intrinsic plausibility

By the end of this section, you'll be able to
- explain how regularization, in its incarnation as margin-maximization, counters data terms to improve generalization
- write a regularized ML program (namely, an SVM), to classify high-dimensional data

HOW GOOD IS A HYPOTHESIS? PLAUSIBILITY — Now to define intrinsic plausiblity, also known as a **regularizer**. We find a hypothesis more plausible when its "total amount of dependence" on the features is small. So we'll focus for now on capturing this intution: *a hypothesis that depends a lot on many features is less plausible.*° We may conveniently quantify this as proportional to a sum of squared weights (jargon: **L2**):° implausibility of $h = (a, b, \cdots) = \lambda(a^2 + b^2 + \cdots)$. In code:

```
LAMBDA = 1.
def implausibility(a,b):
  return LAMBDA * np.sum(np.square([a,b]))
```

Intuitively, the constant λ=`LAMBDA` tells us how much we care about plausibility relative to goodness-of-fit-to-data.

← There are many other aspects we might design a regularizer to capture, e.g. a domain's symmetry. The regularizer is in practice a key point where we inject domain knowledge.

← Food For Thought: When (a, b) represent weights for brightness-width digits features, how do hypotheses with small $a^2 + b^2$ visually differ from ones with small $6.86a^2 + b^2$ (a perfectly fine variant of our 'implausibility')?

Here's what the formula means. Each of three friends has a theory° about which birds sing. Which theory do we prefer? Well, **AJ** seems too confident. Wingspan may matter but probably not so decisively. **Pat** avoids black-and-white claims, but Pat's predictions depend substantively on many features: flipping any one quality flips their prediction. This seems implausible. By contrast, **Sandy**'s hypothesis doesn't depend too strongly on too many features. To me, a bird non-expert, Sandy's seems most plausible.

← **AJ** insists a bird with a wings shorter than 1ft can't fly far, so it's *sure* to sing; Conversely, birds with longer wings never sing. **Pat** checks if the bird grows red feathers, eats shrimp, lives near ice, wakes in the night, and has a bill. If and only if an even number of these 5 qualities are true, the bird probably sings. **Sandy** says shorter wings and nocturnality both make a bird somewhat more likely to sing.

Now we can define the overall undesirability of a hypothesis:°

← We'll use SVM loss but feel free to plug in other losses to get different learning behaviors!

```
def objective_function(examples,a,b):
  data_term = np.sum([svm_loss(x,y,a,b) for x,y in examples])
  regularizer = implausibility(a, b)
  return data_term + regularizer
```

MARGINS — To build intuition let's suppose λ is a tiny positive number. Then minimizing the objective function is the same as minimizing the data term, the total SVM loss: our notion of implausibility breaks ties.

Now, how does it break ties? Momentarily ignore the Figure's rightmost orange point and consider the black hypothesis; its predictions depend only on an input's first (vertical) coordinate, so it comes from weights of the form $(a, b) = (a, 0)$. The $(a, 0)$ pairs differ in SVM loss. If $a \approx 0$, each point has leeway close to 0 and thus SVM loss close to 1; conversely, if a is huge, each point has leeway very positive and thus SVM loss equal to the imposed floor: 0. So SVM loss is 0 as long as a is so big that each leeway to exceed 1.

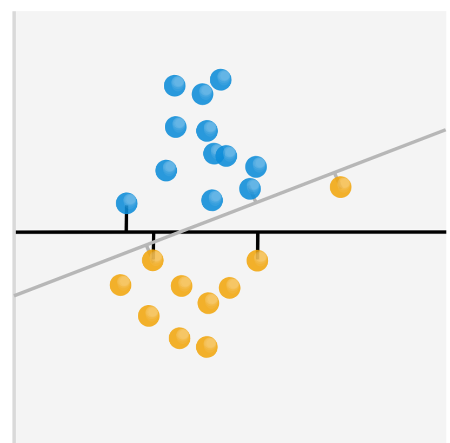

Figure 10: **Balancing goodness-of-fit against intrinsic plausibility leads to hypotheses with large margins.** A hypothesis's **margin** is its distance to the closest correctly classified training point(s). Short stems depict these distances for two hypotheses (**black**, **gray**). If not for the rightmost orange point, we'd prefer **black** over **gray** since it has larger margins. With large λ (i.e., strong regularization), we might prefer black over gray even with that rightmost orange point included, since expanding the margin is worth the single misclassification.

Imagine sliding a point through the plane. Its leeway is 0 at the black line and changes by a for every unit we slide vertically. So the farther the point is from the black line, the less a must be before leeway exceeds 1 — and the happier is the regularizer, which wants a small. So *minimizing SVM loss with an L2 regularizer favors decision boundaries far from even the closest correctly classified points!* The black line's margins exceed the gray's, so we favor black.

For large λ, then this margin-maximization tendency can be so strong that it overrides the data term. Thus, even when we bring back the rightmost orange point we ignored, we might prefer the black hypothesis to the gray one.

Now for some really good intuition-building brain-food!

Food For Thought: Identify which point on the gray curve to the right corresponds to $\lambda = 0$. How about $\lambda = \infty$?

Food For Thought: We have two weight coefficients (corresponding to the horizontal and vertical axes of the Figure). Based on the fit-to-data term, which coefficient is the loss more sensitive to?

Food For Thought: Observe that the weight-vs-λ trajectory is curved: it doesn't interpolate linearly between its $\lambda = 0$ and $\lambda = \infty$ values. Which weight (horizontal or vertical) gets suppressed 'first' as we increase λ from 0?

Food For Thought: By thinking about points at which blue and orange contours are mutually tangent, sketch the weight-vs-λ trajectory described in the Figure. That is: check that the Figure is right!

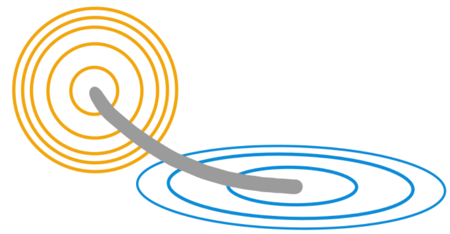

Figure 11: **Regularization suppresses different features *by different amounts*.** We show a contour plot of loss terms over 2D weight space: an L2 regularizer and a fit-to-data term. As we vary λ from 0 (L2 doesn't matter) toward ∞ (data doesn't matter), the optimal weight changes. We show this weight-vs-λ trajectory in gray. **Warning**: For the perceptron and hinge notions of fit-to-data, the latter term won't look so smooth. Still, the moral about regularization applies. (And future models we'll discuss (logistic models, least-squares regression, etc) *are* smooth.)

OPTIMIZATION — Now that we've defined our objective function, we want to find a hypothesis $h = (a, b)$ that minimizes it. We've already discussed how to nudge the weight vector to reduce the badness-of-fit for a datapoint. How do we nudge it to reduce the implausibility? Well, we reduce the λ term simply by moving a, b closer to 0! That is, we combine an update of the form

$$w^{\text{new}} = w^{\text{old}} - \lambda w^{\text{old}}$$

with the data update.

(To get this to match our objective exactly, we should actually write $2\lambda/N$ instead of λ. The 2 comes from the second power in L2's definition; the $1/N$, more importantly, comes from the fact that we have N data terms but just 1 plausiblity term. So if we work row-by-row (datapoint-by-datapoint), we ought to divvy up the plausibility term into N many terms, each of strength λ/N. At this point, we can just abstract this reasoning away by defining a new constant — say L — that secretly is $2\lambda/N$. Later, it'll be good to know where L comes from.)

We end up with

```
ETA = 0.01
ab = initialize()
for t in range(10000):
  xfeatures, y = fetch_datapoint_from(training_examples)
  ab = ab + ETA * ( - L * ab
                  + y * xfeats * (0 if max(0., y*ab.dot(xfeatures))==0 else 1) )
```

Figure 12: With $\lambda = 0.02$ the objective visibly prefers weights near 0. We develop an algorithm to take steps in this plane toward the minimum, 'rolling down' the hill so to speak.

This is the **pegasos algorithm** we'll see in the project. Soon we'll formalize and generalize this algorithm using calculus.

Food For Thought: We've discussed the L2 regularizer. Also common is the L1 regularizer: implausibility of $h = (a, b, \cdots) = \lambda(|a| + |b| + \cdots)$. Hypotheses optimized with strong L1 regularization will tend to have zero dependence on many features. Explain to yourself and then to a friend what the previous sentence means, why it is true, and how we might exploit it in practice.