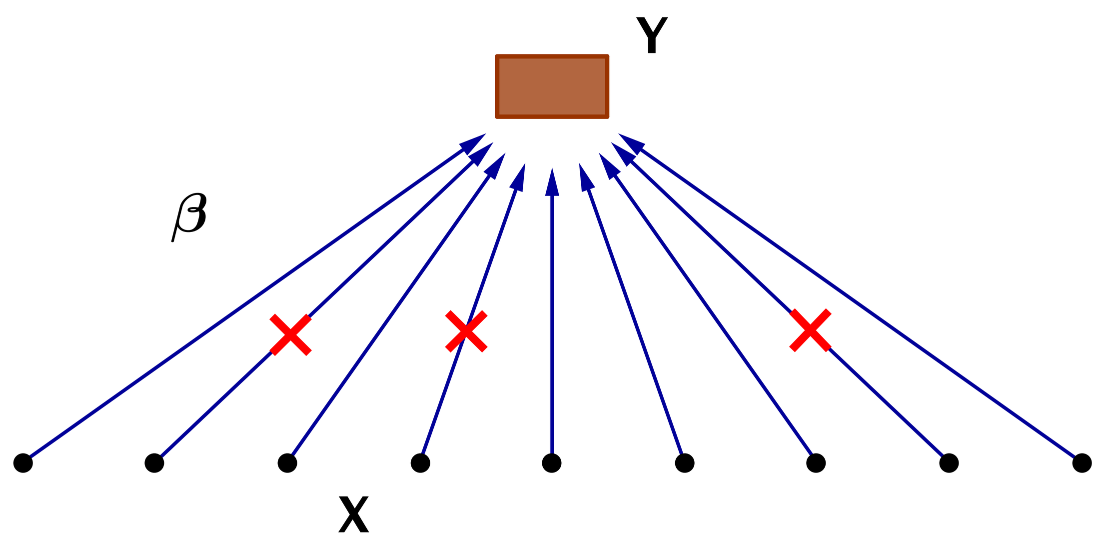
DROPOUT TRAINING AS ADAPTIVE REGULARIZATION

STEFAN WAGER, SIDA WANG, AND PERCY LIANG   STANFORD UNIVERSITY

DROPOUT TRAINING



For a probabilistic model of the form

$$\widehat{\mathbb{P}[y \mid x]} = f\left(\hat\beta \cdot x\right),$$

dropping out a feature is equivalent to setting it to 0. Writing ℓ for the loss (i.e., negative log-likelihood),

$$\hat\beta_{DROP} = \operatorname{argmin}_\beta \left\{ \sum_{i=1}^n \mathbb{E}\left[\ell\left(\beta; \tilde x^{(i)}, y^{(i)}\right)\right]\right\},$$

$$\text{where } \tilde x_j^{(i)} = \begin{cases} 0 & \text{with prob. } \delta \\ x_j^{(i)}/(1-\delta) & \text{with prob. } 1-\delta \end{cases}$$

DROPOUT FOR GENERALIZED LINEAR MODELS

Dropout acts as a *label-independent regularizer*

$$\boxed{\hat\beta_{DROP} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^n \left(\ell\left(\beta; x^{(i)}, y^{(i)}\right) + R(\beta; x_i)\right)\right\}}.$$

In a generalized linear model (GLM),

$$\ell(\beta; x, y) = -y\,\beta \cdot x + A(\beta \cdot x).$$

We can write $\tilde x$ as $\xi \odot x$, where $\xi = 0$ or $1/(1-\delta)$ and $\odot$ is a component-wise product. The dropout loss becomes

$$\begin{aligned} &\mathbb{E}_\xi\left[\ell(\beta; \xi \odot x, y)\right] \\ &\quad = -\mathbb{E}_\xi\left[y\,\beta \cdot (\xi \odot x)\right] + \mathbb{E}_\xi\left[A(\beta \cdot (\xi \odot x))\right] \\ &\quad = -y\,\beta \cdot x + \mathbb{E}_\xi\left[A(\beta \cdot (\xi \odot x))\right] \\ &\quad = \ell(\beta; x, y) + R(\beta; x), \end{aligned}$$

where $R(\cdot)$ is the **dropout regularizer**

$$R(\beta; x) = \mathbb{E}_\xi\left[A(\beta \cdot (\xi \odot x))\right] - A(\beta \cdot x).$$

R is always non-negative because A is convex.

A **second-order expansion** of A gives us

$$R(\beta; x) \approx \frac{1}{2}\frac{\delta}{1-\delta} A''(\beta \cdot x) \sum_{j=1}^p \beta_j^2 x_j^2.$$

This leads to a **quadratic dropout penalty**

$$R^q(\beta; X) = \frac{1}{2}\frac{\delta}{1-\delta} \beta^\intercal \operatorname{diag}\left(X^\intercal V X\right)\beta,$$

where V is diagonal with $V_{ii} = A''(\beta \cdot x)$.

$$\textit{lin. reg.: } R^q(\beta; X) = \frac{1}{2}\frac{\delta}{1-\delta} \sum_j \beta_j^2 \sum_i x_{ij}^2$$

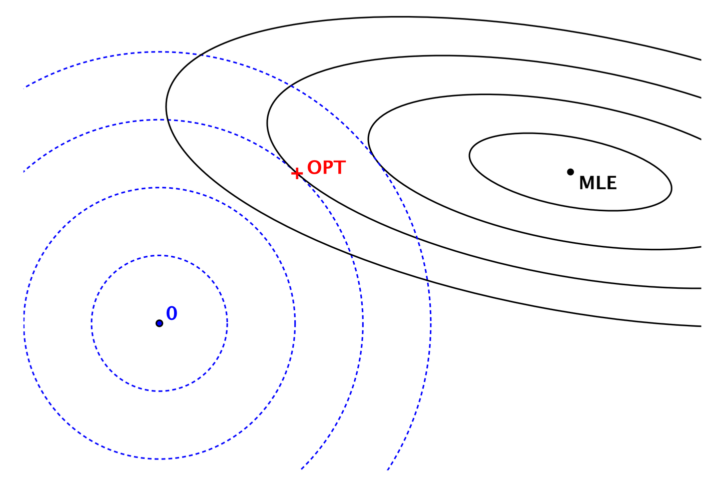
$$\textit{log. reg.: } R^q(\beta; X) = \frac{1}{2}\frac{\delta}{1-\delta} \sum_{i,j} \beta_j^2\, x_{ij}^2\, \hat p_i(1-\hat p_i)$$

Here, $\hat p_i = \sigma(\hat\beta \cdot x_i)$ is the i^{th} prediction.

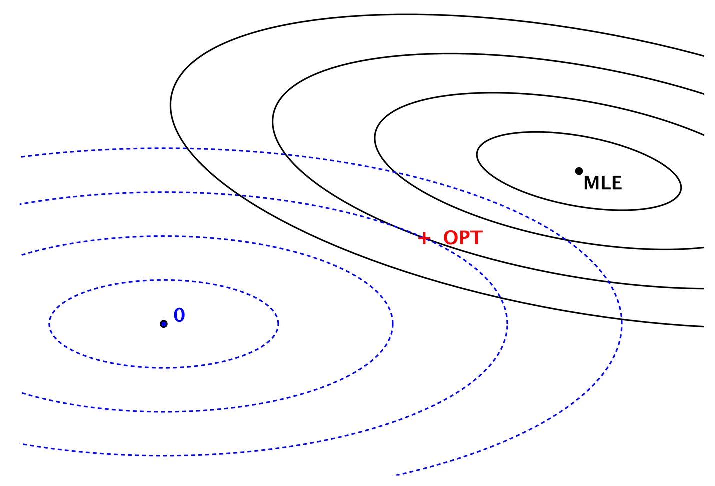
Intuition: For logistic regression, dropout privileges *rare features* and *confident predictions*.

THE DROPOUT REGULARIZER

Level surfaces of the regularizer are shown in blue; likelihood surfaces are black. Dropout acts as an L_2 penalty applied after scaling X by the root inverse *diagonal Fisher information*.



L2 regularization



Dropout regularization

DROPOUT AND ADAGRAD

Stochastic gradient descent uses the update rule

$$\hat\beta_{t+1} = \hat\beta_t - \eta_t\, g_t, \text{ where } g_t = \nabla \ell_{x_t, y_t}(\hat\beta_t).$$

This is equivalent to solving a linearized L_2-penalized problem:

$$\hat\beta_{t+1} = \underset{\beta}{\operatorname{argmin}} \left\{ \ell_{x_t, y_t}(\hat\beta_t) + g_t \cdot (\beta - \hat\beta_t) + \frac{1}{2\eta_t}\|\beta - \hat\beta_t\|_2^2 \right\}.$$

We could use a dropout-like penalty instead

$$\hat\beta_{t+1} = \underset{\beta}{\operatorname{argmin}} \left\{ \ell_{x_t, y_t}(\hat\beta_t) + g_t \cdot (\beta - \hat\beta_t) + \frac{1}{2}\left(\beta - \hat\beta_t\right)^\intercal \operatorname{diag}\left(\sum_{i=1}^t \nabla^2 \ell_{x_i, y_i}\left(\hat\beta_i\right)\right)\left(\beta - \hat\beta_t\right)\right\}.$$

The result is closely related to **diagonal AdaGrad** (Duchi et al., 2010).

SEMI-SUPERVISED DROPOUT

If we have m unlabeled datapoints $\{x_j^*\}$, we can use them to learn a better adaptive regularizer

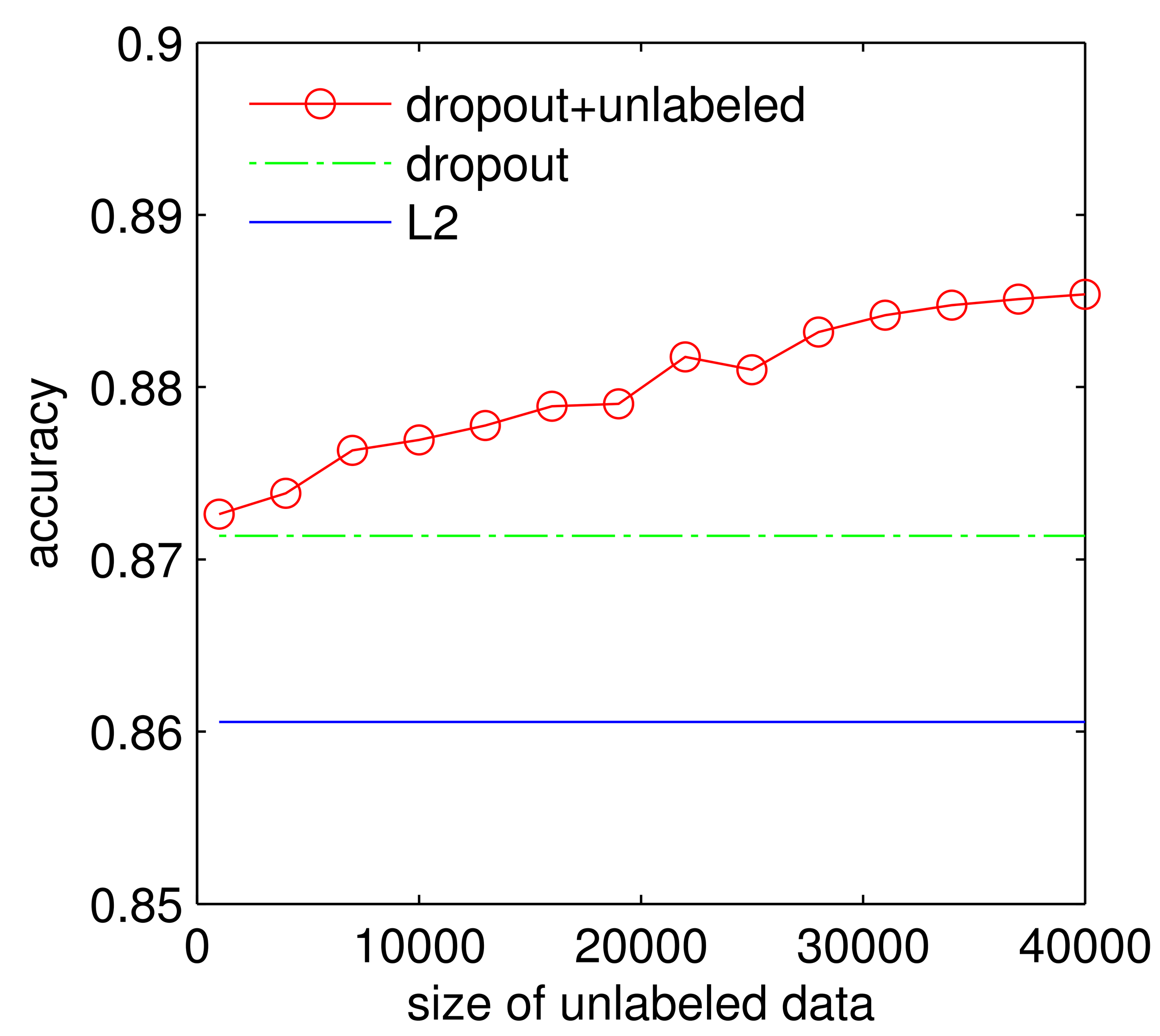
$$R^*(\beta) = \frac{n}{n + \alpha m}\left(\sum_{i=1}^n R(\beta; x_i) + \alpha \sum_{j=1}^m R\left(\beta; x_j^*\right)\right)$$

For the examples below, we split the full dataset into 3 folds of equal size: training, test, and unlabeled. K is the number of classes

Dataset	K	L_2	Drop	+Unlabeled
CoNLL	5	91.46	91.81	**92.02**
20news	20	76.55	79.07	**80.47**
RCV1$_4$	4	94.76	94.79	**95.16**
R21578	65	90.67	**91.24**	90.30
TDT2	30	97.34	97.54	**97.89**

This table is from our follow up paper with Mengqiu Wang and Chris Manning (EMNLP, 2013), which also extends our results to linear-chain conditional random fields.

SEMI-SUPERVISED RESULTS: SENTIMENT CLASSIFICATION



Unigram features, with 10k labeled examples.

IMDB sentiment classification dataset (Maas et al, 2011). Highly polar reviews for both training and test (25k each). 50k unlabeled reviews (not all polarized). We used logistic regression with dropout on unigram/bigram features. Semi-supervised dropout improves on state-of-the-art results.

Methods	Labeled	+Unlabeled
MNB-Uni	83.62	84.13
MNB-Bi	86.63	86.98
Vect.Sent	88.33	88.89
LogReg-Bi	90.13	–
NBSVM-Bi	91.22	–
Drop-Uni	87.78	89.52
Drop-Bi	91.31	**91.98**

References. Multinomial naive Bayes (MNB): Su et al., 2011; Word vectors (Vect.Sent): Maas et al, 2011; Naive Bayes SVM (NBSVM): Wang & Manning, 2012.