextCNN 的卷积窗口采用 $\{3,4,5\}$ 三种大小，每种大小的卷积核各 256 个。我们使用 Adam 优化器 [22] 训练模型，其中学习率设置为 10^{-4}，权重衰减系数 $\lambda = 10^{-4}$。在训练中，我们使用余弦退火算法 [23] 逐步减小学习率。我们选取训练集中的 5% 样本作为验证集，在剩余的 95% 样本上训练 200 轮，批处理大小为 32，取验证集上准确率最高的一轮作为最优模型。由于离散的词语序列输入无法直接进行表征混合，因此我们选取池化后的句子表征和菜系类别标签进行混合，表征混合方法的超参数依照原文 [7] 设为 $\alpha = 1$。

4.3 超参数调优

我们使用 K 折交叉验证方法进行超参数的调优。具体地说，我们首先将训练集随机切分为 20 个互不相交的大小相同的子集，然后利用其中 19 个子集的样本训练模型，利用剩余的 1 个子集测试模型，并将该过程类似地重复 20 次，最后取平均测试结果最好的模型超参数作为最优超参数。

4.4 结果对比

经过模型训练和超参数调优，我们将 TextCNN 模型和其他方法进行对比，如表 5 中所示。评测指标采用准确率，即预测正确的样本占全部样本的比例。我们可以看到使用了数据增广方法的 TextCNN 模型取得了优于 SVM 方法和 BERT 模型的表现，并且深度集成后的 TextCNN 模型进一步提高了测试集上的准确率，最终在比赛工作站取得了第 2 名的成绩，如图 2 所示。

模型	准确率
SVM	81.00±0.05
BERT	81.18±0.12
TextCNN	81.44±0.08
TextCNN(10×)	**82.01±0.06**

表 5: 模型表现对比

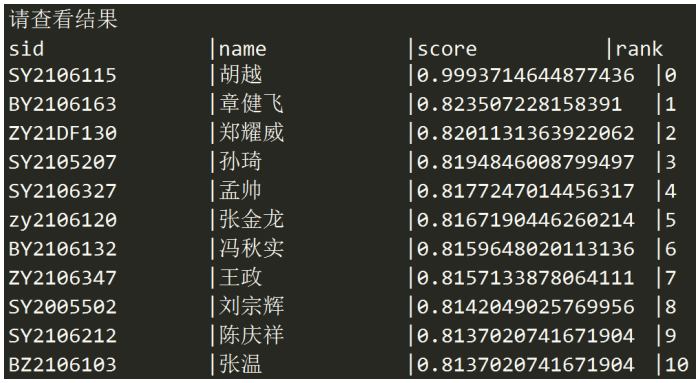

图 2: 比赛工作站排名截图

4.5 消融实验

为了验证模型设计中使用的多种方法与最终结果的关系，我们在数据集上进行了消融实验，实验结果呈现在表 6 中。根据实验结果，我们可以看到使用预训练词向量可以提供丰富的语义信息，提升了 4% 左右的测试准确率。而数据增广方法缓解了标注数据匮乏的问题，带来了 2% 左右的准确率提升。表征混合方法对模型施加了正则约束，在测试集上提升了 1% 左右的准确率。深度集成方法进一步带来 0.5% 左右的准确率提升。

模型	准确率
TextCNN(10×)	82.01±0.06
TextCNN	81.44±0.08
—表征混合方法	80.45±0.21
—数据增广	78.21±0.22
—预训练词向量	74.62±0.13

表 6: 消融实验结果

4.6 可视化分析

为了进一步分析模型的工作原理，我们将 TextCNN 池化层的输出特征，也是 Softmax 层的输入特征进行可视化分析。我们在数据集中随机选取了 200 个样本，使用 t-SNE[24] 降维方法，在图 3 中绘制出了 2 维平面上特征的分布。我们可以看到神经网络模型的高层特征具有良好的聚类特征，验证了模型具备较为可靠的分类效果。

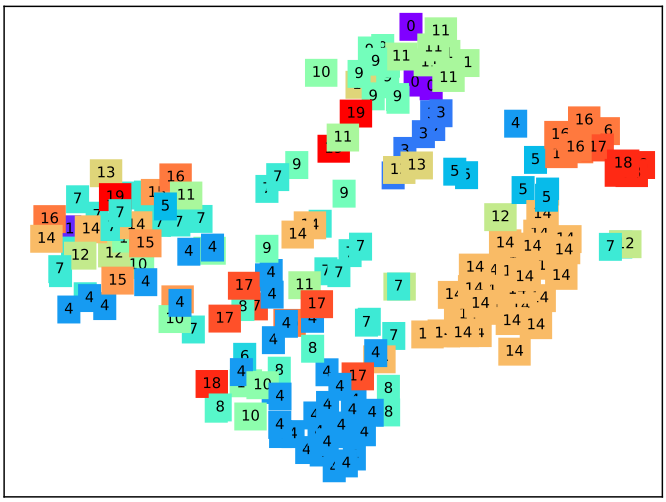

图 3: t-SNE 可视化分析结果

5 结论

在这篇文章中，我们研究了使用深度神经网络模型预测菜系类别的问题。我们首先对菜品预测数据集进行了分析，选用了 TextCNN 架构解决文本数据的多分类问题。为了缓解标注数据不足的问题，我们运用了多种数据增广方法构造额外的训练样本，随机重排和随机删除策略提升了模型在测试集上的表现。我们利用了表征混合方法（MixUp）对模型的输入空间施加线性约束，提高了模型的泛化性能。我们还使用了深度集成方法，综合多次训练的结果，得到了稳定且效果更好的分类器。在实验中，我们使用 K 折交叉验证方法对超参数进行了调优，集成的 TextCNN 在测试集上达到了 82.01% 的准确率，超越了 SVM 方法和 BERT 模型的表现，在比赛工作站上取得了第 2 名的成绩。最后，我们通过消融实验和可视化分析进一步验证了模型设计中各个组成部件的有效性和方法的可靠性。

参考文献

[1] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[2] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[3] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, 2014.

[4] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*, 2014.

[5] Jeffrey Pennington, Richard Socher, and Christopher D Manning. GloVe: global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[6] Jason Wei and Kai Zou. EDA: easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, 2019.

[7] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

[8] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[9] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[10] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489, 2016.

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[14] Anders Krogh and John A Hertz. A simple weight decay can improve generalization. In *Advances in Neural Information Processing Systems*, pages 950–957, 1992.

[15] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[16] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[17] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 2818–2826, 2016.

[18] Takashi Ishida, Ikko Yamane, Tomoya Sakai, Gang Niu, and Masashi Sugiyama. Do we need zero training loss after achieving zero training error? In *Proceedings of the 37th International Conference on Machine Learning, ICML*, 2020.

[19] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations, ICLR*, 2015.

[20] Yaowei Zheng, Richong Zhang, and Yongyi Mao. Regularizing neural networks via adversarial model perturbation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8156–8165, 2021.

[21] David Opitz and Richard Maclin. Popular ensemble methods: An empirical study. *Journal of artificial intelligence research*, 11:169–198, 1999.

[22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[23] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017.

[24] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11), 2008.