Computation, Information, and Intelligence (COMS/ENGRI/INFO/COGST 172), Fall 2005
9/21/05: **Lecture 12 aid – The perceptron convergence theorem, continued**

Agenda: Practice with perceptrons and the PLA; finish up our "one-sided" perceptron convergence theorem.

Follow-ups to last time People asked a lot of very good questions both during and after last lecture. It seemed like it would be useful to collect these up and write some of them down.

Q: How should I think about the fact that the oracle doesn't have to have a $\overrightarrow{w}^{\infty}$ and $\overrightarrow{T}^{\infty}$ "in mind" while it's supplying examples, and yet our *proof* seems to know about that vector and threshold?
A: The "∞" notation is meant to indicate that the existence of a $\overrightarrow{w}^{\infty}$ and $\overrightarrow{T}^{\infty}$ can only be verified "after the dust settles" — that is, after all the (infinite number of) examples and labels have been presented. The oracle may choose not to obey the restrictions, after all (perhaps the universe really is out to personally mess you up), in which case we have no guarantee that our perceptron learning algorithm (or any algorithm) will achieve identification in the limit. What we *can* show is that *if* the oracle's labeled examples were to turn out to satisfy the restrictions we laid out, *then* our PLA would have turned out in the end to have, after a finite number j of examples, produced a weight vector $\overrightarrow{w}^{(j)}$ that correctly assigns labels to all the (infinite number of) examples that were subsequently presented. That is, you can think of the proof as a *retrospective* analysis, studying how the learner performed in the past. (A slight problem with this perspective is that the oracle-learner interaction is considered to go on forever, which raises the question of when exactly does this analysis occur, but maybe it's better not to dwell on such eschatological issues.)

Perhaps a useful way of conceptualizing this is to assume that the oracle is actually using a tremendously complicated algorithm — way more complex than a perceptron function — to decide on the labels of examples; for example, in disease diagnosis, whether a patient has a given disease may depend on unknown factors, and maybe the relationship between the factors that *are* known cannot really be described as linear. Thus, the oracle *never* has a weight vector and threshold in mind. Yet, we can apply the PLA to attempt to learn a perceptron function to explain the data anyway; and the important point is, again, *if* the oracle's labeled examples were to turn out to satisfy the restrictions we laid out (including being consistent with some perceptron function parameterized by a $\overrightarrow{w}^{\infty}$ and $\overrightarrow{T}^{\infty}$ conforming to the given restrictions), *then* our PLA would successfully achieve identification in the limit.[1]

[1] Alternatively, here's a way of thinking about the situation if you prefer to consider the oracle as being allowed to "change its mind". It's also an interesting scenario because it considers the *learner* to be (semi-)evil.

Suppose the oracle is the administrator of a housing lottery who uses some system to decide on the winners based on codenames that students submit. Because the lottery must at least appear fair, there are conditions that the winning codenames must obey, such as not all ending with "Lee". The learner is a student association that tries to discern a pattern in the winning codenames, for obvious reasons. If the oracle notices that students from this association seem to be picking up on the pattern, then the oracle would want to change its system, but without "signaling" to the association (or others) that the housing-lottery authority knows what the association has done, and without shaking the confidence of students in the fairness of the system by making it look like a sudden shift in policy has occurred. Hence, the housing office would want to switch to a system that still could have generated the old winning codes, but doesn't fit the pattern the student association has picked up on.

A related scenario involves the oracle being played by Google, the learner being played by a firm that tries to artificially improve the ranking of a particular webpage, and Google needing to change its ranking function so as to not advantage the particular firm while at the same time not affecting the legitimate rankings of other pages or causing people to think that Google's rankings are not stable.

Q: If perceptron functions are defined by both a weight vector and a threshold value, then why isn't a threshold value specified in the PLA?
A: If our restrictions hold, then the examples will all be consistent with some perceptron that has a threshold of 0. So the PLA we defined also (implicitly) adopts a threshold of 0 always, too. That's why we can think of the check "If $\overrightarrow{w}^{(i-1)} \cdot \overrightarrow{x}^{(i)} \leq 0$" (line 3 of the PLA) as corresponding to, "If the previous hypothesis is wrong, or just barely right, on $\overrightarrow{x}^{(i)}$": the hypothesis perceptron computes the inner product $\overrightarrow{w}^{(i-1)} \cdot \overrightarrow{x}^{(i)}$ and checks it against the threshold value 0.

Q: How does the length of the weight vector that the PLA proposes change when an update occurs?
A: In the example we gave in class, we had $\overrightarrow{w}^{(0)} = (0,0)$, $\overrightarrow{w}^{(1)} = (0,1)$, and $\overrightarrow{w}^{(2)} = (-\frac{4}{5}, \frac{2}{5})$. So, the weight vector increased in length after the first instance, but shrank after the second one. Thus, there's not necessarily a set pattern; it depends on the specific instances presented by the oracle.

Q: Why don't we just force the PLA to produce weight vectors of unit length, since the oracle restrictions specify that $\overrightarrow{w}^{\infty}$ is of unit length?
A: The length of the weight vector has an interesting property in our setting. Recall that all the instances are of length 1. If the weight vector is very long, then adding to it a length-one vector doesn't change the direction of the weight vector very much. So, the length of the PLA's current weight vector can be thought of as corresponding to greater or lesser "inertia" in the system.

Also, a close examination of our proof of the (one-sided) perceptron convergence theorem reveals that "length normalization" (forcing the hypothesis weight vectors to all have unit length) creates a problem with our argument.

Q: Is it enough to show that the cosine between $\overrightarrow{w}^{\infty}$ and the weight vectors that the PLA proposes is increasing at each update?
A: No: if the cosine increases, but by a smaller and smaller amount each time, then it would in principle be possible for the updating process to continue forever.

Q: How does the gap condition make identification in the limit possible?
A: Intuitively, it means that it's sufficient for the learner to get its hyperplane "into the gap" — once there, the oracle can't place instances "close enough to" the learner's hyperplane to force a mistake.

I. Reminder: the restrictions we impose

1. The *one-zero consistency* condition: All the labeled examples turn out to be consistent with some perceptron function $f_{\overrightarrow{w}^{\infty}, T^{\infty}}$ where length $(\overrightarrow{w}^{\infty}) = 1, T^{\infty} = 0$.

2. The *length restriction*: For all i, length $\left(\overrightarrow{x}^{(i)}\right) = 1$.

3. The *gap condition*: There is a $g > 0$ such that for all $\overrightarrow{x}^{(i)}$ and the $\overrightarrow{w}^{\infty}$ specified above, we have that $\overrightarrow{w}^{\infty} \cdot \overrightarrow{x}^{(i)} \geq g$.

II. Reminder: The perceptron learning algorithm This is a "one-sided" version of the algorithm Rosenblatt proposed.

1) Set $\overrightarrow{w}^{(0)}$ to all zeroes.
2) For each example $\overrightarrow{x}^{(i)}$ (i increasing from 1 on),
3) If $\overrightarrow{w}^{(i-1)} \cdot \overrightarrow{x}^{(i)} \leq 0$,

4) set $\overrightarrow{w}^{(i)}$ to $\overrightarrow{w}^{(i-1)} + \overrightarrow{x}^{(i)}$ ("update");
5) otherwise, set $\overrightarrow{w}^{(i)}$ to $\overrightarrow{w}^{(i-1)}$ ("no change").

III. In-class exercises

Q1: Suppose we are dealing with two-component data such that the horizontal axis corresponds to income relative to the average, and the vertical axis corresponds to age with respect to the average. The binary function to be learned takes the value +1 for people eligible for a certain tax credit, -1 otherwise.

Let $\overrightarrow{x}^{(1)} = (-2, -4)$, label= +1, and let $\overrightarrow{x}^{(2)} = (-1, 0)$, label = -1. Is there a perceptron that classifies these examples correctly?

Q2: Is there a perceptron function with a unit-length weight vector and a threshold value of 0 that classifies the above examples correctly?

Q3: Which learning-feasibility restrictions do $\overrightarrow{x}^{(1)}$ and $\overrightarrow{x}^{(2)}$ above satisfy?

Q4: Suppose

$$\overrightarrow{x}^{(1)} = \left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right)$$

with a label of +1. We claim that the PLA produces $\overrightarrow{w}^{(1)} = \overrightarrow{x}^{(1)}$ in response to this instance. Give an instance $\overrightarrow{x}^{(2)} \neq \overrightarrow{x}^{(1)}$ such that the two instances (with labels) satisfy our restrictions and such that the PLA sets $\overrightarrow{w}^{(2)} = \overrightarrow{w}^{(1)}$ in response.

Q5: Now give an $\overrightarrow{x}^{(3)}$ (with label) such that an update occurs.